

**DAIMLER**

# **Sales Operations Engine (SOE):**

## **System documentation**

Project: SOE  
Release: 2.1  
Last published version: 1.2.0  
Date: 2016-01-22



## Document History

Version	Date	Change Description / Reason for Change	Prepared by
	2016-01-22	SOE SRS 042 – Consumer Country SOE SRS 045 – Integration of the V-class	Adam Lima (Capgemini Deutschland GmbH)
	2015-12-04	SOE SRS 058 – Redesign of Batches for License Management SOE SRS 057 – Integration DFS-User SOE SRS 059 - Redesign of Batches for License Management - Part 2	Florentin Mehlbeer (Capgemini)
	27.09.2015	SOE SRS 052 – License management	Dominik Berndl(Capgemini Deutschland GmbH)
	10.09.2015	SOE SRS 047 – Diverse Specification Issues SOE SRS 051 – Service Categories	Florentin Mehlbeer (Capgemini Deutschland GmbH)
	09.09.2015	SOE SRS 037 - Versioning Of Services SOE SRS 041 - Boolean Logic for Equipment Codes SOE SRS 043 - Trust Level SOE SRS 046 - Maintain outletOutletIds in MBC countries UserStory_FetchVehicleDataWithoutLocale	Dominik Berndl (Capgemini Deutschland GmbH)
	22.06.2015	SRS 044 (“Unique Subject”)	Matthias Kühn (Capgemini Deutschland GmbH)
	18.06.2015	SRS 040 (“Change Year”)	Matthias Kühn (Capgemini Deutschland GmbH)
	11.06.2015	SRS 039 (“Support of Adapter Services”)	Florentin Mehlbeer (Capgemini Deutschland GmbH)
	05.06.2015	SRS 036 (“Generic Master Data”) SRS 038 (“Service Activation Rel 2.0”)	Matthias Kühn (Capgemini Deutschland GmbH)
1.2.0	21.05.2015	SRS 035 (“Internationalization”)	Matthias Kühn (Capgemini Deutschland GmbH)
	19.01.2015	<ul style="list-style-type: none"> <li>• Fixed includes of subdocuments: “//”, “/” replaced by “\\”</li> <li>• Fixed main heading of chapters “Document Management“ and “Master Data Change Management”</li> </ul>	Florentin Mehlbeer (Capgemini Deutschland GmbH)
	08.01.2015	<ul style="list-style-type: none"> <li>• Input, Output, Exceptions moved from IIF to corresponding AF</li> <li>• Introduced some cross-references in IIFs to corresponding AFs where they were missing</li> <li>• Replaced section “External View” by its former subsections “Offered Interfaces” (now: “External View – Offered Interfaces”) and “Consumed Interfaces” (now: “External View – Consumed Interfaces”).</li> <li>• Renamed section “Internal View” to “Internal View – Offered Interfaces”</li> </ul>	Florentin Mehlbeer (Capgemini Deutschland GmbH)

		<ul style="list-style-type: none"> <li>Added sections “Dialogs”, “Batches”, “Error Messages” where missing</li> <li>Spelling: “dialogue” changed to “dialog”</li> <li>Removed tracked format changes</li> <li>Set year to 2015 in footers</li> </ul>	
	19.12.2014	SRS 016 (“Change Session++”) deleted	Matthias Kühn (Capgemini Deutschland GmbH)
	08.12.2014	SRS 034 (“Blank Terms Of Use”)	Matthias Kühn (Capgemini Deutschland GmbH)
	06.11.2014	SRS 032 (“Support for consumer country in SARS”)	Diana Balogh (Capgemini Deutschland GmbH)
	21.10.2014	SRS 029 (“External interface to CCM”)	Matthias Kühn (Capgemini Deutschland GmbH)
	20.10.2014	SRS 030 ("Service Assignment Rules for Sales Types")	Thomas Hirschberger (Capgemini Deutschland GmbH)
	09.10.2014	SRS 026 (“Testsimulator for SOE vehicle and service master data”) V1.1 SRS 027 (“Trigger for master data changes”)	Matthias Kühn (Capgemini Deutschland GmbH)
	15.09.2014	Updae AF_SetUserAgreementStateOfConsent (user will not be informed when the acceptance status of the user agreement was not changed)	Juewei Jin (Capgemini Deutschland GmbH)
1.1.0	09.09.2014	Add new IIF_FetchVehicleDataWithoutLocale Add new AF_FetchVehicleDataWithoutLocale Add new IF_ODC_GetVehicleDataWithoutLocale Updated IF_CPD_UpdateMbcServiceAvailability with the Response Codes 100/102	Juewei Jin (Capgemini Deutschland GmbH)
	03.09.2014	SRS 015 (“LifeTraffic SD”) SRS 017 (“Print Approval”) SRS 016 (“Change Session++”) SRS 026 (“Testsimulator for SOE vehicle and service master data”)	Matthias Kühn (Capgemini Deutschland GmbH)
1.05	07.08.2014	SRS 023 (“New attribute master data services”) SRS 024 (“Country specific vehicle product master data”) SRS 024 (“Country specific vehicle product master data”) SRS 010 (“(De)Activation of Skip Loss service”) SRS 025 (“Deferred Service Availability”)	Diana Balogh (Capgemini Deutschland GmbH)
	25.06.2014	SRS 022 (“Country dependent User Agreements”)	Diana Balogh (Capgemini Deutschland GmbH)
	13.06.2014	New error message for AF_InformOfCustomerDataChange/ IF_SOE_InformOfCustomerDataChange	Eva Pfäffle (Capgemini Deutschland GmbH)
	03.06.2014	Dynamic Fonts for Documents	Eric Nzuobontane (Capgemini Deutschland GmbH)

	14.05.2014	SRS 020 (“Synchronous Backend Communication”) – v1.01	Juliane Rusinowski (Capgemini Deutschland GmbH)
	14.05.2014	SRS 011 (“Live Traffic standardisation”) – v1.01	Juliane Rusinowski (Capgemini Deutschland GmbH)
	12.05.2014	Extend Fields returned for not MBC countries with EMAIL and BIRTHDAY <b>AF_GetProfileFieldsUsedByCountry</b> <b>IF_GetProfileFieldsUsedByCountry</b>	Diana Balogh (Capgemini Deutschland GmbH)
	29.04.2014	Anpassungen aus SRS 013 (“Simplify creation of new versions fo document elements”) – v1.0 wieder eingefügt.	Juliane Rusinowski (Capgemini Deutschland GmbH)
	29.04.2014	Anpassungen aus SRS 010 (“(De-)Activation of Skip Loss service”) – v1.0 wieder eingefügt.	Juliane Rusinowski (Capgemini Deutschland GmbH)
1.04	29.04.2014	Anpassungen aus SRS 013 (“Simplify creation of new versions fo document elements”) – v1.0 wurden für Release 1.0 wieder entferrnt.	Juliane Rusinowski (Capgemini Deutschland GmbH)
	29.04.2014	Anpassungen aus SRS 010 (“(De-)Activation of Skip Loss service”) – v1.0 wurden für Release 1.0 wieder entferrnt.	Juliane Rusinowski (Capgemini Deutschland GmbH)
	29.04.2014	SRS 020 (“Synchronous Backend Communication”) – v1.0	Juliane Rusinowski (Capgemini Deutschland GmbH)
	28.04.2014	SRS 018 („Image Support in Documents“) – v1.0	Juliane Rusinowski (Capgemini Deutschland GmbH)
	28.04.2014	SRS 007 (“Communication with the Print Server Provider”) – v1.02	Juliane Rusinowski (Capgemini Deutschland GmbH)
1.03	13.02.2014	SRS 013 (“Simplify creation of new versions fo document elements”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	10.02.2014	SRS 012 (“Communication of user agreement changes”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	07.02.2014	SRS 010 (“(De-)Activation of Skip Loss service”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	07.02.2014	SRS 009 (“Optimize paper consumption”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	07.02.2014	SRS 008 (“Provide MBconnect services to MBconnect module for vehicle assignment”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	07.02.2014	SRS 007 (“Communication with the Print Server Provider”) – v1.01	Eva Pfäffle (Capgemini Deutschland GmbH)
1.02	01.02.2014	Tasklog ID #105 („Sysdoku“) Tasklog ID #116 („SRS004: Verhicle Master Data Import Anpassungen“)	Thorsten Rössner (Capgemini Deutschland GmbH)
	23.01.2014	Tasklog ID #113 (“Benennung IIF_GetMPCBySalesUnit) Tasklog ID #117 (“Relation von VehicleCache nach ModelSeries”) Tasklog ID #104 („Dublettenprüfung von Regeln“) Tasklog ID #99 („Internal view und External View aktualisieren“)	Dominik Willburger (Capgemini Deutschland GmbH)
	22.01.2014	Tasklog ID #101 (“Angepasste Logik SetUserAgreementStateOfConsent”) Tasklog ID #107 (“Länderprüfung in AF_GetAvailableUserAgreementsWithStateForUser”)	Dominik Willburger (Capgemini Deutschland GmbH)

		Tasklog ID #112 (“Unnötiger Konfigurationsparameter”)	
	22.01.2014	SOE SRS U01 (“Extraction of Vehicle Data Cache”) – v1.0	Dominik Willburger (Capgemini Deutschland GmbH)
	18.10.2013	SOE SRS 006 (“Inform of master data changes”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	18.10.2013	SOE SRS 005 (“Change Session”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	11.10.2013	SOE SRS 004 (“Vehicle Master Data Import”) – v1.0	Eva Pfäffle (Capgemini Deutschland GmbH)
	07.10.2013	SOE SRS 002 (“AuthorizationSupport”) – v1.0	Jens Kersting (Capgemini Deutschland GmbH)
	07.10.2013	SOE SRS 003 (“Extension of AccountDataSupport”) – v1.0	Jens Kersting (Capgemini Deutschland GmbH)
	07.10.2013	Tasklog ID #94: Removal of interface “IF_SOE_GetMissingCustomerFieldsForService”	Jens Kersting (Capgemini Deutschland GmbH)
	30.09.2013	SOE SRS 001 (“Legal contracts”) - v1.0	Jens Kersting (Capgemini Deutschland GmbH)
1.01b	30.07.2013	Korrekturen an Schnittstellenattributen	Christian Nicu (Capgemini Deutschland GmbH)
1.01	28.06.2013	System documentation: Version 1.01	Jens Kersting (Capgemini Deutschland GmbH)
	28.06.2013	Tasklog ID #31 (“UVS/Languages II”)	Jens Kersting (Capgemini Deutschland GmbH)
	27.06.2013	Tasklog ID #29 (“One Document for Vehicle Separation”)	Jens Kersting (Capgemini Deutschland GmbH)
	27.06.2013	Tasklog ID #24 (“Market Specific Fields for Services”)	Jens Kersting (Capgemini Deutschland GmbH)
	26.06.2013	Tasklog ID #25 (“User Agreement Service Mapping”)	Jens Kersting (Capgemini Deutschland GmbH)
	25.06.2013	Tasklog ID #27 (“Editable Service ID”)	Jens Kersting (Capgemini Deutschland GmbH)
	24.06.2013	Tasklog ID #20 (“UVS/Languages”)	Jens Kersting (Capgemini Deutschland GmbH)
	21.06.2013	Tasklog ID #18 (“CustomTags Dialog”)	Jens Kersting (Capgemini Deutschland GmbH)
	18.06.2013	Tasklog ID #26 (“InformOfCustomerDataChange”)	Jens Kersting (Capgemini Deutschland GmbH)
	17.06.2013	Remarks of final version	Jens Kersting (Capgemini Deutschland GmbH)
	12.06.2013	Tasklog ID #21 (“GetServicesCoveredByUserAgreement”)	Jens Kersting (Capgemini Deutschland GmbH)
	11.06.2013	Tasklog ID #9 (“GMT Timezone”)	Jens Kersting (Capgemini Deutschland GmbH)
	10.06.2013	Tasklog ID #3 (“UserType Enums”)	Jens Kersting (Capgemini Deutschland GmbH)
1.00	30.04.2013	Final version	Jens Kersting (Capgemini Deutschland GmbH)
0.99	16.04.2013	Version for final review	Jens Kersting (Capgemini Deutschland GmbH)
0.9	28.03.2013	Version for external review	Alex Plischke (Capgemini Deutschland GmbH)
0.8	22.03.2013	Version for internal review	Christian Nicu (Capgemini Deutschland GmbH)

---

## **Table of Contents**

<b>PROJECT:</b>	<b>SOE</b>	1
<b>DOCUMENT HISTORY</b>		3
<b>TABLE OF CONTENTS</b>		7
<b>1</b>	<b>INTRODUCTION</b>	ERROR! BOOKMARK NOT DEFINED.
1.1	<b>Objectives</b>	Error! Bookmark not defined.
1.2	<b>Scope of this Document</b>	Error! Bookmark not defined.
1.3	<b>Reading Recommendations and Document Structure</b>	Error! Bookmark not defined.
1.4	<b>Premises</b>	Error! Bookmark not defined.
1.5	<b>Changes towards conceptual document</b>	Error! Bookmark not defined.
<b>2</b>	<b>CORE BUSINESS FUNCTIONALITY</b>	ERROR! BOOKMARK NOT DEFINED.
2.1	<b>External View</b>	Error! Bookmark not defined.
2.2	<b>Internal View</b>	Error! Bookmark not defined.
2.3	<b>Core Concept</b>	Error! Bookmark not defined.
2.4	<b>Dialogs Overview</b>	Error! Bookmark not defined.
<b>3</b>	<b>COMPONENT „VEHICLE PRODUCTS“</b>	ERROR! BOOKMARK NOT DEFINED.
3.1	<b>Dialogs</b>	Error! Bookmark not defined.
3.2	<b>External View - Offered Interfaces</b>	Error! Bookmark not defined.
3.3	<b>External View - Consumed Interfaces</b>	Error! Bookmark not defined.
3.4	<b>Internal View - Offered Interfaces</b>	Error! Bookmark not defined.
3.5	<b>Implementation</b>	Error! Bookmark not defined.
3.6	<b>Batches</b>	Error! Bookmark not defined.
3.7	<b>Error Messages</b>	Error! Bookmark not defined.
<b>4</b>	<b>COMPONENT “SERVICES”</b>	ERROR! BOOKMARK NOT DEFINED.
4.1	<b>Dialogs</b>	Error! Bookmark not defined.
4.2	<b>External View - Offered Interfaces</b>	Error! Bookmark not defined.
4.3	<b>External View - Consumed Interfaces</b>	Error! Bookmark not defined.
4.4	<b>Internal View - Offered Interfaces</b>	Error! Bookmark not defined.
4.5	<b>Implementation</b>	Error! Bookmark not defined.
4.6	<b>Batches</b>	Error! Bookmark not defined.
4.7	<b>Error Messages</b>	Error! Bookmark not defined.
<b>5</b>	<b>COMPONENT SERVICE MANAGEMENT</b>	ERROR! BOOKMARK NOT DEFINED.
5.1	<b>Dialogs</b>	Error! Bookmark not defined.
5.2	<b>External View - Offered Interfaces</b>	Error! Bookmark not defined.
5.3	<b>External View - Consumed Interfaces</b>	Error! Bookmark not defined.
5.4	<b>Internal View - Offered Interfaces</b>	Error! Bookmark not defined.
5.5	<b>Implementation</b>	Error! Bookmark not defined.
5.6	<b>Batches</b>	Error! Bookmark not defined.

---

<b>5.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>6 COMPONENT „DOCUMENTS“ .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>6.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>6.2 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>6.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>6.4 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>6.5 Implementation .....</b>	Error! Bookmark not defined.
<b>6.6 Batches .....</b>	Error! Bookmark not defined.
<b>6.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>7 COMPONENT „CONTRACT MANAGEMENT“ .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>7.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>7.2 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>7.3 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>7.4 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>7.5 Implementation .....</b>	Error! Bookmark not defined.
<b>7.6 Batches .....</b>	Error! Bookmark not defined.
<b>7.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>8 COMPONENT “LICENSES” .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>8.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>8.2 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>8.3 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>8.4 Internal View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>8.5 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>8.6 Implementation .....</b>	Error! Bookmark not defined.
<b>8.7 Batches .....</b>	Error! Bookmark not defined.
<b>8.8 Error Messages .....</b>	Error! Bookmark not defined.
<b>9 COMPONENT „CONSENTS“ .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>9.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>9.2 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>9.3 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>9.4 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>9.5 Implementation .....</b>	Error! Bookmark not defined.
<b>9.6 Batches .....</b>	Error! Bookmark not defined.
<b>9.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>10 COMPONENT „AVAILABILITIES“ .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>10.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>10.2 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>10.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>10.4 Internal View – Offered Interfaces .....</b>	Error! Bookmark not defined.

---

<b>10.5 Implementation .....</b>	Error! Bookmark not defined.
<b>10.6 Batches .....</b>	Error! Bookmark not defined.
<b>10.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>11 COMPONENT “CONTEXT PROXY” .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>11.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>11.2 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>11.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>11.4 Internal View – Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>11.5 Implementation .....</b>	Error! Bookmark not defined.
<b>11.6 Batches .....</b>	Error! Bookmark not defined.
<b>11.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>12 COMPONENT “ACCOUNTDATASUPPORT” .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>12.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>12.2 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>12.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>12.4 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>12.5 Implementation .....</b>	Error! Bookmark not defined.
<b>12.6 Batches .....</b>	Error! Bookmark not defined.
<b>12.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>13 COMPONENT “MBCONNECT COUNTRIES” .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>13.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>13.2 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>13.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>13.4 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>13.5 Implementation .....</b>	Error! Bookmark not defined.
<b>13.6 Batches .....</b>	Error! Bookmark not defined.
<b>13.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>14 COMPONENT “AUTHORIZATION SUPPORT” .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>14.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>14.2 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>14.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>14.4 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>14.5 Implementation .....</b>	Error! Bookmark not defined.
<b>14.6 Batches .....</b>	Error! Bookmark not defined.
<b>14.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>15 COMPONENT “USERS AND ORGANIZATIONS” .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>15.1 Dialogs .....</b>	Error! Bookmark not defined.
<b>15.2 External View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>15.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.

---

<b>15.4 Internal View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>15.5 Implementation</b> .....	Error! Bookmark not defined.
<b>15.6 Batches</b> .....	Error! Bookmark not defined.
<b>15.7 Error Messages</b> .....	Error! Bookmark not defined.
<b>16 COMPONENT “GENERIC MASTER DATA”</b> .....	ERROR! BOOKMARK NOT DEFINED.
<b>16.1 Dialogs</b> .....	Error! Bookmark not defined.
<b>16.2 External View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>16.3 External View - Consumed Interfaces</b> .....	Error! Bookmark not defined.
<b>16.4 Internal View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>16.5 Implementation</b> .....	Error! Bookmark not defined.
<b>16.6 Batches</b> .....	Error! Bookmark not defined.
<b>16.7 Error Messages</b> .....	Error! Bookmark not defined.
<b>17 COMPONENT “MASTER DATA CHANGE MANAGEMENT”</b> ...	ERROR! BOOKMARK NOT DEFINED.
<b>17.1 Dialogs</b> .....	Error! Bookmark not defined.
<b>17.2 External View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>17.3 External View - Consumed Interfaces</b> .....	Error! Bookmark not defined.
<b>17.4 Internal View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>17.5 Implementation</b> .....	Error! Bookmark not defined.
<b>17.6 Batches</b> .....	Error! Bookmark not defined.
<b>17.7 Error Messages</b> .....	Error! Bookmark not defined.
<b>18 COMPONENT “MASTER DATA REPLICATION EXPORT”</b> ....	ERROR! BOOKMARK NOT DEFINED.
<b>18.1 Dialogs</b> .....	Error! Bookmark not defined.
<b>18.2 External View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>18.3 External View - Consumed Interfaces</b> .....	Error! Bookmark not defined.
<b>18.4 Internal View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>18.5 Implementation</b> .....	Error! Bookmark not defined.
<b>18.6 Batches</b> .....	Error! Bookmark not defined.
<b>18.7 Error Messages</b> .....	Error! Bookmark not defined.
<b>19 COMPONENT “MASTER DATA REPLICATION IMPORT”</b> .....	ERROR! BOOKMARK NOT DEFINED.
<b>19.1 Dialogs</b> .....	Error! Bookmark not defined.
<b>19.2 External View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>19.3 External View - Consumed Interfaces</b> .....	Error! Bookmark not defined.
<b>19.4 Internal View - Offered Interfaces</b> .....	Error! Bookmark not defined.
<b>19.5 Implementation</b> .....	Error! Bookmark not defined.
<b>19.6 Batches</b> .....	Error! Bookmark not defined.
<b>19.7 Error Messages</b> .....	Error! Bookmark not defined.
<b>20 COMPONENT „OUTBOUND MESSAGING“</b> .....	ERROR! BOOKMARK NOT DEFINED.
<b>20.1 Dialogs</b> .....	Error! Bookmark not defined.
<b>20.2 External View - Offered Interfaces</b> .....	Error! Bookmark not defined.

---

<b>20.3 External View - Consumed Interfaces .....</b>	Error! Bookmark not defined.
<b>20.4 Internal View - Offered Interfaces .....</b>	Error! Bookmark not defined.
<b>20.5 Implementation .....</b>	Error! Bookmark not defined.
<b>20.6 Batches .....</b>	Error! Bookmark not defined.
<b>20.7 Error Messages .....</b>	Error! Bookmark not defined.
<b>21 BUSINESS OBJECT MODEL.....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>21.1 Overview .....</b>	Error! Bookmark not defined.
<b>21.2 GenericMasterData .....</b>	Error! Bookmark not defined.
<b>21.3 MasterDataReplicationExport .....</b>	Error! Bookmark not defined.
<b>21.4 MasterDataReplicationImport .....</b>	Error! Bookmark not defined.
<b>21.5 ChangeSessionManagement .....</b>	Error! Bookmark not defined.
<b>21.6 OutboundMessaging .....</b>	Error! Bookmark not defined.
<b>21.7 MBconnectCountries .....</b>	Error! Bookmark not defined.
<b>21.8 AccountDataSupport .....</b>	Error! Bookmark not defined.
<b>21.9 AuthorizationSupport .....</b>	Error! Bookmark not defined.
<b>21.10      ContractManagement.....</b>	Error! Bookmark not defined.
<b>21.11      ServiceManagement.....</b>	Error! Bookmark not defined.
<b>21.12      Documents.....</b>	Error! Bookmark not defined.
<b>21.13      Services .....</b>	Error! Bookmark not defined.
<b>21.14      VehicleProducts .....</b>	Error! Bookmark not defined.
<b>21.15      data types.....</b>	Error! Bookmark not defined.
<b>22 USER / ORGANIZATION / ENTITLEMENTS .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>22.1 Business Roles .....</b>	Error! Bookmark not defined.
<b>22.2 Application Roles.....</b>	Error! Bookmark not defined.
<b>22.3 Assignment of Application Roles to Business Roles</b>	Error! Bookmark not defined.
<b>22.4 UI Entitlements.....</b>	Error! Bookmark not defined.
<b>22.5 Functional Entitlements .....</b>	Error! Bookmark not defined.
<b>23 ROLLOUT ASPECTS .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>23.1 Language maintenance .....</b>	Error! Bookmark not defined.
<b>23.2 Dictionary .....</b>	Error! Bookmark not defined.
<b>24 ERROR MESSAGE RESPONSE CODE MAPPING .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>24.1 Response Codes for Interface Errors.....</b>	Error! Bookmark not defined.
<b>25 OPEN ISSUES .....</b>	ERROR! BOOKMARK NOT DEFINED.
<b>26 REFERENCES .....</b>	<b>639</b>
<b>27 INDEXES.....</b>	<b>640</b>
<b>27.1 Figures .....</b>	<b>640</b>
<b>27.2 Tables.....</b>	<b>642</b>



---

# 1 Introduction

With the project MBconnect Daimler delivers an online connection to Mercedes-Benz passenger cars vehicles. Via this online connection special services from the departments RD, GSP and BC are offered to the end-customer. This online connection requires a contract ("Nutzungsvereinbarung") between Daimler AG and the end-customer. This part of the MBconnect processes is called service achievement ("Leistungserrung"), which is supported by several systems. This document thereby focuses on the specification of the SOE (Sales & Operation Engine) system.

In this chapter, the objectives of the system, scope of this document, reading recommendations and document structure as well as the premises will be mentioned.

## 1.1 Objectives

Prior to the usage of the MBconnect services, every customer has to sign certain user agreements permitting Daimler to collect and analyze the data. The main objective of SOE is to administrate these contracts between Daimler AG and the end-customers. For a holistic, more general view on the SOE system, this specification refers to a document called "SOE Rahmenkonzept\_V0.99".

For managing the user agreements, SOE offers the following main functions:

- Maintenance of vehicle data (sales perspective only)
- Maintenance of MBconnect services (sales perspective only)
- Generation, maintenance and dispatching of documents
- Formatting of user addresses
- Administration of contracts and user agreements

## 1.2 Scope of this Document

This document represents the detailed specification for the system SOE. The following building blocks are described herein:

- Dialogs and documents
- Components and interfaces
- Batches
- External interfaces
- Business object model
- Roles

---

## **1.3 Reading Recommendations and Document Structure**

Following a comprehensive requirements analysis, the conceptual document named “SOE Rahmenkonzept\_V0.99” was created. It contains the parameters and scope for the SOE system. It is recommended to read due to an extended and deeper understanding.

---

Regarding the functions mentioned above, this document is structured as follows:

- Chapter 2: Describes the basic business functionalities and core concepts including an internal and external view on the system.
- Chapter 3: Component “Vehicle Product”: Maintains the vehicle product data, described from a sales perspective.
- Chapter 4: Component “Services”: Maintains the services and the vehicle-service-relations, described from a sales perspective.
- Chapter 5: Component “Documents”: Generates, maintains and dispatches the documents.
- Chapter 6: Component “Contract Management”: Administrates contracts and user agreements.
- Chapter 7: Component “AccountDataSupport”: Provides information on how to layout the customer data in the different systems and proves whether a specific field is mandatory for a specific country or not.
- Chapter 8: Component “MBconnect Countries”: Generates and maintains the countries that are supported by MBconnect.
- Chapter 9: Component "Authorization Support": Responsible for managing the authorization related data in SOE.
- Chapter 10: Component “Users and Organizations”: Provides information on the available markets.
- Chapter 11: Component "Change Session Management": Manages changes concerning vehicle, document element, service and country related master data.
- Chapter 12: Component “Outbound Messaging”: This component manages the outbound communication (e.g. letter and email based communication between Daimler AG and the end customer).
- Chapter 13: Business Object Model: Describes the different data objects and reveals the relationships between them.
- Chapter 14: Migration
- Chapter 15: User / Organization / Entitlements
- Chapter 16: Rollout Aspects
- Chapter 17: Error Messages Response Code Mapping
- Chapter 18: Open Issues

Further, every component is structured in the same way:

- Dialogs (if existing): Contains scribbles of the component’s dialogs and shows the dialog flow.
- Internal view: Describes the internal interfaces provided by the component and used within the SOE system.

- External view: Describes the external interfaces provided by the component and used by external systems.
- Implementation: Gives further insights on the implementation of the external interfaces (also called application functions or AF).
- Batches (if existing): Lists the batch processes administrated by the component.
- Error messages: Lists the possible error messages created by the component.

Though, it is recommended to read the whole specification, different groups may focus on different chapters of this document:

Group	Focus
Readers interested in the review of SOE	All chapters
Readers using adjacent systems	Target Business Functionality (Chapter 2), External interfaces (Chapter 3.3, 4.3, 5.3, 6.2, 7.2 and 9.2), Implementation (Chapter 3.4, 4.4, 5.4, 6.3)
Administrator of SOE	Target Business Functionality (Chapter 2), Dialogs (Chapter 3.1, 4.1 and 5.1), Business Object Model (Chapter 10)
Readers interested in the documents and component "Documents"	Target Business Functionality (Chapter 2), Component "Documents" (Chapter 5), Dialogs (Chapter 3.1, 4.1 and 5.1), Business Object Model (Chapter 10)

Table 1: Reading recommendations

## 1.4 Premises

### 1.4.1 General

- (P1) For I18n use Locale (ISO 639 + ISO 3166 = RFC5646) e.g. "en\_US"
- (P2) For the communication about MB organizational units, the GEMS OutletId (=GSSN "dcxGSSNOutletOutletId") will be used. PC tree will be used.
- (P3) The SOE system always works with the country information associated with the customer. Only for the check of the vehicle configuration executed by POS, the locale and market information of the retailer will be used.
- (P4) SOE supports multiple master data languages. The concrete language list itself is not defined yet. The list will be configured in the Application Configuration (see chapter 2).
- (P5) For the interface to UVS, the ordering market („Bestellermarkt“) is required. It is used for price calculation, which is not needed for SOE. In analogy to PVS, the market Switzerland is used hardcoded for all UVS calls.

- 
- (P6) It is assumed, that MyM can use the same services as agreed by POS. If additional service interfaces are required in the future, they are not part of this specification and therefore have to be declared as CR.
  - (P7) The PDF interface for the printing service is described lonely by the requirements of SOE. It is supposed that it can be implemented by a printing provider.
  - (P8) There will be NO import or export of translations, such as Excel or CSV.
  - (P9) The SOE administration dialogs will not be branded. A change of the layout in the dialogs will not be supported.
  - (P10) The user interfaces described within this specification will be accessible by Microsoft Internet Explorer 8.
  - (P11) The interfaces that will be used by MyMercedes are the same than the ones used by POS.

## 1.4.2 Vehicle Products

- (P12) Model series can be set valid to certain markets. This is mainly needed to be able to filter the list returned by **IF\_SOE.GetServiceMasterData** (→ see chapter 4.2.4). However this does not affect the evaluation of service assignment rules. The evaluation does not consider the market of a model series.

## 1.4.3 Documents and Contracts

- (P13) The current premise is that SOE supports country and language specific documents, descriptions etc. Currently, there will be no support for market specific documents and descriptions.
- (P14) The premise for documents is that they are in PDF format, while email is plain text.
- (P15) All documents are vehicle independent, though they may contain dynamic vehicle information.
- (P16) The formatting of names and addresses is based on the country of the customer address.
- (P17) Customer data changes initiated by MBC POS are instantly available on CPD for POS and other systems.
- (P18) For already maintained user agreements, there is no automatic update on the coverage of services for customers that have already signed that user agreement. This means if the user agreement is changed to cover one more service, there is no update for customers that have already signed the customer agreement previous to the change and will not be able to use that additional service. This change will only affect future consents given.
- (P19) There is no mapping defined among user agreements exclusively available in two different countries. E.g.: if user agreement A is available only in Germany and user agreement B is available only in Ireland. Due to legal specific constraints in Ireland the user agreement B is an adapted version of user agreement A (with minor modifications). A mapping between user agreement A and B is not sup-

---

ported in SOE. This implies that for a customer that moves from Germany to Ireland, no migration of the consent for user agreement A to user agreement B is done.

- (P20) The document “Vehicle Separation Authorization“ will be generated by SOE. Further CPD-Related documents like Mails or other documents are not created by SOE.
- (P21) The deletion of created master data is not allowed in any of the document dialogs. This is necessary to preserve the state of documents. Since the documents are also versioned, there is no need for a delete option.
- (P22) SOE does not consolidate multiple PDF bundles that are to be sent to the same customer into one PDF document: Per trigger type (e.g. user agreement changes, vehicle separation etc) SOE will generate one PDF bundle, even if they are addressed for the same customer.

#### 1.4.4 Services

- (P23) Service assignment rules are defined on a global level (not on a market level).
- (P24) For already existing vehicle account assignments, there is no automatic update for the list of possible MBconnect services. This means, that if a service assignment rule is modified and now points to new services, the already assigned vehicle stays untouched. Only in case of a new assignment or a manual change of the user agreement consents the service assignment rules are re-evaluated.
- (P25) The required customer data fields per service are not country specific. This means, that it is not possible to configure different requirements regarding the recorded customer data for services in different countries. Technically, it is possible to define that a specific field is optional for a specific country, but that it is mandatory for a specific service. It is the responsibility of the administrator to avoid such combinations.
- (P26) Master data relevant: the telematic services offered in China are new services. These services are not covered by any user agreement.
- (P27) The <services to user agreement> assignment is maintained country independent.
- (P28) An MBconnect service can be assigned only to one user agreement. This implies that if a service is covered by a user agreement *not available* in a country X, that service cannot be assigned to another user agreement *available* in that country X.
- (P29) If a service is covered by a user agreement available only for a subset of countries, then this service cannot be made available in other countries without a user agreement.

---

#### **1.4.5 Account data support**

- (P30) The data format for exchanging customer data is the UserType as described in the CPD interface definition document. This specification is based on the version 0.4.1, dated to the 26.04.2013.
- (P31) CPD sends the notification about customer data changes (e.g. a user changes his address country). If a customer moves to a different country, SOE will notify CPD to deactivate the services covered by those user agreements that are not supported in the other country.
- (P32) All country specific logic (e.g. formatting of names and addresses) is derived from the customer address country only.

### **1.5 Changes towards conceptual document**

Following a comprehensive requirements analysis, the conceptual document named "SOE Rahmenkonzept\_V0.99" was created. Since this requirement analysis and in approval with all stakeholders, some changes have been made for this specification. These changes are:

ID	Topic	Description
8	Unverified account	CIAM ID does not represent the leading customer system anymore. Instead, CPD uses its own user ID.
10	Letter mail "Heinz"	SOE defines a dedicated interface for transmitting the generated PDF to a letter mail component. The question concerning the responsibility for printing the letter though remains unanswered.
11	Country specific account dialogs	CPD does not offer a service for country specific structuring of the accounting dialogs. Now, SOE offers a service towards POS supporting the formatting of the customer data.
12	Country specific address formatting	CPD does not offer a service for formatting the customer address towards POS. Now, SOE offers this service towards POS.
13	Information on user agreement changes	-
14	Live Traffic	-
19	Enhancement of service maintenance	SOE now supports the country specific maintenance of services text blocks. Interfaces to POS / MyM will be provided.

Table 2: Changes towards conceptual document

## 2 Core Business Functionality

### 2.1 External View

SOE Regions and SOE MDM need connections to several third party systems and to itself. This chapter describes the systems and the interactions with them from a holistic perspective. Figure 1 illustrates the interfaces between SOE Regions, SOE MDM and the external systems. Technically, all requests between SOE Regions and the external systems are routed through the TSB, all requests between SOE MDM and SOE Regions are routed through the ASB. Due to an improved readability the diagram only shows the combined interfaces. Only for the interfaces "GetFormattedAddress" and "GetContractDataForUser", the TSB has to provide own interfaces which are not detailed in this diagram.

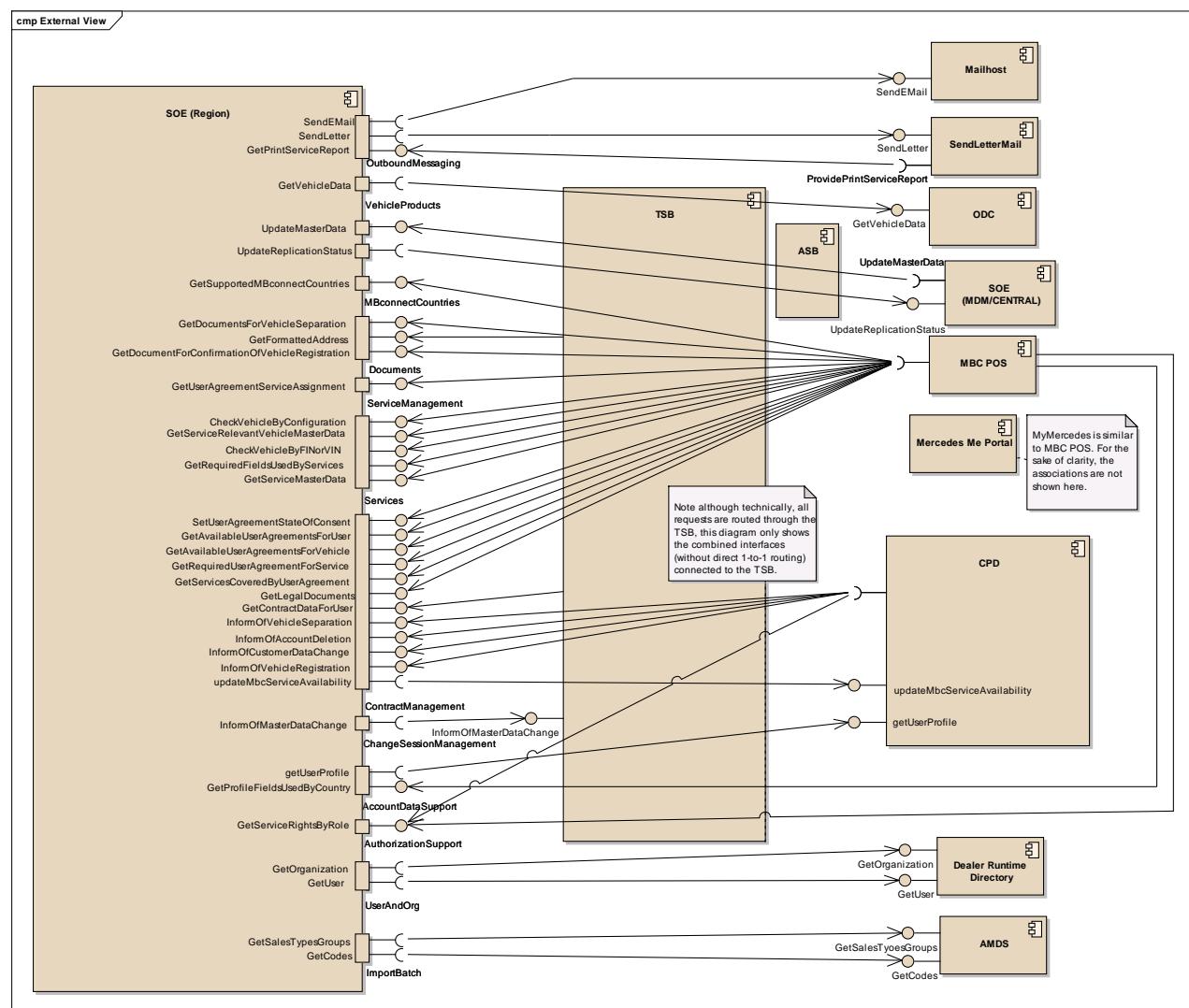


Figure 1: Overview SOE's functional interfaces and the related third party systems that are connected to them

#### SOE Regions are connected to the following systems:

- SOE MDM (Master Data Management)

---

The general SOE instance in which you are able to maintain the master data.

- ODC (Operational Data Cache)

SOE accesses ODC to determine a vehicle's configuration from a FIN/VIN. That configuration has then already correct texts (used for POS or on documents). That is the reason why SOE does not access UVS or VeDoc directly.

- PrintServer

This system is required by SOE whenever real paper documents have to be printed and sent to a customer via letter mail. This is necessary whenever user agreements or terms and conditions change and the customer needs to be informed and has not provided an e-mail address.

- Mailhost

SOE uses an official Daimler mailhost to send out e-mails towards customers. This is needed whenever documents (user agreements or terms and conditions) change or in order to send out copies of the documents a user has agreed to.

- MyMercedes / MBC POS

MyMercedes as well as MBC POS use SOE to access customer data related to contracts. Beside of that, the two systems allow viewing user agreements and accepting them or revoking a given user agreement.

In addition to that MBC POS needs to generate and print PDF-Versions of the user agreements needed during the sales process. SOE also provides the information which fields are mandatory for accepting user agreements or for activation of certain services.

- TSB (Telemetry Service Bus)

Technically, all calls are routed through the TSB. The diagram shown above suggests it by the associations that go through the TSB component. In order to preserve readability for 1-on-1 calls between a third party system and SOE, the diagram shows direct connections anyway.

The TSB is also used to combine one call from a third party system to several calls towards other systems. In our case it combines calls from MBC POS to SOE and CPD. The diagram shown above only depicts the access from TSB to SOE in order to correctly format addresses in the country specific manner and to return a user's contract overview. The interaction between MBC POS or CPD and TSB is not shown as it is functionally not of interest for SOE.

- CPD (Context and Profile Data)

CPD is the master system for storing the customer data and the relationships between a user/customer and a vehicle. More than that, the CPD knows if a certain service is activated or deactivated.

The SOE interacts with the CPD in order to maintain a valid state. SOE informs the CPD about contract related changes and the consequences which services are allowed or not (i.e. triggered whenever a user agrees to a user agreement or revokes his consent).

---

On the other side, CPD provides access to customer profile data (needed when generating documents), informs the SOE whenever a vehicle was registered/deregistered for a customer, when customer profile data changes or if a user account has been deleted/deactivated.

- DRD (Dealer Runtime Directory)

The DRD is used to determine org units or to access user data (incl. the role of a user).

#### **SOE MDM is connected to the following systems:**

- SOE Regions

The regional SOE instances in which regional dynamic data is stored.

- AMDS (Automated Master Data Supply)

SOE accesses AMDS in order to import vehicle master data like model series, sales types and equipment codes including the equipment codes' localized descriptions.

Note: All calls from other systems have to be authorized by the SOE system. In this respect, the TSB system will commit a technical user with each interface call. To improve the readability, the technical user will not explicitly be mentioned in the interface descriptions of this system.

## **2.2 Internal View**

SOE is decomposed internally into several functional components of which each one is responsible for a functional segment exclusively. Figure below shows all of them and their relationships among each other as well as to the external interfaces.

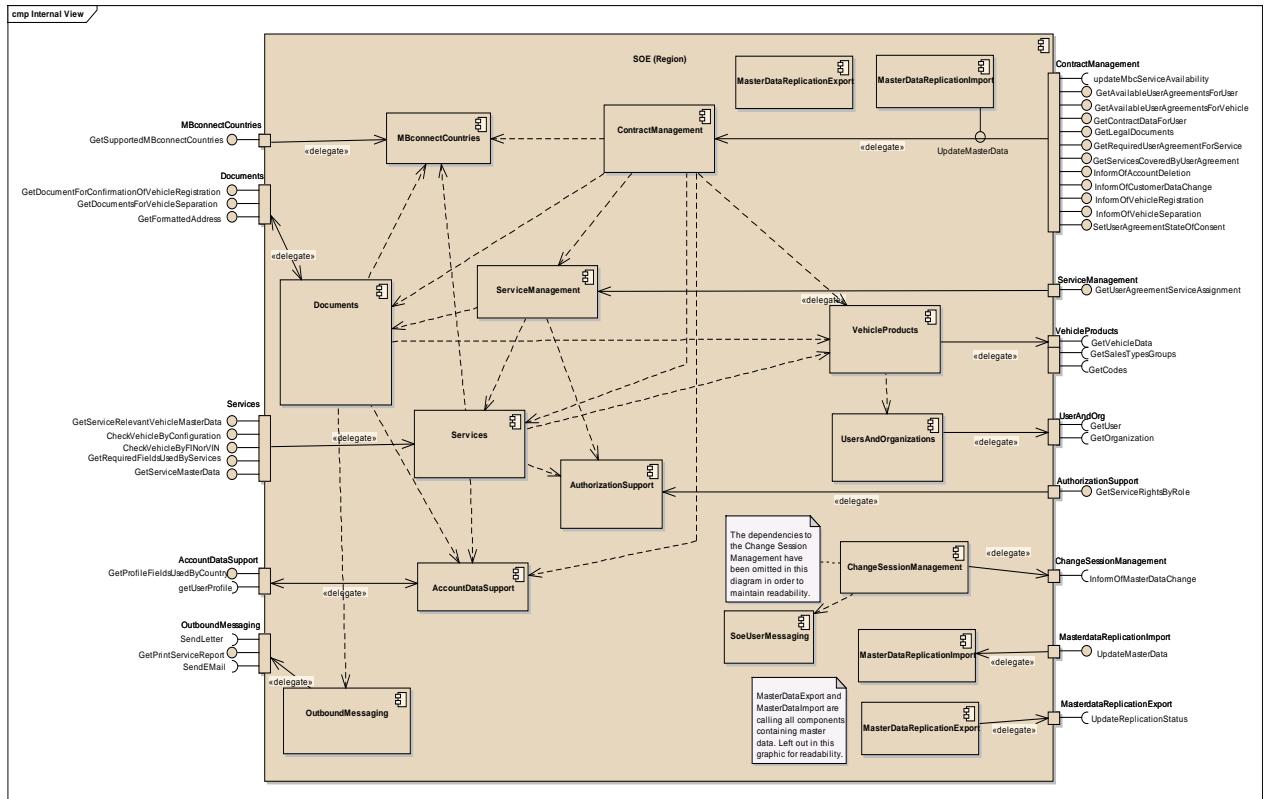


Figure 2: Internal view of SOE showing the functional architecture and the decomposition into components

The details of the components and their interfaces are described in individual sections subsequently in this document. Here only an overview is given:

- **Contract Management**

Contract Management represents the functional core of SOE. Here all contract relevant data is stored (who has contracts i.e. user agreements and when have these been signed, how long are services available for a customer, which contract versions are valid for a customer, etc.) Therefore it offers several external interfaces to provide access to that data and to allow updates on it. In addition to that, ContractManagement offers a couple of “Inform\*"-interfaces which are either used to process changes on customer data of which the CPD informs the SOE or the other way to inform the CPD of changes regarding allowed services from the contractual point of view.

- **Vehicle Products**

This component is responsible to store and manage vehicle related master data. Therefore it provides access to UVS, so that SOE can determine a vehicle's configuration (needed to support the sales process). It is also responsible to store first registration dates for vehicles which have contracts. The component additionally accesses AMDS in order to provide vehicle related master data like model series, sales types and equipment codes.

- **Services**

---

The Services component holds and maintains all master data that is related to MBconnect services. It allows defining rules for which vehicles and equipments certain MBconnect services are allowed/possible and what services are covered by a user agreement. In addition to that it has the knowledge which fields of the UserType (which contains the user profile data) are mandatory (if any) for activating a service. It offers external interfaces used by MBC POS or MyMercedes to externalize its data.

- **Documents**

The Documents component has two major tasks: To generate documents as mail or PDF and to facilitate the administration of documents. Besides of that it can trigger the printout of letters which are sent to customers by post. In addition to that it contains the functional knowledge of how to format an address in a correct country specific way. The Documents component also offers dialogs for functional administrators to allow them to maintain the master data. The component externalizes its services to allow other systems to get documents as PDF or to format an address.

- **AccountDataSupport**

This component encapsulates the knowledge to read a UserType (user profile data) from CPD. Besides formatting the addresses, the component knows which address fields are mandatory in a certain country. The knowledge of mandatory fields is used by MBC POS and therefore accessible from outside.

- **MBconnect Countries**

This component encapsulates the knowledge about an MBconnect country. In detail, countries that are supported by MBconnect are created and maintained here.

- **Change Session Management**

This component models the concept of the “Change Session”.

- **Master Data Replication Export**

This component models the export of master data from SOE MDM to SOE Regions.

- **Master Data Replication Import**

This component models the import of master data into SOE Regions.

- **Outbound Messaging**

This component has the responsibility to establish the communication with the external components and to route the information received from other internal components (e.g. component “Documents” generate emails and letters to be sent out to the customers) to the external messaging systems (MailServer and PrintServer). Additionally this component persists tracking information related to the documents that have been processed and sent to the external messaging systems

- **UsersAndOrganizations**

Provides access to user and org unit data like detail information about org units or hierarchy information. This component also provides a cache for that information.

- Service Management

Is responsible for managing the relationship between the user agreements and the services:

- updating, retrieving and validating the relationship between services and user agreements.
- persisting the entities that store the relationship between services and user agreement
- manage dialogs that combine business cross-functionality of the components "Documents" and "Service"

The diagram shown in Figure 3 illustrates how the customer address is formatted depending on the calling context:

- The parameters <sendFromCountry> and <localeForAddressCountry> are set only within the call AF\_InformCustomer. (The letter template must contain the address of the customer in a specific format (requested by the german post).

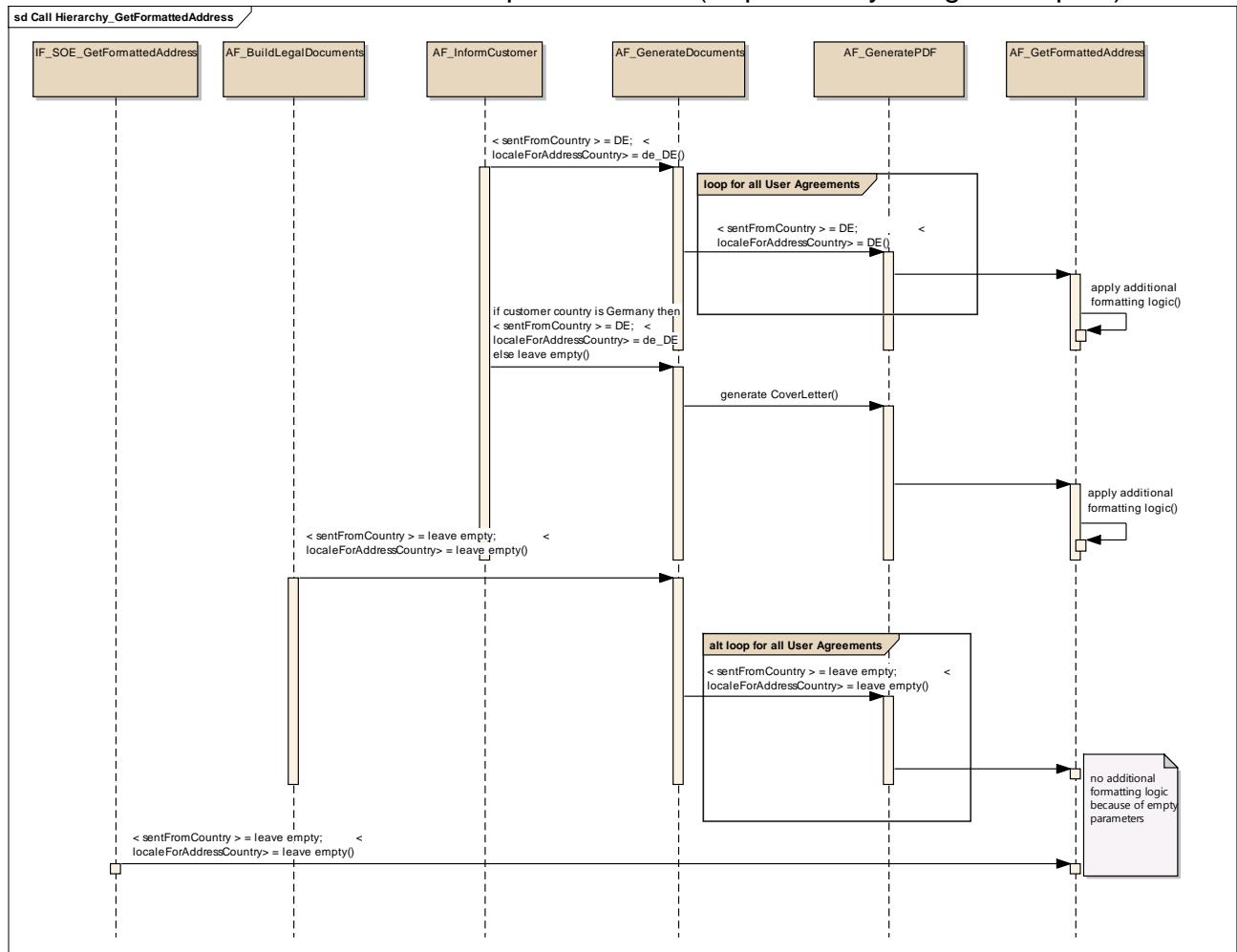


Figure 3: Formatting of addresses

## 2.3 Core Concept

### 2.3.1 Vehicle Master Data and Service Assignment Rules

SOE employs vehicle master data to define availability of MBconnect services for given vehicle configurations in given markets, and offers interfaces that allow systems like MBC POS Module to check which MBC services are available for given vehicles.

#### Vehicle Master Data

Vehicle master data includes information such as model series, sales types, model years, equipment codes and consumer country (available Mbconnect countries) describing which vehicle configurations are available in different markets. As sketched in Figure 4, SOE imports such vehicle master data from source system AMDS, stores it, and allows to maintain it manually in cases it might be necessary.

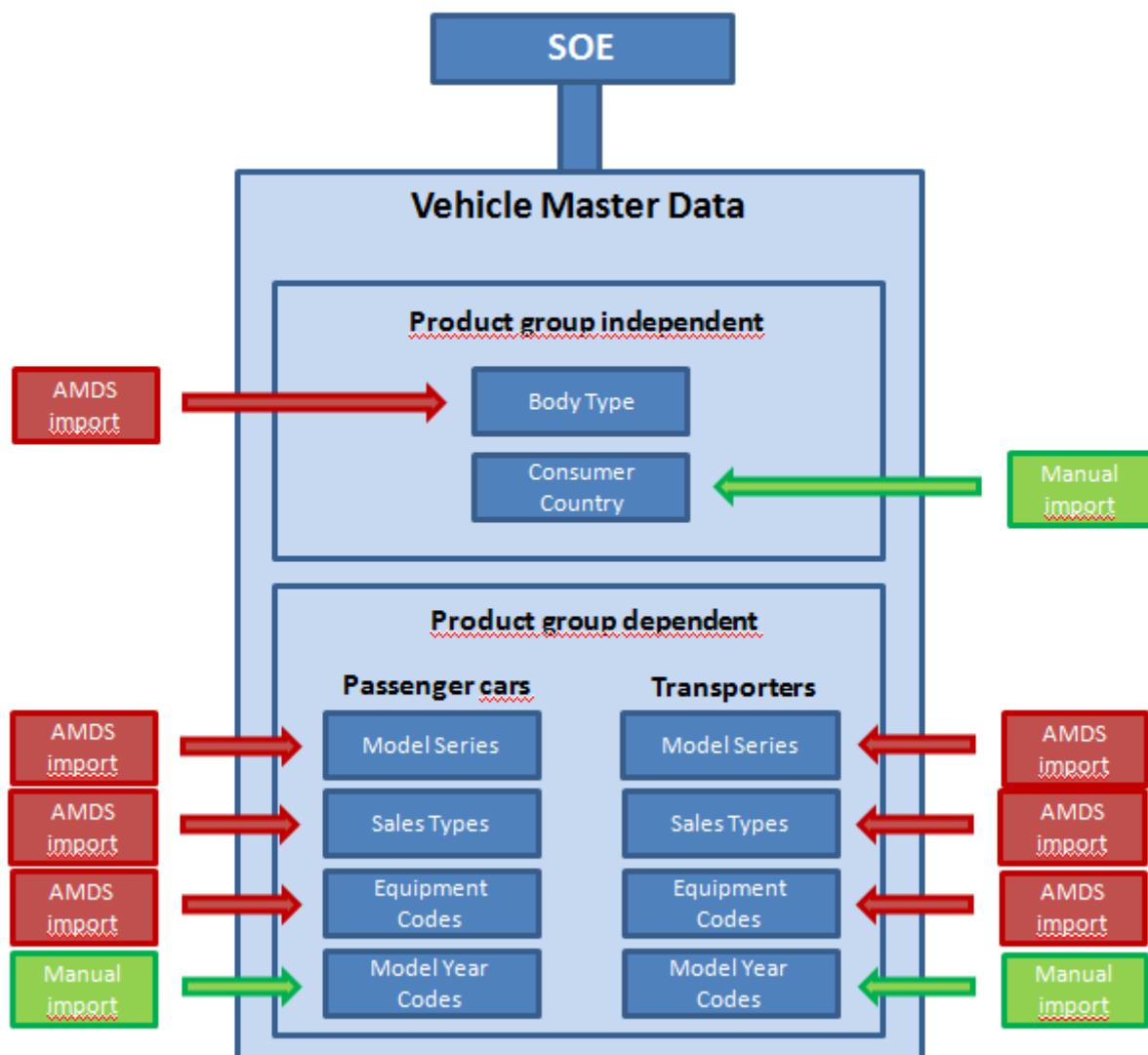


Figure 4: Import and storage of vehicle master data in SOE

#### Product Groups

---

As the various vehicle types, passenger cars, transporters, trucks etc., distributed by the Daimler corporation are developed independently by different departments varying definitions of equipment codes exist depending on the vehicle type. To allow different code definitions, the entirety of existing model series are separated into product groups, e.g. model series 205 (C-Class) is part of the product group "passenger cars" and model series 906 (Sprinter) is part of the product group "transporter". The equipment codes available for each model series are separated into product groups according to the assignment of model series to product groups. By this means, identical equipment codes can have different meanings but can still be distinguished by additionally considering the codes' product groups.

Example:

- Code "H12" means "Holzausführung Wurzelnuss hell" for a C-Class vehicle, which belongs to the product group "passenger cars".
- Code "H12" means "Warmwasserzusatzeizung" for a Sprinter, which belongs to the product group "transporter".

For this purpose, SOE allows to separate vehicle master data, such as model series, sales types, equipment codes into product groups to distinguish multiple meanings of one code, such as the code "H12". This is achieved by adding an attribute "Product group" to the business objects representing the various kinds of vehicle master data.

Another effect of the independent development of sales types is that varying conventions to define sales types exist. Sales types of passenger cars are uniquely identified by their seven-digit baumuster and optional NST. Sales types of transporters (also called sales references) are identified by their eight-digit baumuster, SR1 context, and ModelNsrExtension. The baumuster of transporters contains information, such as the model series or the body type, but no information about the vehicle's motor as opposed to passenger cars' baumusters. This information, e.g. the equipped motor, is encoded in the SR1 context and the ModelNsrExtension. Both the SR1 context and the ModelNsrExtension are ordered arrays of codes which define basic equipment of the sales type including the motor as already mentioned above.

### **Service Assignment Rules**

Based on such vehicle master data, SOE allows to define which MBc services are available for given vehicle configurations and in given markets. Here, pieces of vehicle master data can be combined to form conditions to be fulfilled by a vehicle configuration so that a definable set of MBc services then is available (see Figure 5).

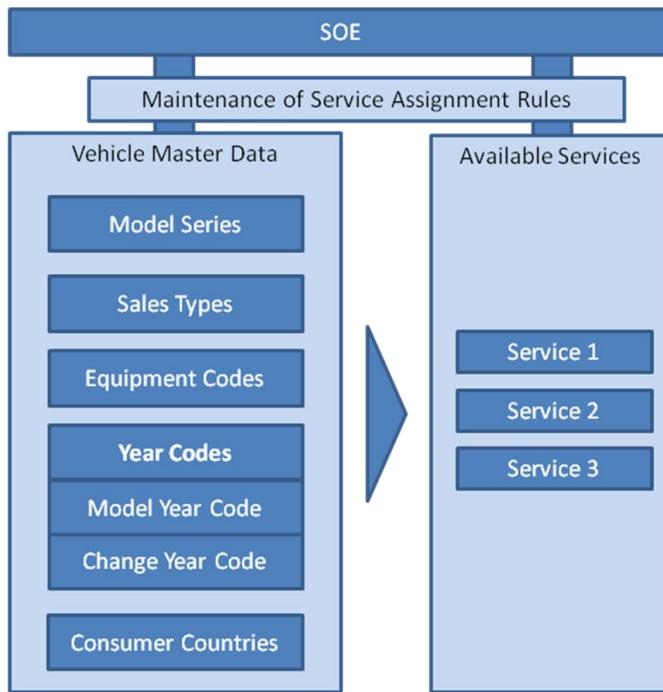


Figure 5: Maintenance of service assignment rules in SOE

In more detail, as illustrated in Table 3, service assignment rules each consist of a model series, an optional set of sales types, a set of year code combinations each including a model year code and an optional change year code, a set of equipment codes and an optional list of consumer countries that together form the conditions for vehicle configurations, and a set of the MBconnect services that are available to vehicles that fulfill the conditions.

Note that Table 3, however, does not include a rule that contains a set of sales types that can be defined by wildcards for baumusters. A wildcard for baumusters may for example take the form 205.0\* and describe the set of Sales Types whose baumuster numbers start with the prefix 205.0 thereby defining model series (205) and body type (0).

Rule ID	Model Series	Sales Types (optional)	Year Code Combinations (Model Year + (opt.) Change Year)	Equipment Codes	Consumer Countries (optional)	Services
1	205		805, 806, 807, 808, 809	06U and B54	„DE, FR, GB, ... (ECE15)“	1
2	205		805, 806, 807, 808, 809	350 and not 05U	„DE, FR, GB, ... (ECE15)“	2
3	205		805, 806, 807, 808, 809	06U	„DE, FR, GB, ... (ECE15)“	30, 31, 32, 33
4	205		805, 806, 807, 808, 809	05U	„DE, FR, GB, ... (ECE15)“	10, 21, 34
5	205		805, 806, 807, 808, 809	05U and (228 or 230)	„DE, FR, GB, ...	22

					(ECE15)“	
6	205		805	B57 and (06U or 05U) and 350 and not 228	„DE, FR, GB, ... (ECE15)“	23
7	205		805 + 055, 806, 807, 808, 809	B57	„DE, FR, GB, ... (ECE15)“	51
8	207		806, 807, 808, 809	06U and B54	„DE, FR, GB, ... (ECE15)“	1
9	207		806, 807, 808, 809	not 350	„DE, FR, GB, ... (ECE15)“	2
10	207		806, 807, 808, 809	06U	„DE, FR, GB, ... (ECE15)“	30, 31, 32, 33
11	207		806, 807, 808, 809	05U	„DE, FR, GB, ... (ECE15)“	10, 21, 34
12	222		806, 807, 808, 809	348	CN	35, 36
13	222		806, 807, 808, 809	348	US	45, 46
14	447	4478*	XZ0	EY3		501, 502, 503, 504, 536
15	447	4478*	XZ0	EY3 && EY5		2

Table 3: Exemplary selection of service assignment rules

The set of MBconnect services available for a given vehicle configuration is determined by finding the rules that are matched by the configuration. When a rule is matched, the MBconnect services contained in the rule are added to the set of available MBconnect services being built.

Example 1: Suppose the configuration is given by model series 205 (implicitly given by the sales type), sales type 205.040.1, model year 805, and set of equipment codes {350, 06U, B57}. By comparing these values with the according values from the rules in Table 3, one finds that the three rules with the IDs 6029, 6030 and 6033 are matched by the exemplary configuration. Hence, the MBconnect services 2, 30, 31, 32, 33 and 23 are available.

Example 2: Suppose the configuration is given by model series 205 (implicitly given by the sales type), sales type 205.040.1, model year 805, change year 055 and set of equipment codes {350, 06U, B57}. By comparing these values with the according values from the rules in Table 3, one finds that the four rules with the IDs 6029, 6030, 6033 and 6039 are matched by the exemplary configuration. Hence, the MBconnect services 2, 30, 31, 32, 33, 23 and 51 are available.

Example 3: Suppose the configuration is given by model series 207 (implicitly given by the sales type), sales type 207.020.3, model year 808 and equipment code {350}. By comparing these values with the according values from the rules in Table 3, none rule can be matched by this vehicle configuration. Especially the rule with the ID 6036 requires that to deliver MBconnect service 2 the vehicle must not have the equipment code 350.

---

Note that, to assure the quality of service assignment rules and avoid overlaps between rules, new rules cannot be created, if they are too similar to existing rules (same model series, sales types, year code combinations, equipment code conditions and consumer country).

### **Filtering of Services with Respect to the Owner's Address Country**

Apart from the vehicle itself, the availability of services depends also on the country in which the vehicle's owner resides, called 'customer address country' or simply 'address country' in this document. Therefore, SOE checks for each service that is supported by the vehicle or vehicle configuration according to the service assignment rules if it is available in the customer's address country, and in case it is unavailable, removes it from the set of supported services.

### **Filtering of Mercedes connect me- and Adapter-services**

There are two classes of services; Mercedes connect me-services, which are available for Mercedes connect me-vehicles exclusively, and adapter-services, which are available for adapter-vehicles exclusively. A Mercedes connect me-vehicle is a vehicle that has a communication module. An adapter-vehicle is a vehicle that does not have a communication module but an On-board diagnostics II-adapter interface (OBD II-adapter interface). If an OBD II-adapter is plugged into the interface, a smart phone can be connected to it and used as a compensational communication module.

It is a functional requirement that adapter-services are unavailable for Mercedes connect me-vehicles from the sales perspective which SOE implements. However, as Mercedes connect me-vehicles support adapter-services according to the service assignment rules, a post-processing filter is applied to the remaining set of services. This filter guarantees that all adapter-services are filtered out for Mercedes connect me-vehicles before the final set of supported services is returned.

#### **2.3.2 Service enablement/disablement in SOE**

In SOE for each existing MBconnect Service an enablement period can be defined. Only during this period of time, the MBconnect Service is available for third party systems and therefore it is available from a sales point of view.

That means, if the start date for enabling a service is reached,

- adjacent systems are informed about the service master data changes (new Service available). That means from now on the MBconnect service is provided by the external interfaces for third party systems.
- for each vehicle, the service is available from a sales point of view: the service availability is updated and CPD will be informed (see Core concept 2.3.4 "Update of service availability").

If the end date for enabling a service is reached,

- adjacent systems are informed about the service master data changes (Service removed). That means, from now on the MBconnect service is not provided anymore by the external interfaces for third party systems.
- for each vehicle, the service is no longer available from a sales point of view, the service availability is updated and CPD will be informed (see Core concept 2.3.4 "Update of service availability").

### 2.3.3 Service contract activation

For each service that is available for a specific vehicle and user, a so called contract is concluded within SOE after the vehicle is registered. Besides some Meta information, this contract specifies the start date of the service for that specific vehicle. For services that have a limited duration, the contract also specifies the end date of that service (see Figure 6).

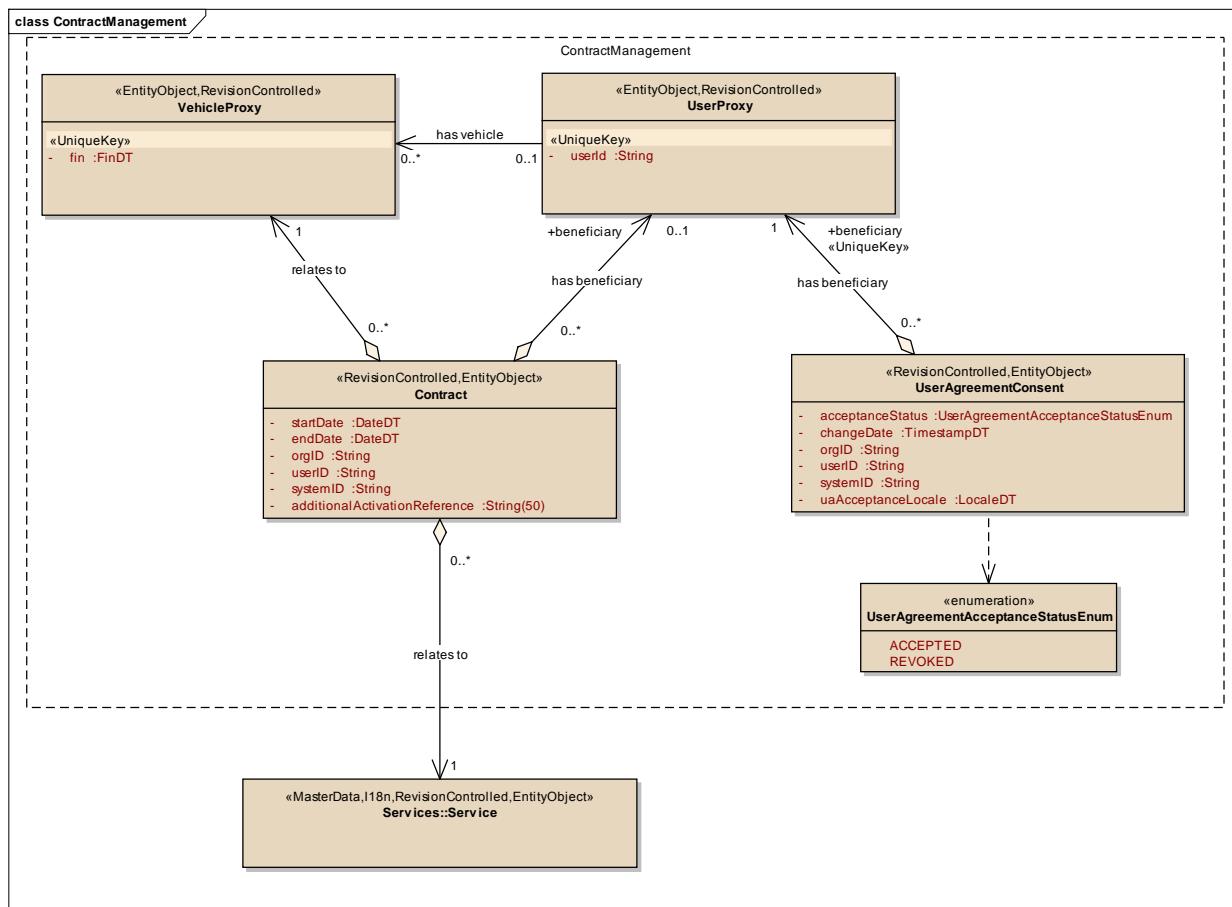


Figure 6: Interplay of vehicle, user and service

For setting the start date, SOE distinguishes two different kinds of start events. Inside SOE service master data, one can choose a contract activation trigger for a specific service. SOE differs between the two triggers “FIRST\_VEHICLE\_REGISTRATION” and “INITIAL\_SERVICE\_ACTIVATION”. Please note that in both cases the signing of a user agreement will not be taken into account for setting the start date of a service.

#### First Vehicle registration

Services with contract activation trigger “FIRST\_VEHICLE\_REGISTRATION” start with the registration of the vehicle (more precisely with the first registration date).

For doing so, adjacent systems call IF\_SOE\_InformOfVehicleRegistration. Afterwards CPD will be informed about the availability (including start and end dates) of the services for each vehicle of the user by calling IF\_CPD\_updateMBcServiceAvailability.

---

## **Initial Service Activation**

Besides the first vehicle registration, the service related contract can also start with the initial service activation. Services with contract activation trigger “INITIAL\_SERVICE\_ACTIVATION” start with the first service activation which can be individually set by each service and for each vehicle.

After the vehicle was registered by calling IF\_SOE\_InformOfVehicleRegistration, SOE will inform CPD about the service availability by calling IF\_CPD\_updateMbcServiceAvailability. Services with contract activation trigger “INITIAL\_SERVICE\_ACTIVATION” will be transferred as available but without any temporal constraints (services that have not been initially activated do not have a start or an end date).

If such a service is initially activated by adjacent systems (e.g. like DaiVB does for LiveTraffic SD), CPD will call IF\_SOE\_InformOfInitialServiceActivation with the start date of the service and a specific activation reference (more precisely called additionalActivationReference). Based on that information SOE will set the start date and – if necessary - calculate the duration of the service for that vehicle. Afterwards, CPD will be informed about the service availability (This time, the start date of the mentioned service will be set).

### **2.3.4 Update of service availability**

There is a nightly batch (see AF\_UpdateServiceAvailabilityForUsersBatch) that updates the availabilities and licenses of B2C-services for users and assigned vehicles if:

- a service has been disabled or the service availability for particular MBconnect countries has been removed,
- a service has been enabled or the service availability for particular MBconnect countries has been added
- a service assignment rule has been added/deleted/modified.

Each time an availability has been added or removed from a Customer, the batch logs this information in the SOE logfile.

### **2.3.5 Internationalization/Translation of master data**

The description of the master data maintained in SOE can be translated using the SOE dialogs. The maintainable locales (combination of language and country) can be configured in the application configuration (see PROP\_LOCALES, chapter 2.3.15).

For each language one locale can be setup as default/fallback locale. For example between the locales “German (Germany)”, “German (Switzerland)”, “German (Austria)” one can be set as default/fallback locale. So if a specific translation is identical for all three locales, it must only be maintained in the default/fallback locale.

The fallback mechanism for each text element works as follows

1. If the text in given language including country is available, this is taken
2. If the text in given language with “default marker” is available, this is taken
3. If the text in English with “default marker” is available, this is taken

- 
4. An empty text or if applicable the entity specific code is used instead of the translation.

Locale	Default	Maintained translation
German (Germany)	X	Übersetzung 1
German (Switzerland)		
German (Austria)		Übersetzung 1 AT
English (United Kingdom)	X	Translation 1
English (Switzerland)		
French (France)	X	
French (Belgium)		

Table 4: Example for default/fallback locales

Table 4 displays an example where translations for seven locales can be maintained in SOE. Three translations are actually maintained.

Locale	Value
German (Germany)	Übersetzung 1
German (Switzerland)	Übersetzung 1
German (Austria)	Übersetzung 1 AT
English (United Kingdom)	Translation 1
English (Switzerland)	Translation 1
French (France)	Translation 1
French (Belgium)	Translation 1

Table 5: Example retrieving translations

Table 5 shows the translations for specific locale settings regarding the default settings in Table 4 (German (Switzerland) has no maintained translation, so the German default value will be shown).

Notes:

- This does only apply for translated descriptions on interfaces and the SOE GUI.
- In the case of document creation an error will always occur on missing translations.
- There will be NO import or export of translations, such as Excel or CSV.

### 2.3.6 Functional Logging/Monitoring

SOE needs a functional logging/monitoring for asynchronous tasks that run unattended by a user. In general, this relates to the batches that are defined.

Besides just writing into the application's log, the following conditions must be monitored in such a way that issues or problems do not remain undetected:

- Errors that occur during a run of a batch

- 
- E-mails that return because a recipient is not known. (This situation is special as the mail reply indicating a wrong address comes asynchronous as an email to the sender's address.)
  - Letters that return to sender (through letter mail). (This is even more special because it cannot be monitored entirely automatic.)

It is sufficient if these problems are detected by the means of monitoring and are sent via mail to a pre-configured recipient that will have to take measures to solve an issue.

### **2.3.7 Administrative Logging**

All administrative activities regarding the change of the dynamic data will be logged in the application's log.

### **2.3.8 Concurrent user access**

Whenever two or more users modify the same set of data within SOE, the changes of the user who first saves the data will be accepted (principle of optimistic locking).

For data that is being modified inside a change session a locking mechanism is implemented such that this data cannot be concurrently modified by other users.

It can happen though, that master data for which modifications are only allowed inside a change session, to be opened in edit mode by two users in parallel. Because the locking mechanism is applied only when saving these modifications, the changes done by the user who first saves the data are accepted. In case a collision appears the following error message shall be displayed for the second user:

Code: OPTLOC\_002

Message: "The data you have been working on is already inside the change session of user: <user>. All your changes will be discarded."

#### **Analysis regarding User Interface**

All user interfaces of SOE will only be used for administration purposes and so collisions are very unlikely.

#### **Analysis regarding External Interfaces**

All external services offered by SOE, which potentially modify data within SOE, are customer and vehicle specific. It is very unlikely that two retailers work on the same customer or vehicle data. Thus collisions are very unlikely from this side, as well.

In case an optimistic locking collision is detected by SOE, the following error message shall be displayed:

Code: OPTLOC\_001

Message: "The data you have been working on concurrently was modified by another user. Please refresh your data and try again."

---

### **2.3.9 Historization of Dynamic Data (Revision Control)**

Dynamic data in SOE is historized, i.e. not only current version is available in the system but also all previous versions. Entities with that property are marked as "Revision Controlled".

Whenever one of the attributes stored in such an entity is changed, a new revision is created. All revisions are marked with timestamps, so that it is possible to determine which revision was valid at a certain point in time. Thus it is possible to restore all previous versions if needed. Historic data must be kept in the system for 10 years (legal requirement).

However, each entity is subject to revision control on its own. That means not an entire group or tree of entities is considered as a revision, but every individual entity on its own. This provides no limitation towards the ability to restore historic versions of entire entity trees. (E.g. it is still possible to restore all contracts a user had on Feb, 28<sup>th</sup> 2012.)

Anyway, SOE does not provide any built in functionality to restore or display historic data, because there is no requirement for that. It only ensures that that the legal requirement to store it is fulfilled. If necessary, historic data has to be accessed directly in on storage level (e.g. data base).

### **2.3.10 Master Data Consistency Checks**

Certain master data types maintained in SOE are related to each other, and therefore system validation is needed to ensure that the master data remains consistent.

The interdependency of these master data types is defined by relationships:

For example: a <service> is assigned to a <UserAgreement> through the <UserAgreementServiceAssignment> entity.

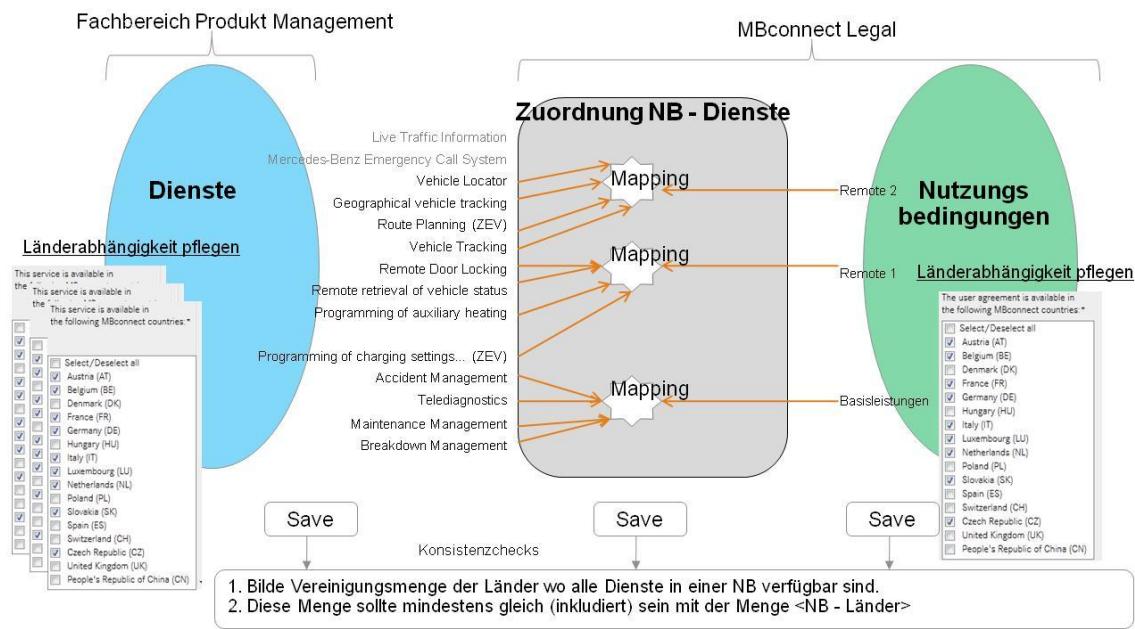
Because both entities are associated to a list of MBconnect Countries (this association indicates their availability in those countries) this associations need to be kept synchronized.

The following "Availability in Countries" synchronization check is introduced for the entities <Service>, <UserAgreement>, <UserAgreementServiceAssignment>: the list of countries where a service is available must include the list of countries where the user agreement (covering the service) is available in.

By defining an <include relationship> it is ensured, that a service can be extended to other countries for which the availability of a user agreement is not yet configured.

SOE implements this synchronization rule as a system validation check in the following scenarios:

- When creating/updating a <Service> (see AF\_SaveService)
- When creating/updating the mapping between a Service and a UserAgreement: <UserAgreementServiceAssignment> (see AF\_SaveUserAgreementServiceAssignment)
- When creating/updating a <UserAgreement> (see When creating/updating a <Service> (see AF\_SaveService))



In case that the consistency check(s) fail(s), SOE displays an appropriate message together with a hint to be followed in order to solve the master data inconsistency problem.

Note: when the updates performed by the department "MBconnect Product Management" fail, the department "MBconnect Legal" needs to be contacted or vice versa. Usually the contacted department needs to perform a prior update on the master data in their ownership.

The following two scenarios are described, to give a glimpse, of how the master data update process concerning the country availability for services and user agreements should be designed. The update process of the master data in SOE required a precise sequence of steps to be followed:

**Scenario#1:** For example, when a service is introduced to a new country the following steps need to be performed in the given sequence:

- Step#1: the department "MBconnect Product Management" updates the country availability of the <Service masterdata>
- Step#2: the department "MBconnect Product Management" informs the department "MBconnect Legal" about changes to <UserAgreement masterdata> and/or to < UserAgreementServiceAssignment masterdata>
- Step#3: the department <MBconnect Legal> updates the <UserAgreement masterdata> and/or < UserAgreementServiceAssignment masterdata>).

**Scenario#2:** When a service is withdrawn from a country the following steps need to be performed in this sequence (*Please note that the feature of withdrawing a service from a country is specified in the upcoming SOE SRS#025, and is not supported currently*)

- Step#1: the department "MBconnect Legal" updates the <user agreement content> and the <UserAgreementServiceAssignment> such that the service is no longer supported in the country
  - Step#2: when the user agreement that covered the service becomes

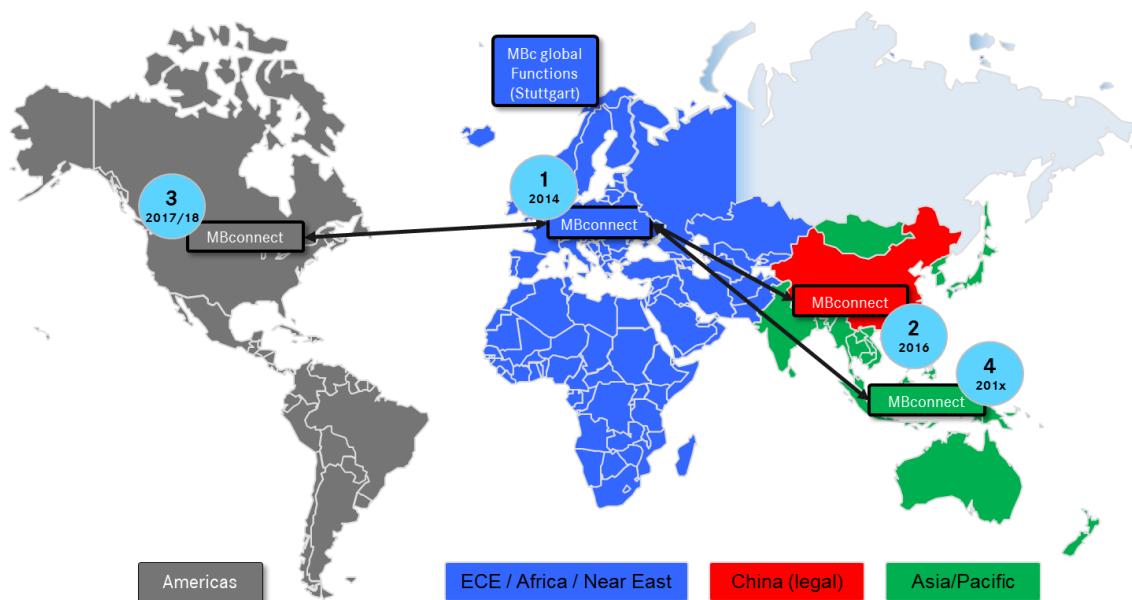
- obsolete, the department “MBconnect Legal” informs the department “MBconnect Product Management” that the availability of the service can be withdrawn from that country
- Step#3: the department <MBconnect Product Management> updates the country availability of the <Service masterdata>

### 2.3.11 Operation Mode

The SOE is able to be operated in different operation modes, which can be configured via the attribute SOE\_OPERATION\_MODE (→ see chapter 2.3.15). There is one mode for operating as the SOE “MDM” (maintenance of master data) and one mode for operating as SOE “REGION”.

With the rollout of MBconnect to other regions, five SOE instances are set up: One instance with operation mode “MDM”, and four instances with operation mode “REGION”:

- The four region instances are local instances storing regionally specific dynamic data, e.g. contracts signed by customers. There is one SOE instance of this kind for each of the four MBconnect regions Europe and Africa, the Americas, China, as well as Asia-Pacific (cf. Figure 7). Additionally, each of these regional SOE instances retains a local copy of the complete set of master data.



Quelle: RD/RTA

Figure 7: The four MBconnect regions

- The fifth SOE installation, SOE MDM, is a master instance for maintenance of master data and does not store any dynamic data. Users of SOE MDM have the right to perform change operations on master data accessible via the SOE user interface.

---

A change operation on master data in SOE MDM triggers master updates in all regional SOE instances. I.e., the affected master data is exported by SOE MDM, transmitted via the internet and imported automatically in each regional SOE instance.

Afterwards, adjacent systems of that region, e.g. CPD, are informed of the master data change independently in each region (cf. Figure 7), which is described in Informing adjacent systems about master data changes (→ see chapter 2.3.12).

Note: Master data which is not maintained via the SOE user interface, e.g. context management data, is updated as part of SOE software updates and not transmitted via the data replication mechanism.

Therefore the dialog frame, dialog flow and application functions (e.g. batches) differ between the MDM instance and the regional instances, too.

These differences are described in UI Entitlements and Function Entitlements (→ see chapter User / Organization / Entitlements).

### **2.3.11.1 Supply SOE Regions with changed master data**

Because changes to master data in the SOE MDM have also been available to SOE Regions, the notifications of master data changes are triggered:

- when the user releases a change session (see chapter 2.3.14 Core Concept “Change Session”):  
Each component must summarize master data changes at a master data entity level, which is determined by a MasterDataReplicationTypeEnum.
- when the user pushes a button in the admin dialog, which leads to a full synchronization:  
IIF\_TriggerReplications is called with an empty input parameter.
- or after an AMDS-Import batch finished successful.

A detailed overview about the replication trigger with and the export/ import of changed master is given in Figure 8.

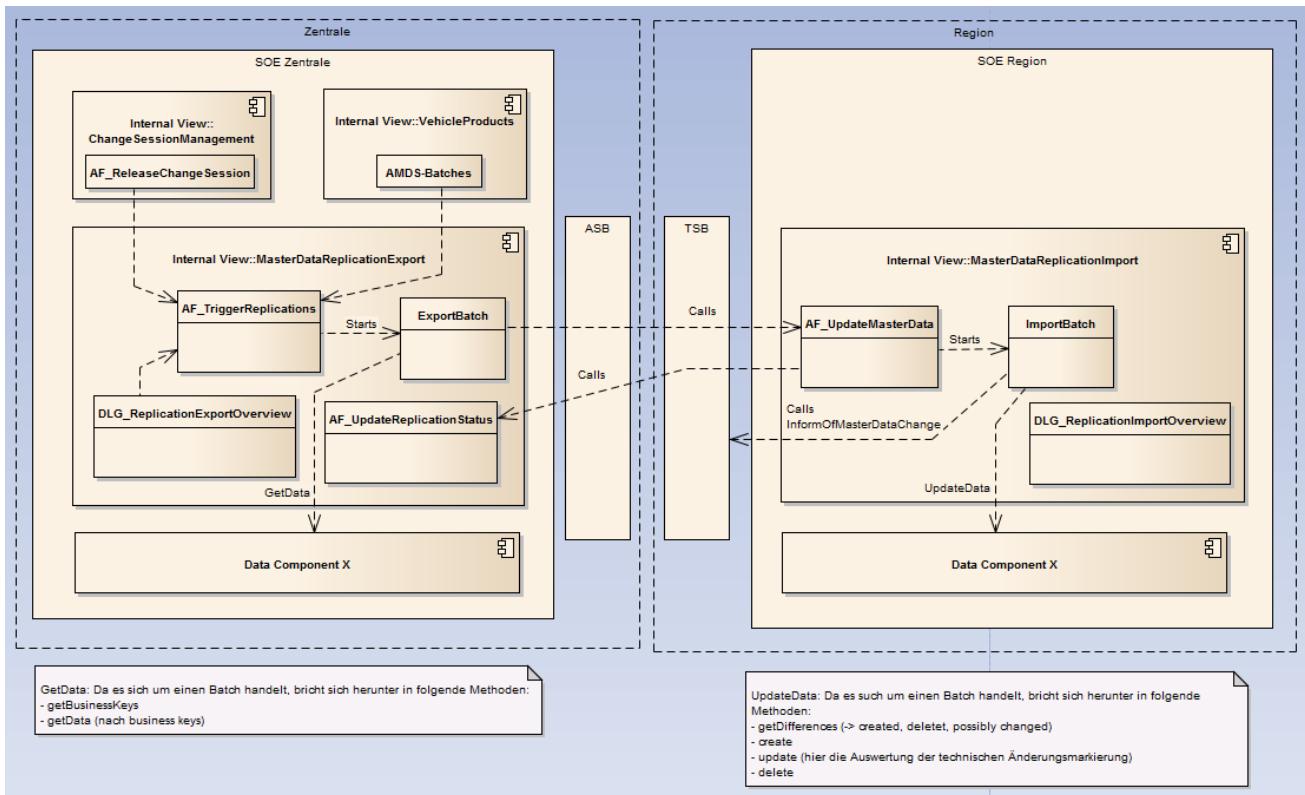


Figure 8: Replication of master data

**Exception:** The master data of the component “Account Data Support” doesn’t have associated UI-Dialogs and modifications upon this data are not restricted to be performed inside a replication from SOE MDM to SOE Regions.

### 2.3.12 Informing adjacent systems about master data changes

Because a SOE Region relevant master data modifications also from a semantic point of view, the notifications for each type of master data changes are triggered when the SOE Region updated the replication relevant master data.

*Note: As an exception the country fields master data is imported via database scripts and is not maintainable via the SOE GUI. To inform adjacent systems about changes regarding this data type, the interface IIF\_InformOfMasterDataChange can be triggered manually by using the dialog DLG\_InformOfMasterDataChanges.*

The master data changes are summarized at this level:

- **MasterDataResourceTypeEnum:** indicates the component for which master data has changed
- **MasterDataDataTypeEnum:** indicates the master data data type that has changed
- **Sub Set:** for each master data data type an optional attribute has been defined that will be used by the adjacent systems to assess the impact of the updated master data.

The table below summarizes the information published by the interface IF\_TSB\_InformOfMasterDataChange when master data has changed:

<b>MasterDataResourceTypeEnum</b>	<b>MasterDataDataTypeEnum</b>	<b>Subset</b>
SERVICES	AUTHORIZATIONDATA	
	SERVICEMASTERDATA	
	SERVICECATEGORIES	
	SERVICEFIELDS	
	VEHICLEMARKETMASTERDATA	
	VEHICLECOUNTRYMASTERDATA	
MBCONNECTCOUNTRIES	MBCCOUNTRYMASTERDATA	
ACCOUNTDATASUPPORT	COUNTRYFIELDS	
DOCUMENTS	USERAGREEMENTSERVICE	
	ASSIGNMENT	
	USERAGREEMENTMASTERDATA	

Table 6: Overview master data changes relevant to adjacent systems

*Note: The notifications are sent grouped by master data data type for which changes has been detected.*

The adjacent system will need to decide:

- if it is interested in the change: based on the “MasterDataDataTypeEnum”
- if the change is relevant: Based on the “Subset”

A button in the admin dialog is given, which triggers the InformOfMasterDataChange for all existing combinations of MasterDataResourceTypeEnum and MasterDataDataTypeEnum with an empty Subset, to inform the adjacent system about master data changes.

### 2.3.13 Testing Service Master Data

#### Documents: Preview function

In the SOE the function is given to get a preview of maintained documents. This function has the possibility to create a PDF file as a preview file for each document. It is necessary to specify conditions like the country and the language of the document to create a preview file, which will be created with the selected conditions.

#### Service Assignment Rules: Test simulation dialog

A user has the possibility to visualize and test service master data with different criteria for a specific vehicle in the SOE.

This possibility will improve the

- Support processes: If there is any problem in another system, which uses service assignment rules (-> Support Ticket), the user is able to test if the problem is in the SOE (source), so if a service assignment rule is defined incorrect for example.
- Maintenance of the rules: The user will be able to test (newly created) service assignment rules before releasing them to the productive system.
- Bug analysis and testing after bug fixing: If a bug is discovered, the user will be able to retest the service assignment rules after the bug was fixed and only after that, release the correct data to the productive system.

---

## **Service Master Data: Maintenance dialog**

While maintaining service master data, the possibility is given to get an overview about the existing and the new created service master data in SOE. So it is possible to check the input data immediately after creation or after a change.

### **2.3.14 Change Session**

The change session is needed each time a creation, modification or deletion of the following master data is done:

- vehicle related master data
- document element related master data
- service master data
- country master data

All master data in SOE is under the control of a change session.

A master data element can exist at a time in both forms: once inside a change session (the form that is being modified or the “changed form”) and the “productive form” (the form that has been released in production).

The change session concept has the following characteristics:

1. Inside the change session an indefinite number of elements can be created, edited or deleted.
2. The change session needs to be opened and released manually.
3. Only one change session per user is allowed.
4. The element becomes automatically part of the user’s change session when the user either saves the modification, when the user creates a new element or a new version of the element or when the user deletes an element.
5. When a user edits a master data element, this is locked for other users. Locked elements are visible for other users only in their productive form.  
If no productive form available (master data element is being created) then other users won’t see the element. If a productive form exists for a locked element, then other users have no possibility to edit the content of the element. All dialogs of locked elements will be opened in read-only mode.

A change session is comparable to a sandbox: When a user works inside a change session he will be able to see:

- the productive master data (the productive form) of the entities he is not editing and
- the modified master data he is currently adding or modifying.

All entities that need a change session are linked to it via the entity called ChangeOperation.

Based on this link the system can differentiate between productive and changed forms of master data. This entity also stores the type of the modification done inside a change session: whether the master data has been created, updated or deleted.

### 2.3.14.1 Manage change sessions: “Change Session Management”

When a user wants to release a change session he has the following possibilities:

- Release (commit) all changes: all modifications done inside the change session will be enabled for productive use.
- partially discard changes: the changes inside a change session are traced at element level. The user can decide to discard individual unwanted changes.
- discard the change session: all elements together with the change session will be discarded.

#### 2.3.14.1.1 Releasing a change session

When a change session is released, each component will be triggered to release their master data elements. Each component that works with a change session needs to implement this feature.

The figure below describes the steps that need to be performed by each component for which elements have been modified.

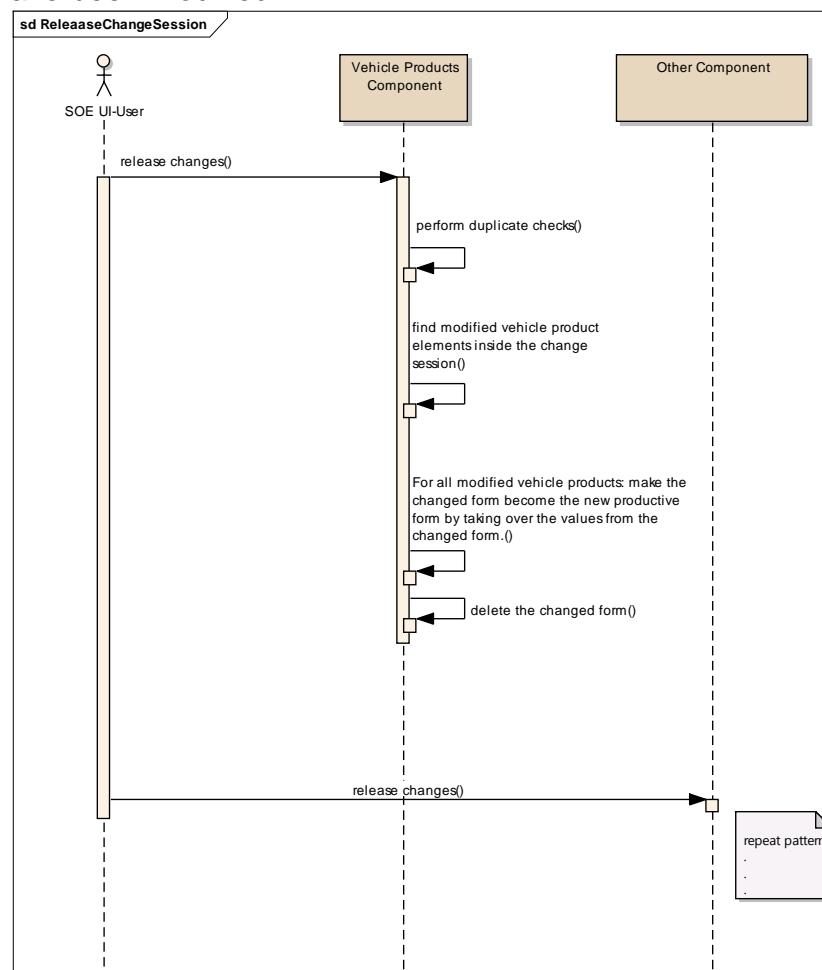


Figure 9: releasing a change session

Each component that works with a change session needs to implement this feature.

The behaviour is generic:

- **Loop for all changed elements**
  - Perform duplicate checks:

Check if the element to be released already exists as a released form, based on the business key that is specific to each element. E.g. For a model series to be released, check to see if there already exists the given model series based on the attribute ModelSeries.ModelSeriesID.

In case a duplicate has been identified, the releasing of the change session will be aborted with an error containing the first identified duplicate.

- If an element has been modified then:
  - o Make the changed form become the new productive form by taking over the values from the changed form.
  - o Delete the changed form.
- If the element has been created:
  - o Create the productive form from the changed form and delete the changed form.
- If the vehicle product element has been marked for deletion:
  - o Delete the productive and the changed form.
- **Delete the change session.**

Independently of the change operation (either ChangeOperation.NEW, ChangeOperation.UPDATE, ChangeOperation.DELETE) the changed form won't be under the revision control. Only when a change session is released, the revision control information of the productive form will be updated.

### 2.3.14.1.2 Discard modifications of a change session

The modifications done inside a change session can be discarded.

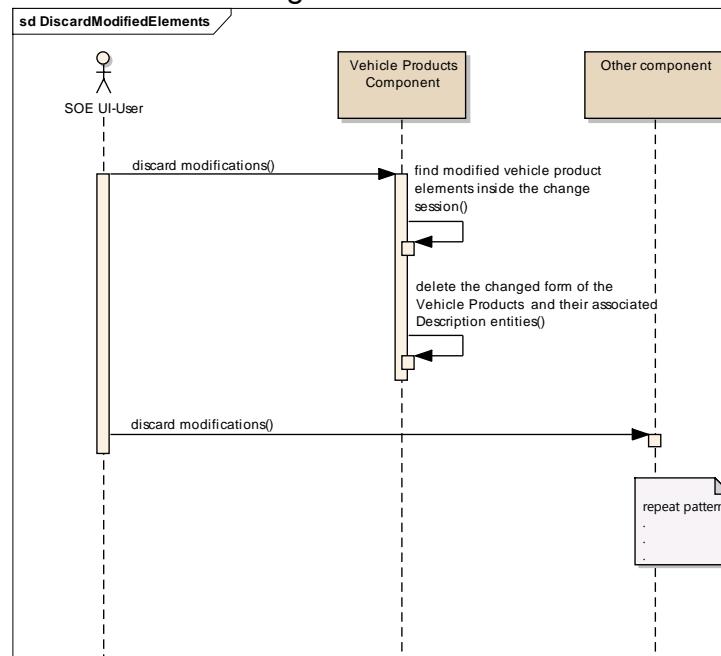


Figure 10: Discard modifications inside change session

Each component that works with a change session needs to implement this feature.  
The behaviour is generic:

- Fetch all elements that are inside the change session of the user
- Loop for each element:
  - o If the element has a productive form then delete the changed form and all the changed forms of the associated elements that have been created inside the change session.
  - o Else delete the changed form of the element.

#### 2.3.14.1.3 Delete a change session

When a change session is deleted, the following actions are triggered:

- Discard all modifications done inside the change session (see previous subchapter)
- Delete the change session

#### 2.3.14.1.4 Taking over a change session

It is possible for a user to take over changes done by another user, by acquiring his change session. A user doesn't need special permissions in order to use this feature.  
Each component that works with a change session needs to implement this feature.

The behaviour is generic:

- all master data elements that are in the change session of the other user are brought into the change session of the current logged on user.
- the change session of the other user is deleted

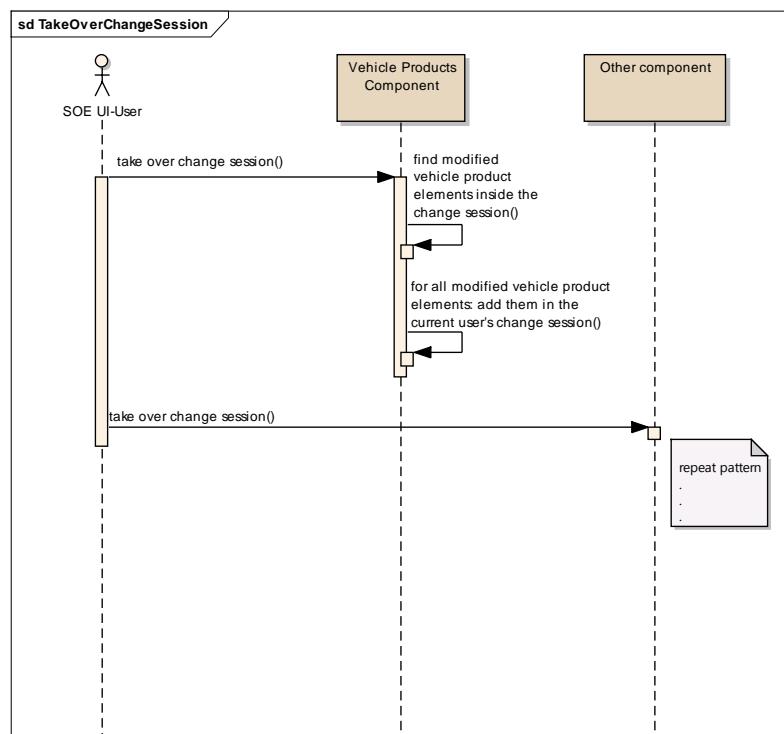


Figure 11: Taking over a change session

### 2.3.14.1.5 Displaying all modifications of a change session

The modifications done inside a change session are collected from each component. Each component that works with a change session needs to implement this feature. The behaviour is generic:

- Load all elements that have been modified during the change session
- Extract from the modified elements the following attributes (if available):
  - o element type
  - o element name
  - o further attributes if available in the business key of the modified element

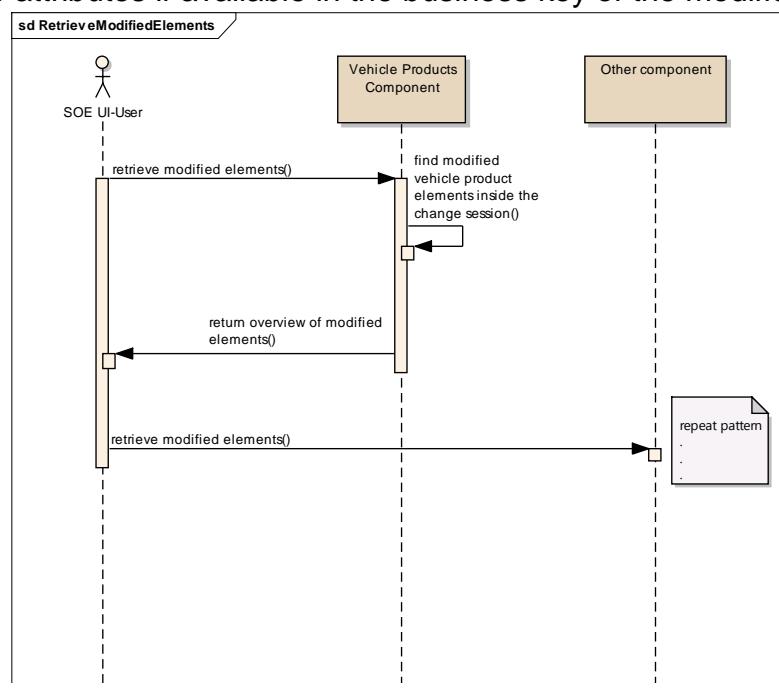


Figure 12: Retrieve information about what modifications have been done

### 2.3.14.2 Loading of master data elements

#### Master Data accessible to a UI-User

The UI-User will be able to see only:

- master data in its productive form (even if another user is modifying the master data inside his change session)
- master data that is currently being added/modified by him
- 

*Note: master data that is being created by other users in their change sessions will not be accessible for the UI- user.*

All initialization function of the dialogs that load the following master data element types:

- Document, VersionedDocument
- CustomTag
- DocumentBlock, VersionedDocumentBlock
- DocumentTemplate, VersionedDocumentTemplate
- UserAgreementServiceAssignment
- ModelSeries
- SalesType, SalesTypeDescription

- BodyType, BodyTypeDescription
- Equipment, EquipmentDescription
- Service, ServiceMaster, ServiceAssignmentRule
- GenericMasterData
- MbcCountry
- ResourceAuthorization

need to be extended by a mechanism for fetching master data elements for the logged on user.

The mechanism works like this:

- For master data elements that are inside the change sessions of other users: return the productive form (if available).
- For master data elements that are inside the change session of the user for which the master data is loaded: return the not productive form.
- For all other elements return the only available form: their productive form.  
Additionally for each element two flags are returned:
  - flag indicating whether the element is currently being edited inside a change session.
  - flag indicating whether the change session in which the element is being edited belongs to the logged on user.

These flags will be evaluated for example inside the “init”-functions of the above dialogs.

### **Master Data accessible by SOE Clients via external interfaces**

SOE Clients will be always provided with the productive form of the master data. For the master data that doesn't need a change session, all changes will be available to other systems immediately after saving them in SOE.

#### **2.3.14.3 Working with master data elements inside a user's change session**

An element becomes automatically part of the user's change session when:

- A new element or a new version of the element is created
- An existing element is modified
- An element is deleted.

Each component that works with a change session needs to implement this “apply session” on master data element feature.

The behaviour is generic and will be called ApplyChangeSessionOnElement (it will be later referenced by all application functions that either SAVE, CREATE or DELETE master data).

#### **Step 1: If the element is being modified then check if it is already inside a change session**

- If the element is *already inside the change session of another user*: Discard the modification of the element. Depending on the element type throw this error:

Element Type	Associated Error
--------------	------------------

Element Type	Associated Error
ModelSeries	VEHPROSESSION_001
SalesType	
BodyType	
Equipment	
Document	DOCSESSION_001
DocumentTemplate	
DocumentBlock	
CustomTag	
UserAgreementServiceAssignment	
Service	SRVSESSION_001
ServiceAssignmentRule	SRVSESSION_002
ServiceMaster	SRVSESSION_003
MbcCountry	MBCSESSION_001

Execution stops here.

### Step2: Determine the associated elements of the modified element

- Retrieve all associated elements of the modified element

Parent Element	Associated Element
ModelSeries	-
SalesType	List of SalesTypeDescription
BodyType	List of BodyTypeDescription
Equipment	List of EquipmentDescription
Document	VersionedDocument
DocumentTemplate	VersionedDocumentTemplate
DocumentBlock	VersionedDocumentBlock
UserAgreementServiceAssignment	-
CustomTag	-
Service	-
ServiceAssignmentRule	-
ServiceMaster	AbstractProfileDataFieldRelationship
ResourceAuthorization,	-
MbcCountry	-

Note: the association works in both directions!

e.g.

- if the associated VersionedDocumentElement is modified also the parent element (Document) is considered modified.
- If the name of the document is being modified, all associated VersionedDocumentElements are considered modified.

### Step 3: Bring the element and its associated elements in the change session:

- If the element is already inside the change session of the logged on user then save modifications directly on the changed form

- Else copy the modifications done on the element together with all associated elements into new entities. and add these entities to the user's change sessions.

The following figure depicts what happens **when a master data element** (e.g. ModelSeries) **is created** inside the change session of the user:

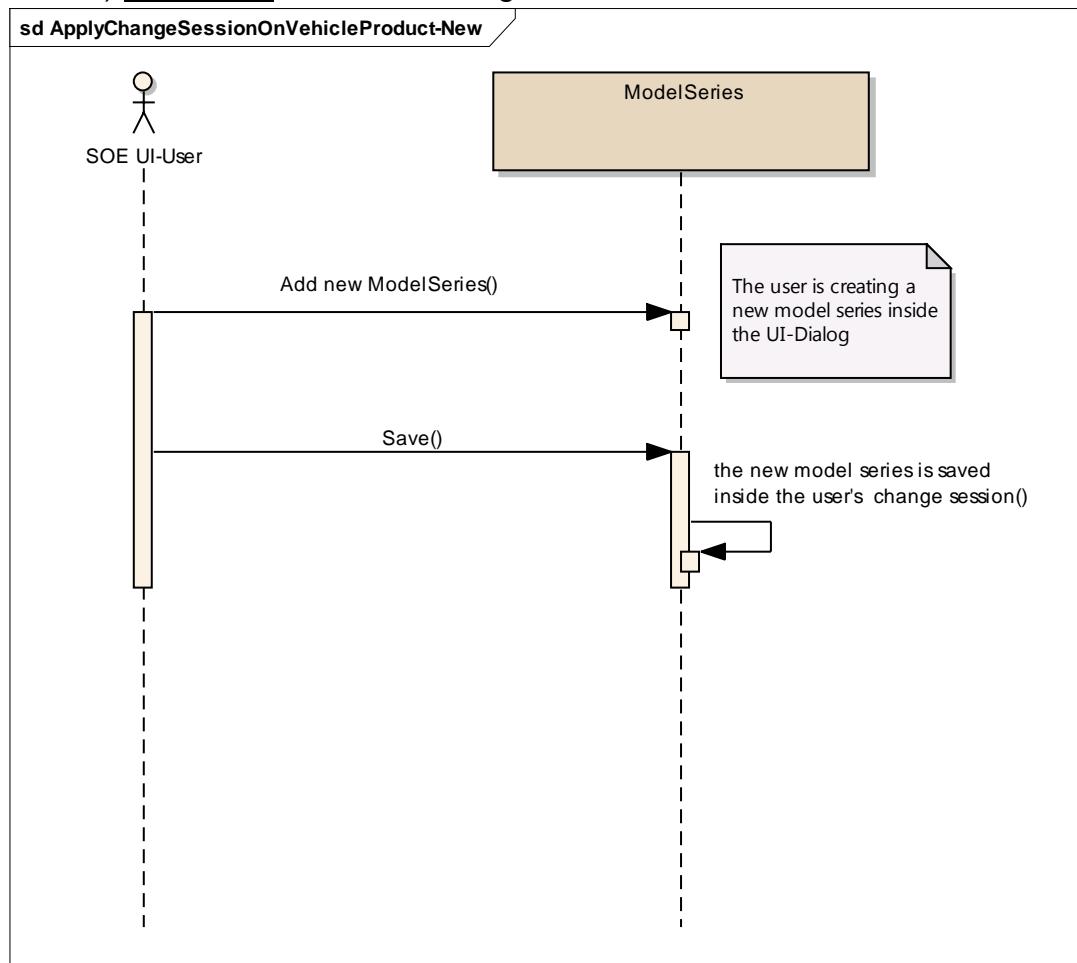


Figure 13: creating a new model series

When the change session is released, then create the productive form from the changed form. Then delete the changed form.

The following figure depicts what happens **when a master data element** (e.g. Equipment) **is modified** inside the change session of the user:

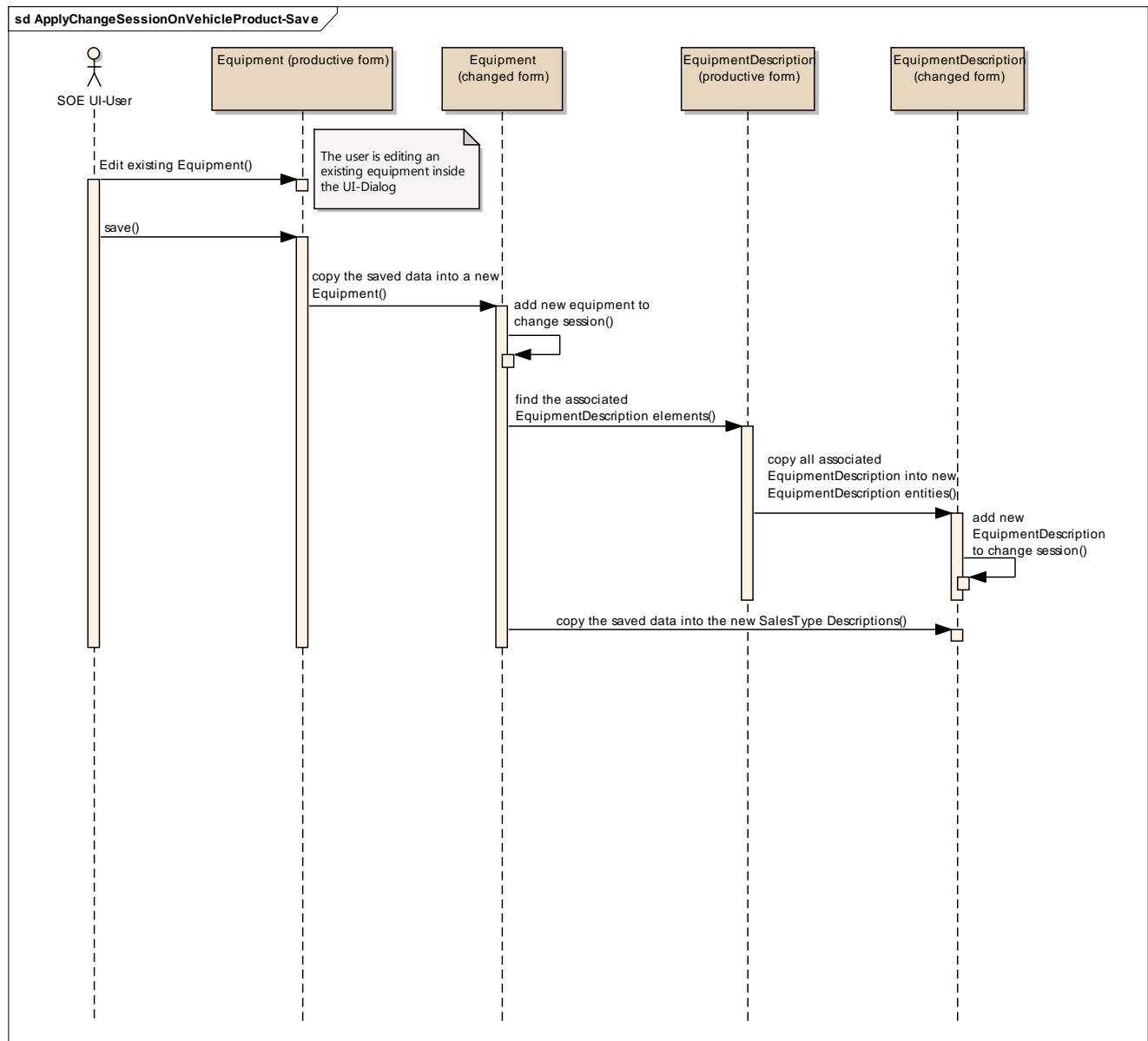


Figure 14: modifying existing equipment

When the change session is released, then make the changed form become the new productive form by taking over the values from the changed form. The changed form is deleted.

The following figure depicts what happens **when a master data element** (e.g. Equipment) **is deleted** inside the change session of the user:

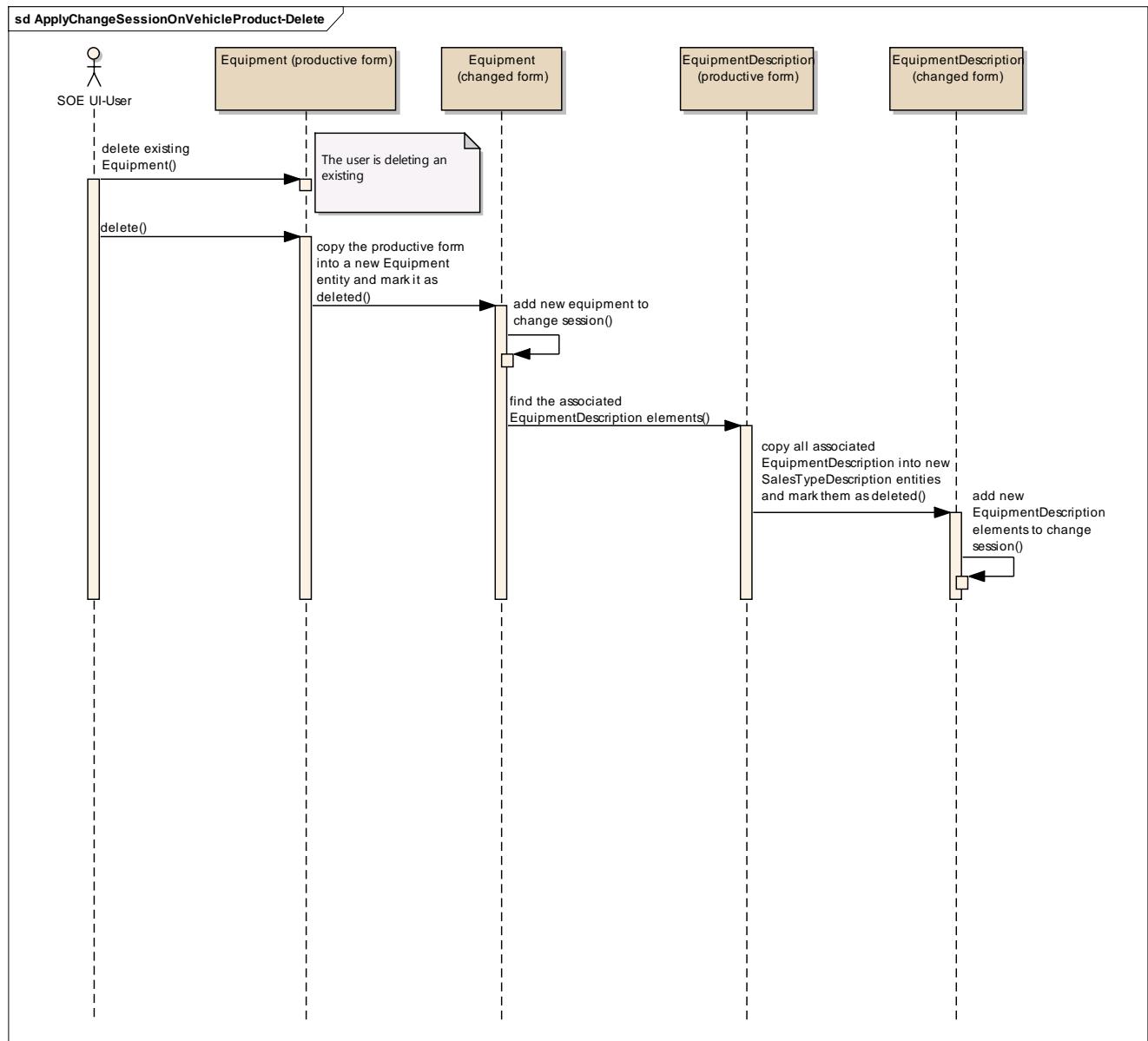


Figure 15: deleting existing equipment

When the change session is released, the productive and the changed form of the element will be deleted.

### 2.3.15 Application Configuration

Several parts of SOE are configurable and can be modified without changing the functional logic of the system. These parameters are referred to throughout this document as “application configuration”. To be exact, the application configuration supports the following properties:

Property	Description
PROP_OPERATION_MODE	Determines the operation mode of a SOE. The operation mode SOE is operated in two possible modes: “MDM”: SOE operates as a system for master data management “Region”: SOE operates as a system offering services for adjacent systems

	and generating regional dynamic data within a regional instance of the systems landscape
PROP_REGIONID	Determines the unique ID of a SOE Region, if the PROP_OPERATION_MODUS = REGION. For example "CN" for the Chinese SOE Region.
PROP_REGIONS_TO_TRIGGER	Lists all existing PROP_REGIONIDs for PROP_OPERATION_MODUS = MDM.
PROP_MARKETS	Lists all supported markets by the system. The stored information contains the ISO code (ISO 3166 Alpha 2) for each market and the market number (VBET) used by UVS and AMDS.
PROP_LOCALES	Lists all supported locales (e.g. "de_DE"). Each locale contains the ISO codes of the language (ISO 639-1) and the country (ISO 3166 Alpha 2), the used font style and the information whether the locale is considered as the default locale for its language. Each translation is associated to a specific locale.
PROP_LOCALES_FALLBACK	Lists the main languages, i.e. those which are printed in bold font on the translation dialogs.
PROP_LOCALES_SUPER_FALLBACK	Contains the super fallback language which is used in case a requested language could not be found in the PROP_LOCALES and the PROP_LOCALES_FALLBACK.
PROP_GUI_TRANSLATION	Contains the translation table for the GUI elements.
PROP_PAGING_ITEMS	Configures the "selectable items per page" field for paging (e.g. "10, 20, 50, 100").
PROP_EMAIL_CONFIG	Configures the e-mail sender's email-address.
PROP_EMAIL_SENDER_NAME	Configures the name of the email sender.
PROP_CONTINENT	List of continents which are used to call the DRD to retrieve the markets within the continent.
PROP_AMDS_SALESTYPE_GROUPCLASSES	List of sales type group classes that are retrieved from AMDS and relevant for SOE. Currently, this list contains the sales type group class "BR" providing model series, and the class "BODY" providing body types. Note: Not configurable in the implementation because special source code is needed to work on the specific types.
PROP_AMDS_MBCONNECT_INCAPABLE_MODELSERIES	Lists all model series that are not MBconnect capable.
PROP_PREVIEW_DATA	Stores preview data for all enumeration values of the PrintableDataFields enumeration. This values will appear on the document (e.g. user agreement) selected for previewing the effect of document-elements modifications done inside a change session.
PRINT_SERVER_OUTBOUND	Stores the path to the directory on the file server of the print service supplier. At this path, the print server supplier will upload CSV files containing information of the processed PDF documents.
PRINT_SERVER_INBOUND	Stores the path to the directory on the file server of the print service supplier. At this path, SOE uploads PDF documents that need to be printed and delivered as a letter to the customer.
PRINT_SERVER_ARCHIVE	Stores the path to the directory on the file server of the print service supplier. At this path, the print service provider copies the PDF documents that have been printed.
PRINT_SERVER_CSVARCHIVE	Stores the processed CSV-Files.

PROCESS LETTER DURATION	Stores the processing duration of a daily letter workload: from printing to sending it to a customer
PROP_DLG_RULES_MAX_ELEMENTS	Tells how many items of one type (e.g. Services) to display for a service assignment rule in the overview dialog.
PROP_DLG_LANGUAGES	Defines the available languages for all SOE dialogs.
PROP_LOCALE_FOR_ADDRESS_COUNTRY	Defines the language of the address country when sending letters. E.g. when sending them from Germany, the address countries need to be printed in German language to be properly recognizable by the postal service companies.
PROP_EMAIL_FUNC_SUPPORT_PORT_ADDRESS	The email address of the functional SOE support.
PROP_EMAIL_FUNC_SUPPORT_PORT_NAME	The email address long name of the functional SOE support.
PROP_EMAIL_FUNC_SUPPORT_PORT SUBJECT	The email subject when informing the functional support about new documents which are ready to be sent to customers and need approval.
PROP_EMAIL_FUNC_SUPPORT_PORT_TEXT	The email text when informing the functional support about new documents which are ready to be sent to customers and need approval.

Table 7: Properties of Application Configuration

SOE's application configuration does not need to be changeable through UIs. It is sufficient to make changes by scripts or changes of files. Please note, that there is a differentiation between static data/master data which can be directly changed through the means of the system (by dialogs) and the application configuration.

There is no functional requirement that changes on the application configuration take effect at runtime without a server restart or redeploy as long as it is not explicitly stated for a specific configurable element.

In addition to the simple configuration values above, there is also a more complex configuration for country specific fields (see 2.3.19 / Configuration of Profile Data Fields).

### 2.3.16 Operational Aspects

For the operations (=IT Betrieb) the SOE system provides a status page which contains operational information about the SOE version and build information.

For the PDF-Documents that couldn't be processed by the print service provider, a <TimeStamp>\_error.csv file will be written and stored on the output directory (PRINT\_SERVER\_OUTBOUND). The file contains the list of reference numbers (associated to the PDF-Documents for which the processing has failed). The semicolon ";" is used as a delimiter.

The outbound directory is monitored by CA Wily Introscope®.

When a new <TimeStamp>\_error.csv file is written, the content of the file (if it is not empty) is sent by Wily via an email to a predefined SOE User.

### 2.3.17 Documents

In order to be able to create, print and hand over documents to the customer (e.g. user agreements), the documents need to be maintained first. This chapter describes how documents are structured and how they are generated. Emails are maintained in the same way documents are. An example of how a generated document filled with sample

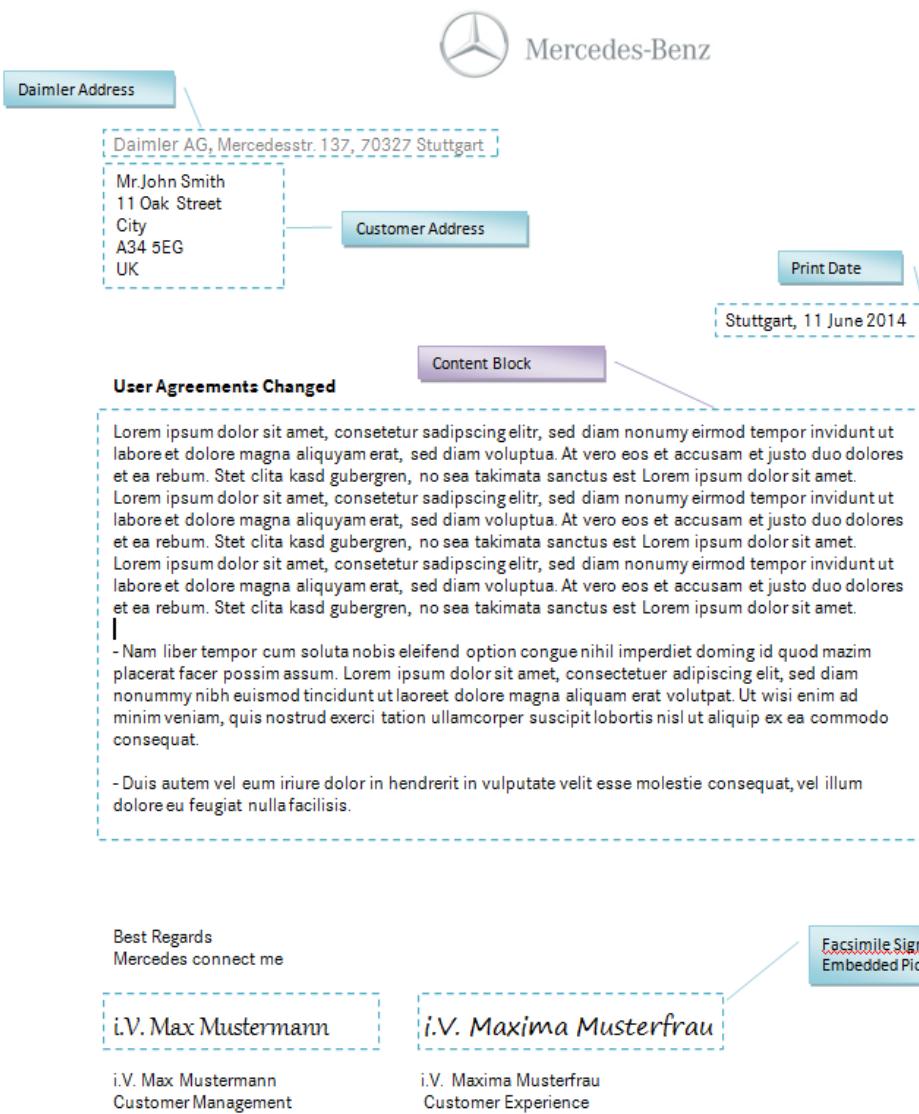
---

data may look like can be found in Figure 16: Example layout for a cover letter. Such a document is not static; instead it is built dynamically together when requested.

Refer to Document Generation (-> see chapter 2.3.17.6) to see a more detailed description of how documents are generated.

Following terms are used within the document chapter and are mentioned here to allow an easier understanding while reading through the chapter:

- Document Trigger – A trigger represents a certain event that occurred (e.g. customer signed a user agreement) and specifies which documents need to be sent out to inform the customer
- Document Template – The document template is a PDF that is primarily used as a background onto which content is rendered. A document template can also be used as a static document, e.g. the terms and conditions, that is attached as an additional page to the main that is generated
- Data Fields – Data fields are XML tags which are mapped to dynamic content. E.g the data field “firstName” would represent the first name of a customer
- Document Definition – Specifies how the document is structured
- Document Block – Specifies the localized content that is rendered onto the document template
- Custom Tag – Specifies a country specific structure of data fields
- Document Image – the image is embedded inside the document. Supported formats: jpg, tif, gif, bmp, png wmf.
- Font Style – Specifies the font style that is used to display the text on a document



Daimler AG, Sitz und Registergericht Stuttgart, HRB Nr. 19360, Vorsitzender des Aufsichtsrates: Manfred Bischoff  
Vorstand: Dieter Zetsche, Vorsitzender; Günter Pflug, Rüdiger Grube, Andreas Renschler, Rodo Uebber, Thomas Weber  
Deutsche Bank AG, Stuttgart  
BLZ 600 700 70, Kto.: 1 116 656  
IBAN: DE58 6007 0070 0111 6656 00  
BIC (S.W.I.F.T.-CODE): DEUT DE FF 655

Dresdner Bank AG, Stuttgart  
BLZ 600 800 00, Kto.: 9 015 000 00  
IBAN: DE 19 6008 0000 0091 5200 00  
BIC (S.W.I.F.T.-CODE): DRES DE FF 600

Commerzbank AG, Stuttgart  
BLZ 600 400 71, Kto.: 5 303 300  
IBAN: DE36 6004 0071 0530 3300 01  
BIC (S.W.I.F.T.-CODE): COBA DE FF 600

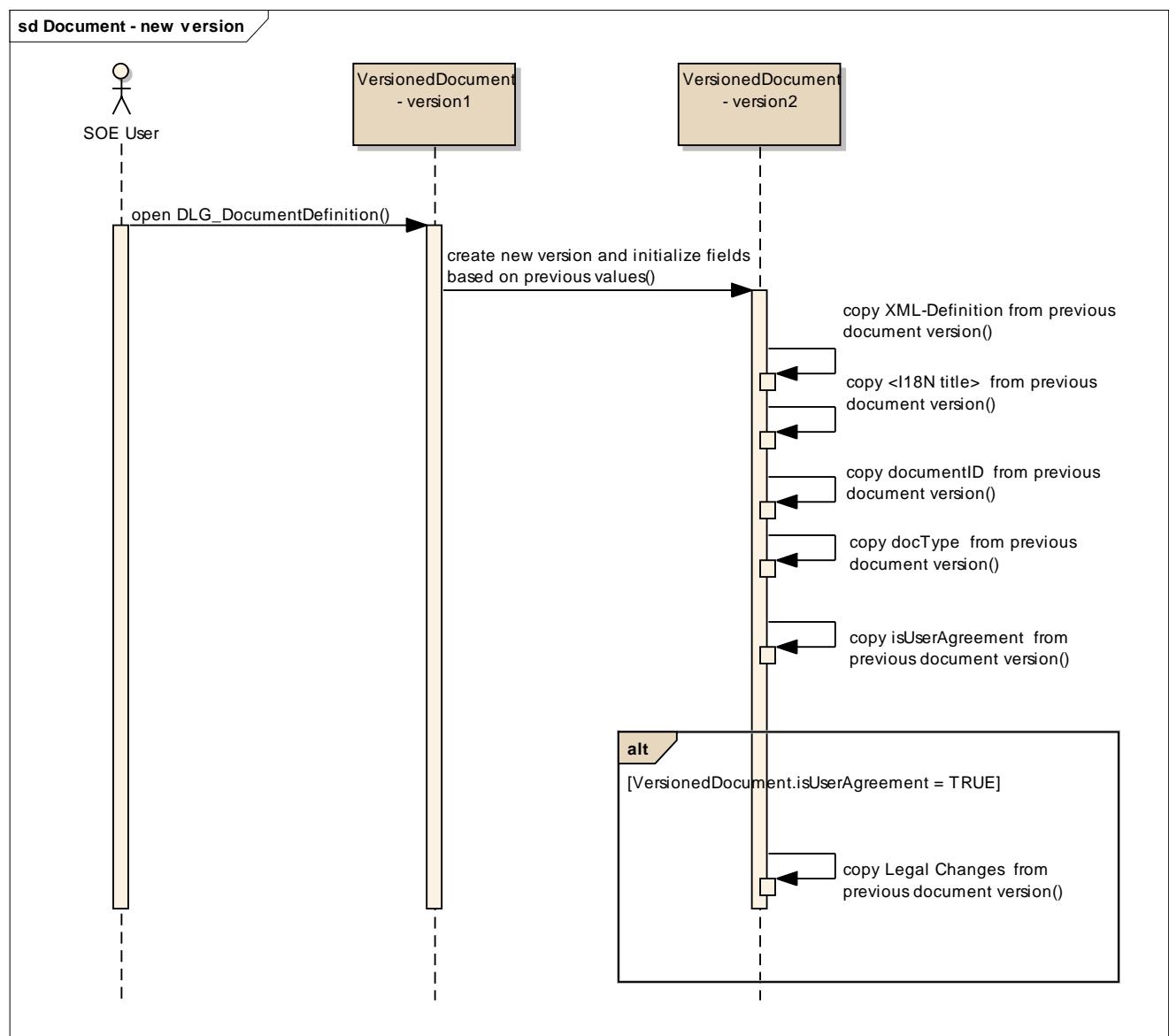
Daimler AG - 70546 Stuttgart  
Telefon +49 (0)711 17-2 22 44  
USI-Id-Nr. DE 812 526 215  
dialog@daimler.com  
www.daimler.com

Mercedes-Benz - sind eingetragene Marken der Daimler AG, Stuttgart, Deutschland

Figure 16: Example layout for a cover letter

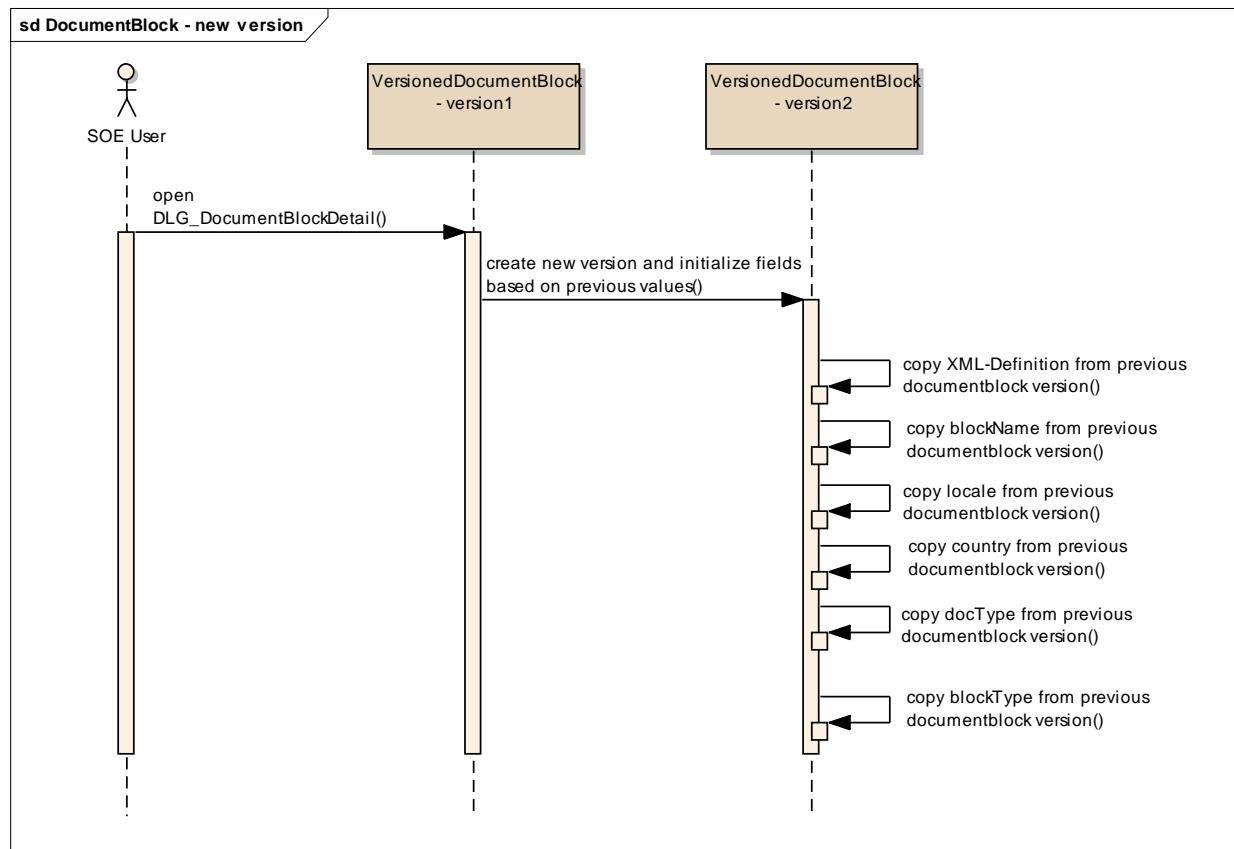
When a new document element is generated some attributes will be copied from the available previous version.

The following sequence diagram illustrates what attributes are prefilled when a new version of a document is created.



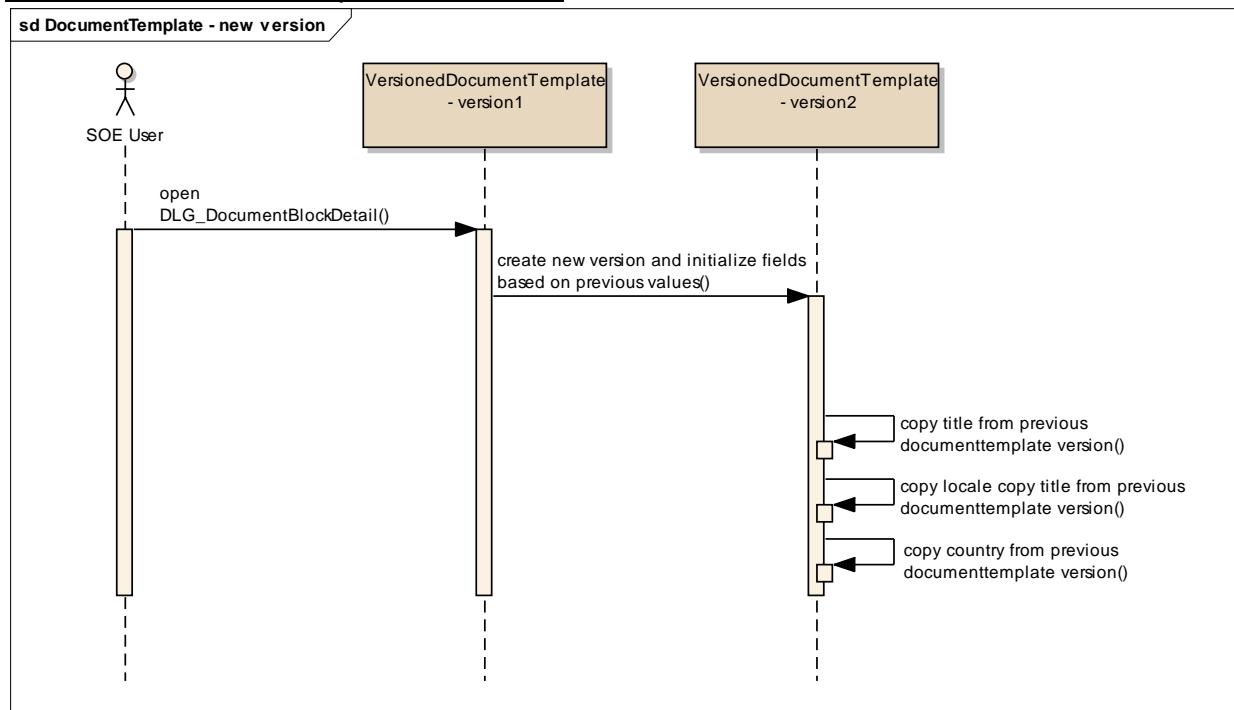
**Figure 17:** Creating a new version of a document based on previous latest version

The following sequence diagram illustrates what attributes are prefilled when a new version of a document block is created.



**Figure 18:** Creating a new version of a document block based on previous version

The following sequence diagram illustrates what attributes are prefilled when a new version of a document template is created.



---

**Figure 19:** Creating a new version of a document template based on previous version

### 2.3.17.1 General Document Definition Structure

This chapter provides an overview of the general layout options of a document definition. The layout and content is defined via XML.

Figure 20 illustrates the different XML blocks and data fields that the document definition can contain.

An example is shown in Figure 16: Example layout for a cover letter that illustrates the use of custom tags (colored in turquoise), document blocks (colored in purple) and document template (i.e. the Mercedes star, the complete footer).

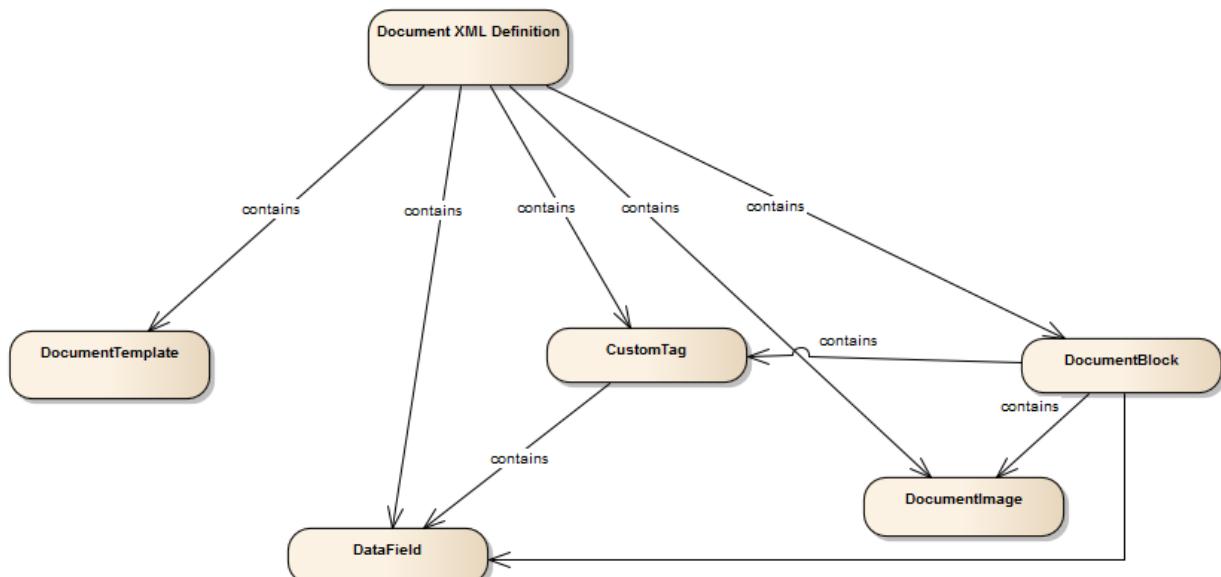


Figure 20: Document XML Definition Structure

### 2.3.17.2 Preview Document changes

A preview feature is available in order to visualize the effect of document-elements modifications done inside a change session.

The data used for previewing a document is provided by two sources:

- property file PROP\_PREVIEW\_DATA: the here stored data is used for all requested combinations of country and language
- the content of the document stored inside the entity <XMLDocumentDefinition> and all referenced <CustomTag> entities as defined by SOE System Specification.

If the current user has made any changes upon the master data document elements that impacts the document and a preview is requested, then a PDF document including these changes will be generated.

---

### **2.3.17.3 Running Number Generator**

#### Document Reference Number for the Customer Communication

The reference number is generated and used in outgoing customer communication. E.g. the reference number is printed on the cover letter (Anschriften) or in the email body sent to a customer when he is notified about certain contractual changes.

It is a numeric value stored in a character string. The numeric value is of a positive number (use as data type either a long or an integer) that starts at 1. The maximum value is 9999999999.

The reference number has a length of 10 characters and has leading zeros for the missing positions, e.g. numeric value = 1. Reference number = "0000000001".

*Note: in order to avoid an overflow, when the reference number reaches 9000000000 the following warning will be written in the application log: [WARN "Archive old entries stored by the CommunicationLogBook table. The threshold of 9000000000 entries has been reached].*

### **2.3.17.4 Composition of Cover Letters**

The figure below shows how the content of the cover letter associated to the document trigger "User Agreement – legal document changed" is generated.

For each contractual change (depicted in the left part of the figure) a summary will be presented directly in the content of the cover letter.

The summary of a legal change will be defined inside a document block that is attached to each new version of a legal document (e.g. for the new version of the user agreement "Basic Services", an associated document block that describes the legal change is provided).

When the cover letter is generated, all changes of legal documents are bundled inside the cover letter. Their associated "Legal Changes" Block is extracted and resolved inside the cover letter (depicted in the right part of the figure). The newer versions of the user agreements are attached either as appendixes to the cover letter or as attachments to the email that an end-customer receives when he is informed about new legal changes.

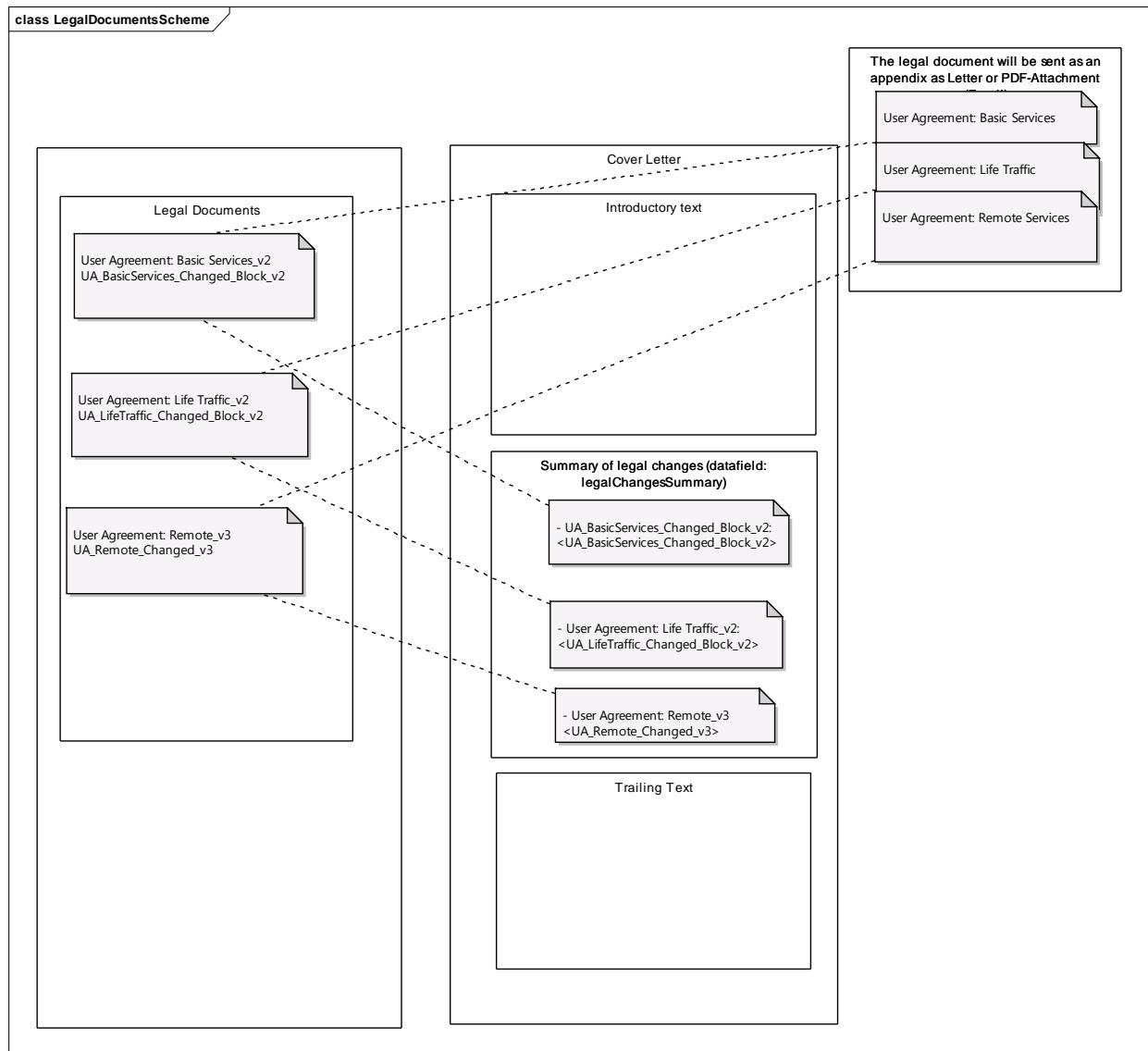


Figure 21: Generation of the cover letter associated to the changed contract information

### 2.3.17.5 Formatting a customer address

The way an address is formatted, depends on the usage context.

If the address is generated in order to be printed on documents triggered by internal logic of SOE to be send to the customer (either via letter or email), then:

- If the customer has the address country in Germany, then the address needs to be displayed consistently throughout all documents bundled into the letter / attached to the email:
  - e.g. both the cover letter and the user agreements must contain the same format of the address
- If the customer does not have the address country in Germany, then:
  - the address displayed on the cover letter is formatted based on the rules defined by the German Post.
  - the address displayed on all other attachments bundled with the cover letter is formatted like preconfigured inside the country specific CustomTag CUSTOMER\_ADDRESS\_LONG

If the address is generated as an answer to a request via an external interface then the customer address is formatted like preconfigured inside the country specific CustomTags: CUSTOMER\_ADDRESS\_LONG or CUSTOMER\_ADDRESS\_SHORT. e.g.: in order for a customer address to be displayed by the front end systems (e.g. inside POS Module), the external interface IF\_SOE\_GetFormattedAddress is called. The table below summarizes an example of how an address of a customer living in Germany versus a customer living in United Kingdom is formatted depending on the usage context.

Usage Context	Any Adresse outside Germany	Address inside Germany
<b>Cover Letter</b>	Mr Hugh Grant Kensington Gore London LONDON SW7 2AP VEREINIGTES KÖNIG-REICH	Karl Heinz Alte Gasse 5 70599 Stuttgart
<b>User Agreement sent together with a Cover Letter</b>	Mr Hugh Grant Kensington Gore London London SW7 2AP Great Britain	Karl Heinz Alte Gasse 5 70599 Stuttgart
<b>User Agreement as E-Mail attachment</b>	Mr Hugh Grant Kensington Gore London London SW7 2AP Great Britain	Karl Heinz Alte Gasse 5 70599 Stuttgart
<b>User Agreement generated on Front end request</b>	Mr Hugh Grant Kensington Gore London London SW7 2AP Great Britain	Karl Heinz Alte Gasse 5 70599 Stuttgart Deutschland
<b>Front End call requesting only the address</b>	Kensington Gore London London SW7 2AP Great Britain	Alte Gasse 5 70599 Stuttgart Deutschland

### 2.3.17.6 Document Generation

A document is generated with the help of the document definition that describes the structure of the document and how all of it fits together. Which documents need to be created is determined by their association to the document triggers. The first step is to retrieve the document definition of a certain document (e.g. a user agreement) and use it to retrieve the localized document template. Any document block and custom tag that is specified in the document definition is then resolved (to the respective localized version) expanded into the document definition. All data fields within the document definition are now resolved to their respective dynamic content. The available content is then rendered onto the document template in accordance to the font style sheet that specifies the look and feel of the fonts. The result is a dynamically built document, with localized content.

### **2.3.17.6.1 Document Trigger**

Certain actions by the customer result in documents to be printed out (for a customer signature) or sent out (mail or email). Such an action for instance is when a customer signs user agreements at the dealership. To be able to sign such said document, it needs to be printed in a certain format first. Another action for instance, is when user agreements change over time and the customer needs to be informed, either via mail or email, about the new documents. The amount of specific documents is not fixed though and can be maintained. For instance, a new user agreement might be added to the list if a new service is available and is ought to receive its own user agreement.

For those reasons, the SOE needs to be able to decide if the consequence of a certain action leads to a document be mailed, emailed or merely sent to another system (the conglomerate of the three distribution methods is called “output type”). A trigger for those actions is needed that defines which output type is expected. In return, an administrator can select the specific documents, emails and attachments for each expected output type.

The following matrix shows the association between the trigger and the supported output type:

Trigger \ Output type	PDF	Email	Letter
User Agreement - accepted		X	X
User Agreement - declined		X	X
User Agreement – lookup (MyM)	X		
User Agreement – signature (POS)	X		
User Agreement – legal document changed		X	X
Vehicle Registration			X
Vehicle Registration Confirmation	X		
Vehicle Separation Authorization	X		
Vehicle Separation Information			X

Table 8: Document Trigger and Channel association

Note: Emails that are generated by the SOE are in plain text only (no RTF, HTML). Documents can be attached to emails as Multipurpose Internet Mail Extensions (MIME).

### **2.3.17.6.2 Document Definition**

Because a document may have dynamic content (e.g. customer information), there is a need for a document definition, that describes the structure of the document and how it all fits together. The document definition itself is country and language independent.

The document definition therefore specifies:

- Which document template is used (see chapter 2.3.17.6.3 Document Template)
- Which document blocks are used in the document (see chapter 2.3.17.6.6 Document Block)
- Which custom tags are used in the document (see chapter 2.3.17.6.7 Custom Tag)
- Which data fields are used in the document (see chapter 2.3.17.6.5 Dynamic Content)

- Which font style is used to display the particular content (see chapter 2.3.17.6.4 Font Style Sheet)

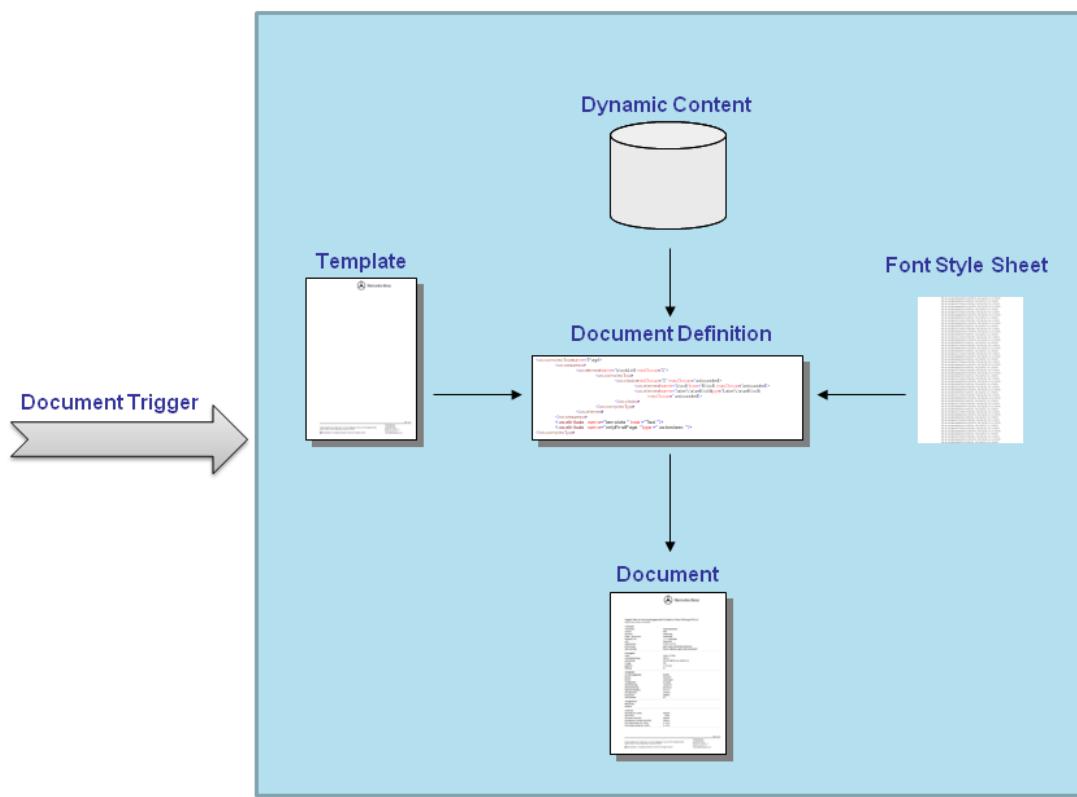


Figure 22: Document Definition Overview

### 2.3.17.6.2.1 Available XML Configuration Elements

This chapter describes the requirements regarding the XML document configuration possibilities.

The following elements and attributes are available:

**Document** – The root node which contains basic information:

- TemplateId/TemplateVersion: Refer to the id and the version of an uploaded PDF template to be used as background for the document currently being created. If the uploaded PDF has less pages than the document currently being created, the last page of the PDF will be used for all additional pages. Usually the PDF will consist of one title page and one page for subsequent pages.

**Page** – Defines logical pages. The main purpose is to structure a document logically. At the end of a logical page a page break will be inserted. However a logical page might stretch across several pages in the generated PDF. A page may contain a Header, Footer and Blocks. If they occur, they have to be defined in that order in the page definition. Nonetheless none of them must occur.

**Header/Footer** – Defines a recurring header/footer which is repeated on every page of the document. The header may contain one or many **Blocks**.

**Block** – Defines a part of the document which contains other elements.

---

A block has/may have the following attributes:

- Font: see chapter 2.3.17.6.4 - Font Style Sheet. The font will be applied to all sub-elements of the block, unless defined otherwise in a sub-element.
- Content alignment: (left (default), center, right): Defines the alignment of contained elements.
- Column: This attribute allows positioning Blocks or other document elements horizontally in columns. All sibling elements (= elements on the same XML level as the current) will be put in columns, too. The amount of columns is derived from the highest available column number in the current XML level.
- Line: Defines whether a new Block is indicated by a horizontal line which runs on top of the Block. Possible values: NONE (default), SOLID.
- Condition: A string which refers to a condition which is evaluated during document generation. If the condition applies, the Block will be printed, otherwise it will be skipped. The applicable conditions depend on the document and thus are not described here.

The following elements can be contained in a Block:

- Block: A nested block.
- Text: Some fixed text.
- CR: A carriage return.
- Content: A placeholder which shall be replaced with business data, e.g. the name of the customer.
- Document Blocks: see chapter 2.3.17.6.6 - Document Block.
- Custom Tags: see chapter 2.3.17.6.7 Custom Tag.
- Document Image: see chapter 2.3.17.6.8

**StaticAttachment** – Defines if and which Document Template is attached at the end of the document. The position of this element within the Document Definition does not matter, as the template is always attached at the end of the document.

A StaticAttachment has the following attributes:

- TemplateId/TemplateVersion: Refer to the id and the version of an uploaded PDF template to be used as an attachment for the document currently being created

An example document could look as follows. Note that the XML syntax does not have to completely match what's written here.

```
<document templateId=USERAGREEMENT templateVersion=1>
  <header>
    <block ContentAlignment=CENTER font=DOCUMENTHEADERFONT>
      <text>Test header text</text>
    </block>
  </header>
```

```
<page>
  <block>
    <text column=0>Text in left column</text>
    <text column=1>Text in right column</text>
  </block>
  <content field=vehiFin />
  <CR/>
  <customTag tagId=CUSTOMER_ADDRESS_LONG />
  <CR/>
  <text condition=PREVIEW>This is a preview only!</text>
  <docBlock docBlockId=MAIN_CONTENT_1 />
  <CR/>
  <block columns = "2" >
    <block>
      <image name="FACSIMILE_1" version="1">
    </block>
    <block>
      <image name="FACSIMILE_2" version="1">
    </block>
    <block>
      <text> i.V. Max Mustermann </text>
    </block>
    <block>
      <text> i.V. Maxima Musterfrau </text>
    </block>
    <block>
      <text> Customer Management </text>
    </block>
    <block>
      <text> Customer Experience </text>
    </block>
  </block>
  </page>
  <staticAttachment templateId=TERMS templateVersion=1 />
</document>
```

---

### **2.3.17.6.3 Document Template**

Header and footer often contain very general information that is reused on many pages within a document, as well as across many documents. To allow relatively easy reusability, such information can be used as a template, which is stored as a PDF. Such a PDF can be reused across many documents, which is then combined (the text is printed onto the PDF) with the dynamic content for the actual document. This procedure has the benefit of separating static (the PDF) and dynamic content and therefore the reuse of static content. Completely static documents that have no variable information printed on top of it are also good candidates for being used as a template (e.g. terms and conditions). Because Document Templates are PDF specific, they cannot be used for maintaining emails. The Document Template is referenced inside the Document Definition by its name and version.

Note: The terms and conditions have to be added to the user agreements as a document template, so that it can be printed out as an additional page of the user agreement. This is due to the fact that the terms and conditions are not requested separately by the external systems. Each user agreement therefore has the terms and conditions attached to them.

Each template is defined by the following properties:

- PDF Name
- Country (ISO 3166, e.g. "DE" for Germany)
- Locale (ISO 639, e.g. "de\_DE" for German (Germany))
- Version

### **2.3.17.6.4 Font Style Sheet**

A document can have many fonts and styles for various content fields. The font style can be configured in the application configuration (see PROP\_LOCALES, chapter 2.3.15), which contains style definitions per country and language. Each definition describes the font, typeset, font size, style and color. Additionally fonts can be configured for different locales (country and language combination). For example, certain languages require characters which are not available in standard configured fonts. For documents to be available in these languages, locale-based font configuration should be used to make documents available in the right language with correct representation of all characters. Default fonts are used where none are configured.

Note: Font styles are of no consequence for emails, as emails are plain text only and should therefore not be used for that purpose. Otherwise it will lead to a validation error of the XML.

Following fonts types are available:

- documentHeaderFont
- documentSubHeaderFont
- blockHeaderFont

- labelFont
- contentFont
- messageFont
- subscriptionFont
- textFont

Example configuration for a font type:

DE.de.documentHeaderFont=c063003t.ttf,Cp1252,12,0,000000

- Country = DE (Germany)
- Language = de (German)
- documentHeaderFont = name of the style sheet
- c063003t.ttf = name of the TrueType Font
- CP1252 = name of the type set
- 12 = font size in pt
- 0 = font style regular (1=bold, 2=italic)
- 000000 = color black

If the specific configuration cannot be found, a default configuration will be used instead. This default configuration will also be specified in the application configuration.

### 2.3.17.6.5 Dynamic Content

Dynamic content is referenced via data fields. During the document generation, data fields are resolved and matched against available data. If no data is available to back up the data field, then the data field is omitted. The available data fields are all customer, vehicle and contract attributes.

Here is a list of available data fields that can be used:

Data Field	Formatting	Description
companyCity	-	The city of the company.
companyCountry	-	The country of the company.
companyHouseNo	-	The house number of the company.
companyName	-	The name of the company.
companyStreet	-	The street of the company.
companyZip	-	The zip code of the company.
confirmationCode	-	Code for the vehicle confirmation
custAdditionalAddressLine1	-	Additional address of the customer.
custAdditionalAddressLine2	-	Additional address of the customer.
custAdditionalAddressLine3	-	Additional address of the customer.
custAddressCountry	Localization depends on the input locale or the customer's country.	The country of the customer's address.
custBirthday	Date, formatted based on the customer's country.	The birthday of the customer.
custCity	-	The city of the customer.

Data Field	Formatting	Description
custDoorNo	-	The door number of the customer.
custEmail	-	The email of the customer.
custFirstName	-	First name of the customer.
custFloorNo	-	The floor number of the customer.
custHouseName	-	The house name of the customer.
custHouseNo	-	The house number of the customer.
custLandlinePhone	-	The landline number of the customer.
custLastName1	-	Last name of the customer (part 1).
custLastName2	-	Last name of the customer (part 2).
custMiddleInitial	-	The middle initial of the customer.
custMobilePhone	-	The mobile phone number of the customer.
custNamePrefix	-	The prefix of the customer's name (e.g. Count)
custPreferredLanguage	Localization depends on the input locale or the customer's country.	The preferred language of the customer.
custProvince	-	The province of the customer.
custShortSalutation	Localization depends on the input locale or the customer's country.	The short form salutation of the customer (e.g. Mr. or Ms.).
custState	-	The state of the customer.
custStreet	-	The street of the customer.
custStreetType	Localization depends on the input locale or the customer's country.	The street type of the customer.
custTitle	Localization depends on the input locale or the customer's country.	The title of the customer (e.g. Prof.).
custZip	-	The zip code of the customer.
documentValidFromDate	Date; localization depends on the input locale or the country.	The valid from date (for new customers) of the underlying user agreement.
documentVersion	-	The version of the underlying user agreement.
printDate	Localization depends on the input locale or the country.	Can be used to place and show the date when a document has been printed.
vehiBaumuster	-	The baumuster of the vehicle.
vehiBaumusterDescription	Localization depends on the input locale or the customer's country.	The localized description of the baumuster.
vehiFin	-	The FIN/VIN of the vehicle.
vehiFirstRegDate	Date, formatted based on the customer's country.	The first registration date of the vehicle.
documentReferenceNumber	-	Will be referenced inside cover letter templates (Anschriften) or email body templates.
legalChangesSummary	Localization depends on the input locale or the customer's country.	Placeholder for summarizing changes between different versions of legal documents. This data field is intended to be used in the cover letter (see Figure 21: Generation of the cover letter associated to the changed contract information)
titleOfLegalDocuments	Localization depends on the input locale or the customer's country.	Placeholder for summarizing affected user agreements. This data field is intended to be used in the subjects of cover letters.

Table 9: List of available data fields

---

### **2.3.17.6.6 Document Block**

The document block contains localized content that is referenced in the non-localized document definition only by its name and version. For instance, localized content may be a welcome text that is written in the German language. This allows the document block references to be used across multiple document definitions, while the correct localized document block is determined dynamically when the document is built. The Document Block is referenced by its name and version. Further a Document Block can contain a Custom Tag and references to other document blocks.

*Note: a circular dependency check (the validation that a document block does not directly or indirectly references itself) is out of scope.*

#### **2.3.17.6.6.1 Available XML Configuration Elements**

This chapter describes the requirements regarding the XML document configuration possibilities.

The following elements and attributes are available:

**Block** – Defines a part of the document which contains other elements.

A block has/may have the following attributes:

- Font: see chapter 2.3.17.6.4 - Font Style Sheet. The font will be applied to all sub-elements of the block, unless defined otherwise in a sub-element.
- Content alignment: (left (default), center, right): Defines the alignment of contained elements.
- Column: This attribute allows positioning Blocks or other document elements horizontally in columns. All sibling elements (= elements on the same XML level as the current) will be put in columns, too. The amount of columns is derived from the highest available column number in the current XML level.
- Line: Defines whether a new Block is indicated by a horizontal line which runs on top of the Block. Possible values: NONE (default), SOLID.
- Condition: A string which refers to a condition which is evaluated during document generation. If the condition applies, the Block will be printed, otherwise it will be skipped. The applicable conditions depend on the document and thus are not described here.

The following elements can be contained in a Block:

- Block: A nested block.
- Text: Some fixed text.
- CR: A carriage return.
- Content: A placeholder which shall be replaced with business data, e.g. the name of the customer.
- Document Image: see chapter **2.3.17.6.8**

An example document block could look as follows. Note that the XML syntax does not have to completely match what's written here.

```

<block font=BODY_TEXT_ITALIC>
    <content field=vehiFin />
    <CR/>
    <text condition=PREVIEW>This is a preview only!</text>
    <block>
        <image> name = "CAR_SYMBOL" version="1" </image>
    </block>
</block >
```

### 2.3.17.6.7 Custom Tag

A custom tag is country specific and contains a list of data fields that define the structure of the custom tag. The necessity to use a custom tag results from information that may need to appear structurally different on a document for a specific country. For instance, the customer address has a different structure based on the country. The formatting of an address may differ across countries: e.g. For France, the country name must appear in upper case. Custom tags allow the configuration of country specific, but language independent structures and can then be used across multiple documents definitions, without the need of creating separate document definitions for each country. See Figure 20 for an overview of how the custom tag fits into the document definition. The custom tag is referenced by its CustomTagType within the Document Definition or the Document Block.

Following custom tag types are available:

- CUSTOMER\_ADDRESS\_LONG – defines the structure of a customer address. The long version is intended to include not only the rump address (i.e. the street, city, ...) but as well the full name and if necessary the salutation of a person.
- CUSTOMER\_ADDRESS\_SHORT – defines the structure of a customer address in the short version (the short version is intended to be represent a address without name fields, i.e. just the rump of an postal address)
- CUSTOMER\_SIGNATURE\_FIELD – defines the structure of a signature area where the customer signs
- SALUTATION – defines the salutation of the customer
- DAIMLER\_ADDRESS – defines the structure of the Daimler address (The address itself is configurable through the system's configuration)

#### 2.3.17.6.7.1 Available XML Configuration Elements

This chapter describes the requirements regarding the XML document configuration possibilities.

The following elements and attributes are available:

**Block** – Defines a part of the document which contains other elements.

A block has/may have the following attributes:

- 
- Font: see chapter 2.3.17.6.4 - Font Style Sheet. The font will be applied to all sub-elements of the block, unless defined otherwise in a sub-element.
  - Content alignment: (left (default), center, right): Defines the alignment of contained elements.
  - Column: This attribute allows positioning Blocks or other document elements horizontally in columns. All sibling elements (= elements on the same XML level as the current) will be put in columns, too. The amount of columns is derived from the highest available column number in the current XML level.
  - Line: Defines whether a new Block is indicated by a horizontal line which runs on top of the Block. Possible values: NONE (default), SOLID.
  - Condition: A string which refers to a condition which is evaluated during document generation. If the condition applies, the Block will be printed, otherwise it will be skipped. The applicable conditions depend on the document and thus are not described here.

The following elements can be contained in a Block:

- Block: A nested block.
- Text: Some fixed text.
- CR: A carriage return.
- Content: A placeholder which shall be replaced with business data, e.g. the name of the customer.

The `<content/>` element has the optional attribute: “lettertext”.

When given, the value of this attribute will indicate whether the `field` specified inside the `content` element will be displayed either in upper or in lower case.

The attribute “lettertext” supports the following attributes: {“CAPITAL”, “SMALL”}.

When “CAPITAL” is defined, then the text of the `field` is displayed as capital letter text.

When “SMALL” is defined, then the text of the `field` is displayed as small letter text.

When missing, the content of the `field` is displayed as is.

An example custom tag could look as follows. Note that the XML syntax does not have to completely match what's written here.

```
<block font=BODY_TEXT_ITALIC>
  <content field=vehiFin />
  <content field=vehiBaumuster />
  <content field=vehiFirstRegDate />
  <content field=custAddressCountry lettertext="UPPER"/>
  <CR/>
  <CR/>
  <text condition=PREVIEW>This is a preview only!</text>
```

```
</block>
```

### 2.3.17.6.8 Image

The document image contains no localized or country specific content. It is referenced either in the document definition or in the document block by its name and version.

#### 2.3.17.6.8.1 Available XML Configuration Elements

Because an image is always placed inside a `<block>` element, in order to position the image the available attributes of a `<block>` element can be used:

- Content alignment: (left (default), center, right): Defines the alignment of contained picture (or other elements supported inside a `<block>`).
- Column: This attribute allows positioning Blocks (and images contained inside a block) horizontally in columns. All sibling pictures (= elements on the same XML level as the current) will be put in columns, too.

An example where the positioning of an image could look as follows. Note that the XML syntax does not have to completely match what's written here.

```
<block column = "3" align="CENTER">
  <block align = "LEFT">
    <image name="IMG_1" version="1">
  </block>
  <block align = "CENTER" >
    <image name="IMG_2" version="1">
  </block>
  <block align = "RIGHT">
    <image name="IMG_3" version="1">
  </block>
</block>
```

### 2.3.18 Handling of Profile Data within SOE

SOE does not store profile master data, as this is done in the system CPD. However SOE has the following contact points with profile data:

- Providing information to other systems which profile data is mandatory for a specific country based on a configuration stored in SOE.
- Providing information to other systems whether all necessary profile data is available to activate a specific MBconnect service.
- Generating documents containing profile information according to a configuration stored in SOE.
- Formatting customer addresses for other systems according to a configuration stored in SOE.

Between systems in the MBconnect cluster, customer data is transferred in the so called “UserType” format. This format contains a fixed list of all possible customer data attributes for all countries.

However within SOE, a more generic approach is needed, to efficiently provide the above mentioned functionality. Thus the transient user data within SOE is handled as shown in Figure 23.

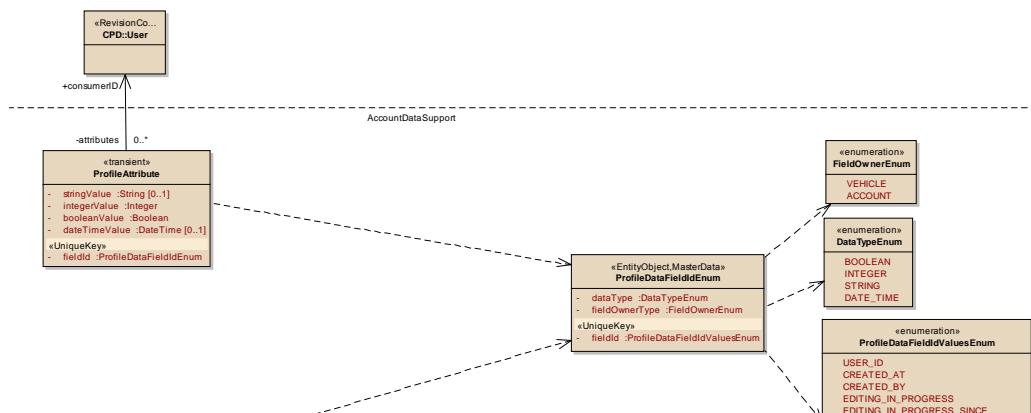


Figure 23: Handling of profile data within SOE

On the interfaces dealing with profile data, the following mapping shall be applied:

For each retrieved parameter from CPD, an instance of ProfileAttribute is created. If a parameter is empty/not available, NO ProfileAttribute is created for this parameter. If one is created, depending in the data format of the attribute, the value is mapped to

- ProfileAttribute.StringValue or
- ProfileAttribute.IntegerValue or
- ProfileAttribute.BooleanValue or
- ProfileAttribute.DateTimeValue.

The attribute ProfileAttribute.DataType is set accordingly. SOE distinguishes between account and vehicle related profile data.

The attribute ProfileAttribute.FieldId and ProfileAttribute.FieldOwnerType are set to what's printed in Table 10: Mapping of CPD parameters to the SOE data model.

Parameter Name (CPD)	Mand./Opt. (CPD)	Format (CPD)	FieldID (SOE)	FieldOwnerType (SOE)
userId	Opt.	String	USER_ID	ACCOUNT
accountVerified	Opt.	Boolean	ACCOUNT_VERIFIED	ACCOUNT
createdAt	Opt.	Date	CREATED_AT	ACCOUNT
createdBy	Opt.	String	CREATED_BY	ACCOUNT
editingInProgress	Opt.	Boolean	EDITING_IN_PROGRESS	ACCOUNT
editingInProgressSince	Opt.	Date	EDITING_IN_PROGRESS_SINCE	ACCOUNT
optInMailStatus	Opt.	String	OPT_IN_MAIL_STATUS	ACCOUNT
salutation	Opt.	String	SALUTATION	ACCOUNT

Parameter Name (CPD)	Mand./Opt. (CPD)	Format (CPD)	FieldID (SOE)	FieldOwnerType (SOE)
title	Opt.	String	TITLE	ACCOUNT
namePrefix	Opt.	String	NAME_PREFIX	ACCOUNT
firstName	Opt.	String	FIRST_NAME	ACCOUNT
middleInitial	Opt.	String	MIDDLE_INITIAL	ACCOUNT
lastName1	Opt.	String	LAST_NAME_1	ACCOUNT
lastName2	Opt.	String	LAST_NAME_2	ACCOUNT
Email	Opt.	String	EMAIL	ACCOUNT
landlinePhone	Opt.	String	LANDLINE_PHONE	ACCOUNT
mobilePhone	Opt.	String	MOBILE_PHONE	ACCOUNT
mobilePhoneNumberVerified	Opt.	Boolean	MOBILE_PHONE_NUMBER_VERIFIED	ACCOUNT
addressCountry	Opt.	String	ADDRESS_COUNTRY	ACCOUNT
state	Opt.	String	STATE	ACCOUNT
province	Opt.	String	PROVINCE	ACCOUNT
city	Opt.	String	CITY	ACCOUNT
zip	Opt.	String	ZIP_CODE	ACCOUNT
Street	Opt.	String	STREET	ACCOUNT
streetType	Opt.	String	STREET_TYPE	ACCOUNT
houseNo	Opt.	String	HOUSE_NUMBER	ACCOUNT
housename	Opt.	String	HOUSE_NAME	ACCOUNT
floorNo	Opt.	String	FLOOR_NUMBER	ACCOUNT
doorNo	Opt.	String	DOOR_NUMBER	ACCOUNT
additionalAddressLine1	Opt.	String	ADDITIONAL_ADDRESS_LINE_1	ACCOUNT
additionalAddressLine2	Opt.	String	ADDITIONAL_ADDRESS_LINE_2	ACCOUNT
additionalAddressLine3	Opt.	String	ADDITIONAL_ADDRESS_LINE_3	ACCOUNT
birthday	Opt.	Date	BIRTHDAY	ACCOUNT
preferredLanguage	Opt.	String	PREFERRED_LANGUAGE	ACCOUNT
offlineUser	Opt.	Boolean	IS_OFFLINE_USER	ACCOUNT
marketingCountry	Opt.	String	MARKETING_COUNTRY	ACCOUNT
SubscribedToNewsletter	Opt.	Boolean	IS_SUBSCRIBED_TO_NEWSLETTER	ACCOUNT
ContactedByEmail	Opt.	Boolean	IS_CONTACTED_BY_EMAIL	ACCOUNT
ContactedByPhone	Opt.	Boolean	IS_CONTACTED_BY_PHONE	ACCOUNT
ContactedByLetter	Opt.	Boolean	IS_CONTACTED_BY_LETTER	ACCOUNT
privacyPolicyAcceptedAt	Opt.	Date	PRIVACY_POLICY_ACCEPTED_AT	ACCOUNT
privacyPolicyAcceptedBy	Opt.	String	PRIVACY_POLICY_ACCEPTED_BY	ACCOUNT
privacyPolicyAcceptedVersion	Opt.	String	PRIVACY_POLICY_ACCEPTED_VERSION	ACCOUNT
serviceDealer	Opt.	String	SERVICE DEALER	VEHICLE

Table 10: Mapping of CPD parameters to the SOE data model

### 2.3.19 Configuration of Profile Data Fields

SOE offers a mechanism to other systems, which tells them which profile data fields are needed for creating and updating a user account in a specific country or activating a specific service. The information about the data fields is distinguished in general information and country specific information.

Inside SOE this data is available in specific tables (see Figure 24: Structure of profile data fields as provided to other systems). There is no maintenance dialog for this data, which means it can only be changed by an application deployment. Figure 24 displays the structure of how this information is provided to other systems.

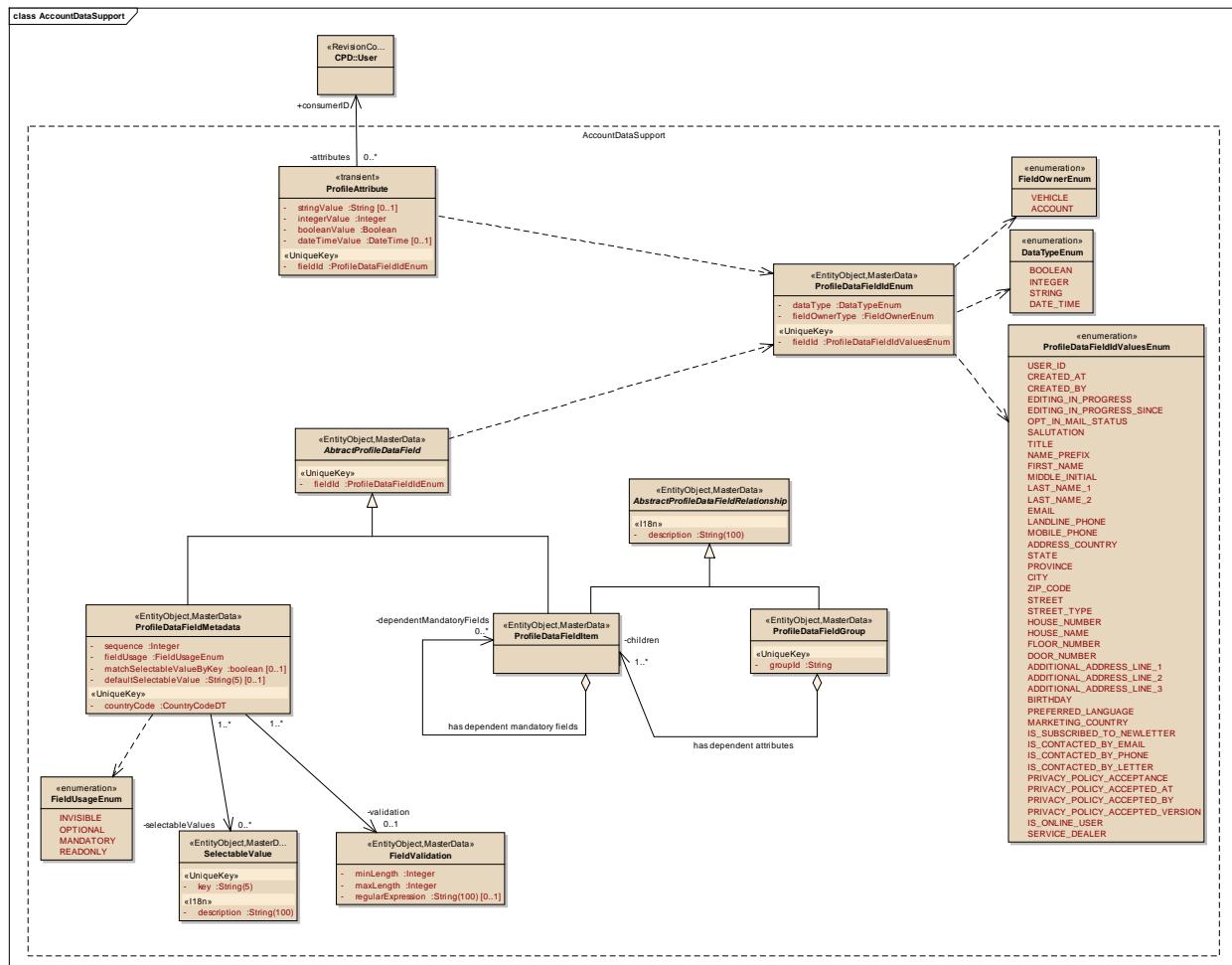


Figure 24: Structure of profile data fields as provided to other systems

### 2.3.19.1 General information

More precisely a list of fields is provided. This list is absolute for all countries. Per field the following general information is available:

- an absolute id specifying the field in a global context
- the ownership determining whether the field belongs to the user's account or a user's vehicle
- dependency towards other fields and groups
- the data type associated to that field

The structure of fields can be configured as follows:

The **dependentMandatoryFields** attribute ensures that optional fields become mandatory under certain combinations of fields. The optional field refers to a list of fields for which at least one becomes mandatory when the first field is used (for further descrip-

---

tion see chapter 8). The depended mandatory fields can be selected per service as required information.

Example: The field communicationChannelPhone refers to the fields mobilePhone and landlinePhone. If the field communicationChannelPhone is used, also at least one of the fields mobilePhone and landlinePhone have to be used.

The profile group object describes virtual groups in the profile data fields. In detail, different sets of profile data fields can be individually grouped in the application configuration. The groups can be selected per service as required information. As required information, the service can be activated only if at least one of referenced profile data fields is filled. In contrast to the profile data fields, the groups themselves are not supposed to be shown in the customer profile.

Example: The group “communication channel” is required for activating a service called “Maintenance Management”. The group itself has the children “communicationChannelEmail”, “communicationChannelLetter”, “communicationChannelPhone”. In that context, the service can only be activated if at least “communicationChannelEmail”, “communicationChannelLetter” or “communicationChannelPhone” is checked.

At the moment, groups are only provided for account related profile data fields.

### **2.3.19.2 Country specific information**

Besides the structural information about the profile data fields, SOE also provides information on the Meta information about the account related profile data fields and which profile data fields are mandatory for signing a user agreement in a specific country. In detail, SOE offers the following country specific information:

- the order sequence in which the field should be displayed on the user interface
- information, whether the field is invisible, visible but optional, visible and mandatory
- optionally a more detailed validation rule can be provided (length, regular expression)
- optionally possible values for populating dropdown boxes including texts in the requested locale and default values if needed

The usage (invisible, optional, mandatory) of a field is defined by the fieldUsage attribute. The usage determines if the data field is required in order to sign a user agreement in a specific country.

### **2.3.20 MBconnect role concept**

The role concept for MBconnect includes the setting of rights to roles for specific resources. For the moment, only two resources types are available for configuration. These types are “DLG” and “SERVICEMASTER”. DLG is used to control the rights of the SOE administration roles for the dialogs in SOE whereas SERVICEMASTER is used to control the rights of the MBconnect roles administrating the servicemasters.

---

For the moment, only “SERVICEMASTER” shall be provided to adjacent systems, so it shall be explained in this chapter in more detail.

Basically, each servicemaster is enriched by a set of rights that control the use of the servicemaster by a specific role. For each servicemaster and role, the rights can be set individually. These roles are:

Right	Description
MBC.MBC_RETAIL	The retailer who is using POS.
MBC.MBC_SUPPORT	The CAC support who is using POS.
MBC.VEHICLE_MASTERUSER	The master user who is using MBconnect services.
MBC.VEHICLE_SUBUSER	The sub user who is using MBconnect services.
MBC.DFS	The user working for Daimler Financial Services.

Table 11: MBconnect servicemaster roles

Thereby, for a given servicemaster a role can be assigned to one of the following rights. The rights are ordered from none rights to full rights. They are:

Right	Description
NONE	The representative of the specific role does not have any rights regarding the service.
READ	The service can be read and is visible to the representative of the specific role holding that right.
ACTIVATE	The service can be activated by the representative of the specific role holding that right. This also includes the right to see a service (READ).
DEACTIVATE	The service can be deactivated by the representative of the specific role holding that right. This also includes the right to see a service (READ).
WRITE	The service is visible to the representative of the specific role holding that right and further can be activated and deactivated.

Table 12: MBconnect servicemaster rights

For the assignment of the rights to the roles, a set of rules will be considered:

- If a servicemaster needs a signed user agreement in order to be activated, the servicemaster has to be visible to the master user (“WRITE” right).
- A sub user only can have the rights “NONE” and “READ”.

---

## 2.4 Dialogs Overview

### 2.4.1 Basic crosscutting dialog functionality

#### 2.4.1.1 Change detection / confirmation dialog

All data maintenance dialogs shall be equipped with change detection. When the user tries to navigate away from a dialog with unsaved changes, the user is informed about this fact and asked if he/she really wants to navigate away from the dialog.

#### 2.4.1.2 Working on operational data

Most of the administrative changes executed on operational data are only possible inside a change session. In order for the changes to take effect the change session must be released. Remove warning on initial startup screen and place it for all SOE dialogs that haven't been changed inside this change request.

#### 2.4.1.3 Basic dialog elements and features

##### Paging

When showing tables with a large amount of data, paging shall be applied. Additionally the user has the possibility to select the amount of entries shown per page. The selectable amount of items is configurable in the application configuration (see PROP\_PAGING\_ITEMS, chapter 2.3.15). Upon changing the selection, the page is refreshed.

##### Table sorting

Basically tables shall be sortable alphabetically by clicking the column header. If a table cannot be sorted, it will be mentioned in the dialog's specification.

##### Checkboxes and delete button in overview dialogs

When pressing the delete button in overview screens, at least one row must be selected before using the available checkboxes. After a confirmation popup the selected entries will be deleted by an AF defined in the specific dialog.

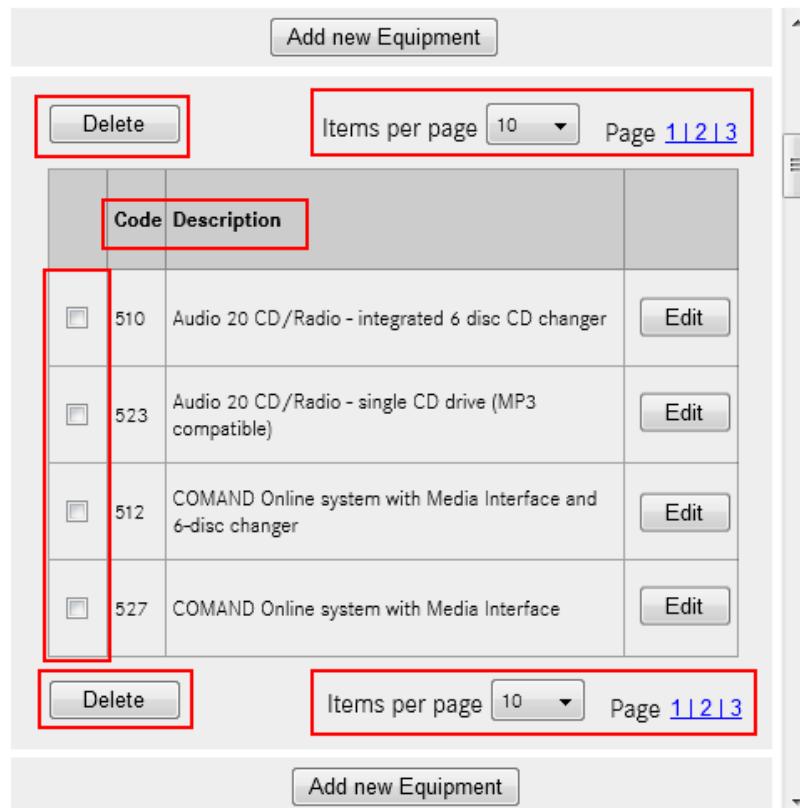


Figure 25: Illustration of basic dialog elements (marked in red)

#### 2.4.1.4 Basic considerations

##### Translation of labels

All labels on the dialogs are supposed to be translatable in the application configuration (see PROP\_GUI\_TRANSLATION, chapter 2.3.15). The language chooser on the DLG\_Main allows the user to switch the language. Initially the application comes with the locales “German (Germany)” and “English (United Kingdom)”. The default language will be “en\_UK”. If a translation is not available, then simply the technical key of the specific label shall be displayed.

##### Translation of displayed (master) data

If a certain language is chosen, this is also the language that translatable (master) data will be displayed in screens. In case a specific and also a default/fallback translation is missing, nothing shall be displayed, as defined in the translation mechanism for data (see Internationalization/Translation of master data, chapter 2.3.5).

##### Input field length

In general the allowed length for textboxes is not written down in the dialog specification but in the data model. Also implicit necessary format checks such as in date or decimal fields are considered as obvious and hence are not specified explicitly as validation rules.

#### 2.4.1.5 Translation tables

For maintaining translations of master data the same dialog element is used among all dialogs. It consists of a table which displays all available languages and their transla-

---

tions. As described before, for each language there is one main locale, which is printed in bold font in the translation dialog. Note: The translation table cannot be sorted. Instead it is always sorted by language name and country name.

Description	
Language	Translation
<b>German (Germany)</b>	Live Traffic
German (Austria)	Live Verkehr
German (Switzerland)	
<b>English (United Kingdom)</b>	Live Traffic
English (Switzerland)	
<b>French (France)</b>	Trafic en direct
French (Belgium)	
French (Switzerland)	
<b>Italian (Italy)</b>	El tráfico en vivo
Italian (Switzerland)	

Figure 26: Translation table as used in all dialogs

## 2.4.2 Dialog frame and dialog flow

*Note: The SOE can operate in different operation modes (→ see chapter Application Configuration). It depends on the user's permissions and the SOE operation mode which dialogs are part of the dialog frame and the dialog flow. This is described in UI Entitlements (→ see chapter User / Organization / Entitlements).*

*In this chapter, dialog frame and dialog flow comprise all general available SOE dialogs.*

Figure 27 shows the main dialog frame of SOE. It consists of four parts:

1. Menu (bottom left)
2. User Name and Language chooser (top right)
3. Actual dialog (bottom right)
4. Action Menu bar (top left)
5. The “about” button which opens DLG\_Info

The "Area for dialog" section contains the dialogs as specified further on in this specification. It will be filled depending on what's selected in the menu.

Note: The user session will be managed by PAI and so the user name will also be provided from there.

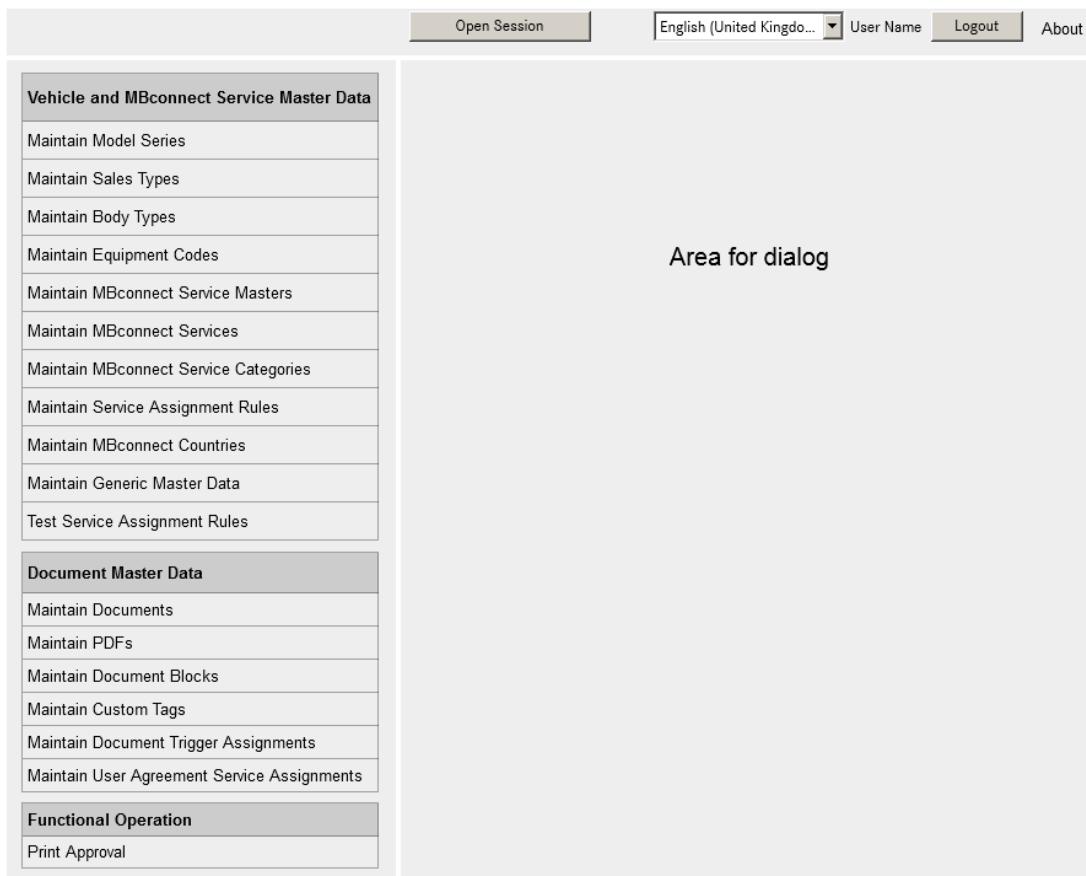


Figure 27: DLG\_Main

Working with the change session impacts the mode in which most dialogs are opened:

- Dialog is opened in edit-mode: if the user has opened the change session and the data is not being edited by another user.
- Dialog is opened in read-only mode: if the user has not opened the change session or if the user has opened the change session and the data is edited by another user. This mode is additionally indicated by the text “The dialog is opened in read-only mode” as can be seen in Figure 28 below.
- The possible states of the common elements displayed by the “overview”-dialogs of each component are summarized in Table 13 below.

The dialog is opened in read-only mode.

Document	User Agreement - Live Traffic	Versions
DocType	PDF	
<input checked="" type="checkbox"/> is User Agreement		
<b>User Agreement Detail</b> Is Enabled No <input type="button" value="Enable"/> Valid from (new customer) 01.01.2014 Info Letter 07.01.2014 Valid from (existing customer) 01.03.2014 Info Email 15.01.2014		
Language	Translation	
German (Germany)	Live Traffic	
German (Austria)	Live Traffic	
German (Switzerland)		
English (United Kingdom)	Live Traffic	
English (United States)		
XML		
<...>		
<input type="button" value="New Version"/>		<input type="button" value="Save"/>

Figure 28: Exemplary dialog (DLG\_DocumentDefinition) opened in read-only mode

Linked Label	Type	State Description
View	Button	<u>Visible:</u> Is shown if there is no open change session related to the current user OR the element is part of the change session of a different user
Edit	Button	<u>Visible:</u> If there is a open change session related to the current user AND the element is not changed by a different user (in a different change session)
<Add new master data element>	Button	<u>Enabled:</u> If there is a open change session related to the current user <u>Disabled:</u> otherwise
Opened sessions	Label/Icon	<ul style="list-style-type: none"> <li>If the element is changed by the current user than decorate with icon:  (placeholder: "my change session")</li> <li>If the element is changed by a different user (in a different change session):  (placeholder: "others change session")</li> <li>If there is no open change session related to the current user: no label or placeholder is displayed</li> </ul>

Table 13: States of the common elements displayed by the "overview"-dialogs

Figure 29 shows the basic dialog flow of SOE. When being within a dialog of the middle column in Figure 29, it is always possible to navigate away by pressing a different button in the menu area of the DLG\_Main. In contrast to this the dialogs shown in the right

---

column of Figure 29 are modal, which means is not possible to leave them unless using the OK or Cancel buttons.



---

Figure 29: Dialog flow of SOE

#### 2.4.2.1 DLG\_Info

*Note: This dialog is accessible directly from DLG\_Main and available for all SOE users without restrictions. It is not a standalone dialog.*

The dialogue opens when clicking on the “about” label on the top right corner of the SOE application.

This dialog displays technical information on the build, the currently logged in user and the environment. The dialog opens as a popup dialog and can be displayed or disposed without affecting any of the dialogs in the background, allowing the user to easily obtain information that may be valuable to the support team.

Label	Content
Build version	Build version, application constant.
Build number	Build number, application constant.
Build date	Build date, application constant.
User Name	Name of the SOE user.
User ID	ID of the SOE user.
List of user roles	Entitlement roles of the SOE user.
Environment	Development stage, can be retrieved from the application configuration.
Operation Mode	The operation mode of a SOE instance (PROP_OPERATION_MODE).
Region ID:	The ID of a SOE Region instance (PROP_REGIONID). Note: Only visible if the operation mode is “REGION”.
Host name	Name of the server.
URL	Current URL.
Time stamp	Current date and time.
Tracking ID	Tracking ID (optional, only if available).

Table 14: Content of DLG\_Info

---

## About

### Build Information

Build version	0.18.0.0-SNAPSHOT
Build number	stable-1353
Build date	2014-04-25_11-39-35

### User Information

User name	Christian Nicu
User ID	CHRNICU
User entitlement groups	SOE.SOE_DOC_MDM, SOE.SOE_VEHICLE_MDM

### Environment Information

Environment	DEV1
Operation Mode	REGION
Region ID	CN
Host	s415vmm148.detss.corpintra.net
URL	<a href="https://soe-dev1.es.corpintra.net/soe/view/components/main/main.jsf">https://soe-dev1.es.corpintra.net/soe/view/components/main/main.jsf</a>
Time stamp	2014-04-29 19:13:15.069
Tracking ID	bz5paGDsDAe7YzCcvipPBIV



Figure 30: Outlook of DLG\_Info

### 3 Component „Vehicle Products“

#### 3.1 Dialogs

##### 3.1.1 DLG\_ModelSeriesOverview

This dialog lists all available model series and allows their creation, deletion or editing.

It displays the information if a model series is exclusively maintained in SOE or if it has been imported by AMDS.

The screenshot shows a software interface for managing model series. At the top, there is a button labeled "Add new Model Series". Below this are buttons for "Delete" and "Items per page" set to 10, along with a "Page" navigation bar showing pages 1, 2, and 3. The main area is a table with the following data:

	Model Series	Product group	Maintained in SOE	Opened Sessions	Action
<input type="checkbox"/>	123	P	No		<input type="button" value="Edit"/>
<input type="checkbox"/>	124	P	No		<input type="button" value="View"/>
<input type="checkbox"/>	203	P	No		<input type="button" value="Edit"/>
<input type="checkbox"/>	204	P	No		<input type="button" value="Edit"/>
<input type="checkbox"/>	205	P	No		<input type="button" value="Edit"/>
<input type="checkbox"/>	207	P	Yes		<input type="button" value="Edit"/>
<input type="checkbox"/>	215	P	No		<input type="button" value="Edit"/>
<input type="checkbox"/>	447	T	No		<input type="button" value="Edit"/>

At the bottom, there are buttons for "Delete", "Items per page" set to 10, and a "Page" navigation bar showing pages 1, 2, and 3. There is also a "Add new Model Series" button at the very bottom center.

Figure 31 DLG\_ModelSeriesOverview

### 3.1.1.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>	-	Load all available model series (independent of the market) together with the permitted actions for each loaded model series by calling the loading algorithm described in Chapter 2.3.14.2 ("Loading of master data elements") and sort them by ModelSeries.ModelSeriesID (see details in subchapter "Dialog Elements States")
"Add new Model Series"	Button	Switches to the dialog DLG_ModelSeriesDetail (new mode) (→ see chapter 3.1.2) to create a new model series.
"Delete"	Button	Display error messages in the according message popup if there was at least one model series selected with ModelSeries.MaintainedInSOE="false". Otherwise after displaying a confirmation popup delete the selected model series by calling the AF_DeleteModelSeries (→ see chapter 3.5.10).
"Edit" (1..n)	Button	Switches to the dialog DLG_ModelSeriesDetail (new mode) (→ see chapter 3.1.2) to view or edit the details of the model series. The dialog is opened in edit mode.
"View" (1..n)	Button	Switches to the dialog DLG_ModelSeriesDetail (new mode) (→ see chapter 3.1.2) to view the details of the model series. The dialog is opened in read-only mode.

Table 15: Buttons and functions (DLG\_ModelSeriesOverview)

### 3.1.1.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Table column "Model Series"	Label	The model series.	ModelSeries.ModelSeriesID
Table column "Product group"	Label	The product group of the respective model series.	ModelSeries.ProductGroup
Table column "Maintained in SOE"	Label	The information if a model series is created in SOE ("Yes") or imported by AMDS.	ModelSeries.MaintainedInSOE
Table Column "Opened Sessions"	Label/Icon	Indicates whether the entity ModelSeries is edited inside a change session.	-

Table 16 Form fields and front-end data objects (DLG\_ModelSeriesOverview)

### 3.1.1.3 Dialog field validation

None.

### 3.1.1.4 Configurability (incl. setting for roles)

None.

### 3.1.1.5 Dialog Elements States

Linked Label	Type	State Description
Add new Model Series, Delete	Button	<u>Visible:</u> Always <u>Enabled:</u> If there is an open change session related to the current user.
All other elements	Varies	<u>Visible:</u> Always <u>Enabled:</u> Always

Table 17 Dialog Element States (DLG\_ModelSeriesOverview)

### 3.1.2 DLG\_ModelSeriesDetail

This dialog allows the user to initially create, edit or view a model series in detail.

In order to create a new model series, a model series id must be entered. Additionally a model series has to be set valid for at least one market.

The dialog additionally gives the information if an existing model series is exclusively maintained in SOE or has been imported by AMDS.

For MBC POS module only these model series are relevant that are not out of production. Therefore the dialog offers the possibility to maintain this information for each model series manually.

Model series\* 117

Product group\* P

This model series is exclusively maintained in SOE.

Valid for markets

- Austria (GS0008201)
- Belgium/Luxembourg (GS0008202)
- Denmark (GS0008203)
- France (GS0008204)
- Germany (GS0008205)
- Italy (GS0008206)
- Norway (GS0008207)
- Netherlands (GS0008208)
- Spain (GS0008209)
- Sweden (GS0008210)
- Switzerland (GS0008211)
- United Kingdom (GS0008212)

Model series is out of production

Save Cancel

Figure 32: DLG\_ModelSeriesDetail

### 3.1.2.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>	-	<p><u>Edit and Read only mode:</u>  Load the model series entity including all markets it is valid for. Also load all available markets and sort them by OrgUnit.Name.</p> <p>Use the loading algorithm described in Chapter 2.3.14.2 ("Loading of master data elements")</p> <p><u>New mode:</u>  Load all available markets and sort them by OrgUnit.Name.</p>
"Save"	Button	<p>Save the currently created model series (new mode) or the current changes of the model series (edit mode) by calling AF_SaveModelSeries (→ see chapter 3.5.9) and navigate back to DLG_ModelSeriesOverview (→ see chapter 3.1.1).</p> <p>Display potential error messages in the according message popup.</p>
"Cancel"	Button	<p>Discard all changes and navigate back to DLG_ModelSeriesOverview (→ see chapter 3.1.1).</p>

Table 18: Buttons and functions (DLG\_ModelSeriesDetail)

### 3.1.2.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Model Series	Textbox	The model series.	ModelSeries.ModelSeriesID
“Product group”	Listbox	The product group of the model series.	ModelSeries.ProductGroup
Valid for markets	Checkbox list	If a checkbox is chosen, the model series is assigned to specific markets.	<p><u>Text:</u> OrgUnit.Name + “(“ + OrgUnit.OutletId + “)”</p> <p><u>Available values:</u> Calls IIF_GetMarketsForContinent to retrieve the available markets</p>
Maintained in SOE	Label	Information if a model series is exclusively maintained in SOE or has been imported by AMDS.	<p><u>Text:</u> If ModelSeries.maintainedInSOE=”true” display: “This model series is exclusively maintained in SOE.” Otherwise ModelSeries.maintainedInSOE=”false” display: “This model series has been imported by AMDS.”</p>
Out of Production	Checkbox	If the checkbox is chosen the model series is out of production.	ModelSeries.OutOfProduction
ReadOnlyInfo	Label	Presents this information: “The dialog is opened in read-only mode.”	-

Table 19: Form fields and front-end data objects

### 3.1.2.3 Dialog field validation

Linked Field	Validation	Errormessage
Model series id (only in new mode)	A value must be provided.	VEHPRO_008. Use the screen field name as placeholder 1.
	The model series id must be provided in the format xxx must be numeric.	VEHPRO_016
Product group (only in new mode)	A value must be provided.	VEHPRO_008. Use the screen field name as placeholder 1.
Valid for markets (only in new mode)	At least one market has to be selected.	VEHPRO_017

Table 20: Dialog field validations (DLG\_ModelSeriesDetail)

### 3.1.2.4 Configurability (incl. setting for roles)

None.

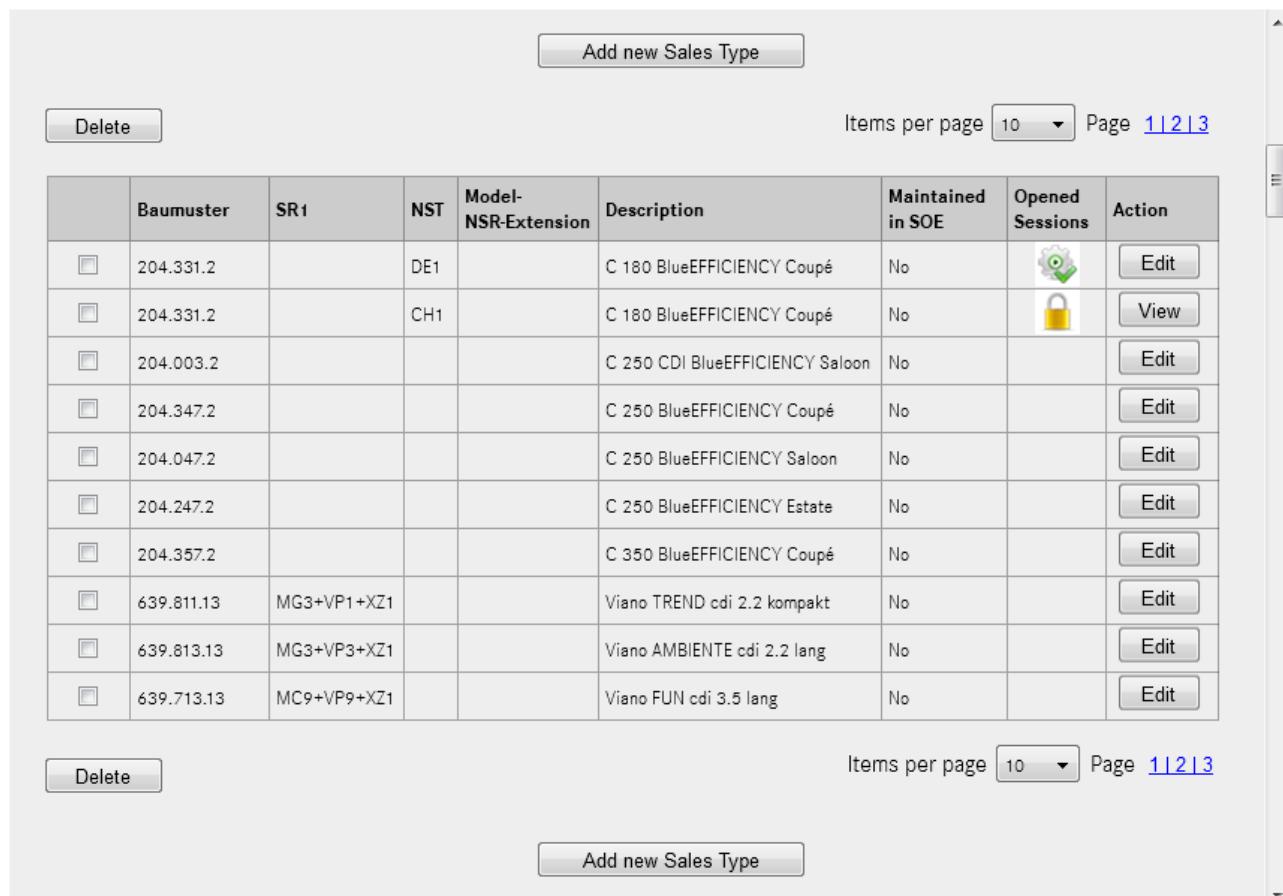
### 3.1.2.5 Dialog Elements States

Linked Label	Type	State Description
Model Series	Textbox	<u>Visible</u> : Always <u>Enabled</u> : The Textbox is only editable if in new mode.
Product group	Listbox	<u>Visible</u> : Always <u>Enabled</u> : The listbox is only editable if (ModelSeries.MaintainedInSOE="true" and the dialog is in edit-mode) or in new mode.
ReadOnlyInfo	Label	<u>Visible</u> : if dialog is in read-only mode <u>Invisible</u> : if dialog is in edit mode <u>Enabled</u> : -
All other elements except for "Cancel"	Varies	<u>Visible</u> : Always <u>Enabled</u> : if dialog is in edit mode <u>Disabled</u> : if dialog is in read-only mode
"Cancel"	Varies	<u>Visible</u> : Always <u>Enabled</u> : Always

Table 21: Dialog Elements State (DLG\_ModelSeriesDetail)

### 3.1.3 DLG\_SalesTypesOverview

This dialog shows a list of the sales types available in SOE. Sales types are imported from source system AMDS and, if necessary, can be maintained in SOE. Starting from this dialog, sales types can be added, edited or deleted.



The screenshot shows a table with the following data:

	Baumuster	SR1	NST	Model-NSR-Extension	Description	Maintained in SOE	Opened Sessions	Action
<input type="checkbox"/>	204.331.2		DE1		C 180 BlueEFFICIENCY Coupé	No		<button>Edit</button>
<input type="checkbox"/>	204.331.2		CH1		C 180 BlueEFFICIENCY Coupé	No		<button>View</button>
<input type="checkbox"/>	204.003.2				C 250 CDI BlueEFFICIENCY Saloon	No		<button>Edit</button>
<input type="checkbox"/>	204.347.2				C 250 BlueEFFICIENCY Coupé	No		<button>Edit</button>
<input type="checkbox"/>	204.047.2				C 250 BlueEFFICIENCY Saloon	No		<button>Edit</button>
<input type="checkbox"/>	204.247.2				C 250 BlueEFFICIENCY Estate	No		<button>Edit</button>
<input type="checkbox"/>	204.357.2				C 350 BlueEFFICIENCY Coupé	No		<button>Edit</button>
<input type="checkbox"/>	639.811.13	MG3+VP1+XZ1			Viano TREND cdi 2.2 kompakt	No		<button>Edit</button>
<input type="checkbox"/>	639.813.13	MG3+VP3+XZ1			Viano AMBIENTE cdi 2.2 lang	No		<button>Edit</button>
<input type="checkbox"/>	639.713.13	MC9+VP9+XZ1			Viano FUN cdi 3.5 lang	No		<button>Edit</button>

Buttons at the top: Add new Sales Type, Delete, Items per page (10), Page 1|2|3. Buttons at the bottom: Delete, Add new Sales Type, Items per page (10), Page 1|2|3.

Figure 33: DLG\_SalesTypesOverview

### 3.1.3.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all sales types (independently of the market) and sort them by SalesType.Baumuster, SalesType.NST.
“Add new Sales Type”	Button	Switch to dialog DLG_SalesTypeDetail (→ see chapter 0) for creation of a new sales type.
“Delete”	Button	After a confirmation popup, delete the selected sales types by calling AF_DeleteSalesTypes (→ see chapter 3.5.12). Display error messages in the according message popup.
“Edit”	Button	Switch to dialog DLG_SalesTypeDetail (→ see chapter 1.1.4) to edit the sales type. The dialog is opened in edit mode.
“View”	Button	Switch to dialog DLG_SalesTypeDetail (→ see chapter 0) to view the sales type. The dialog is opened in read-only mode.

Table 22: Buttons and functions (DLG\_SalesTypesOverview)

### 3.1.3.2 Form fields and front-end data objects

Field	Type	Details / Default	Business Object. Attribute
Table column “Baumuster”	Label	The baumuster of the sales type.	SalesType.Baumuster
Table column “SR1”	Label	The SR1 of the sales type.	SalesType.SR1
Table column “NST”	Label	The national sales type.	SalesType.NST
Table column “ModelNsrExtension”	Label	The ModelNsrExtensionof the sales type.	SalesType.ModelNsrExtension
Table column “Description”	Label	The description of the sales type (localized to the user’s language).	SalesTypeDescription.description
Table column “Maintained in SOE”	Label	Was the sales type imported from source system AMDS ("No") or created in SOE ("Yes").	SalesType.MaintainedInSOE
Table Column “Opened Sessions”	Label/Icon	Is the sales type being edited in a change session.	-

Table 23: Form fields and front-end data objects (DLG\_SalesTypesOverview)

### 3.1.3.3 Dialog field validation

None.

### 3.1.3.4 Configurability (incl. settings for roles)

None.

### 3.1.3.5 Dialog Elements States

Linked Label	Type	State Description
Add new Model Series, Delete	Button	<u>Visible:</u> Always <u>Enabled:</u> If there is an open change session related to the current user
All other elements	Varies	<u>Visible:</u> Always <u>Enabled:</u> Always

Table 24 Dialog Element States (DLG\_SalesTypesOverview)

### 3.1.4 DLG\_SalesTypeDetail

This dialog allows to add new sales types or to edit and to view existing ones.

Furthermore, the translations of sales type descriptions can be maintained using this dialog.

The screenshot shows the DLG\_SalesTypeDetail dialog box. At the top, there are fields for 'Baumuster\*' (set to 2042312), 'NST' (empty), 'Model Series\*' (set to 204), and 'Body Type\*' (set to Estate). Below these, a message states: "This Sales Type is exclusively maintained in SOE." The main area contains two tables. The left table, titled 'Description', maps languages to translations for a sales type. The right table, titled 'Valid for markets', lists countries with checkboxes indicating which are valid for that market. At the bottom are 'Save' and 'Cancel' buttons.

Language	Translation
German (Germany)	C 180 T-Modell RL
German (Austria)	C 180 T-Modell RL
German (Switzerland)	
German (Belgium)	C 180 T-Modell RL
German (Luxembourg)	C 180 T-Modell RL
German (Netherlands)	
English (United Kingdom)	C 180 Estate
English (Ireland)	C 180 estate RHD
English (Hungary)	
English (Netherlands)	
French (France)	C 180 Station Wagon cond. à droite
French (Belgium)	C 180 Station Wagon cond. à droite
French (Luxembourg)	C 180 Station Wagon cond. à droite
Italian (Italy)	
Spanish (Spain)	
Czech (Czech Republic)	C 180 BlueEFFICIENCY kombi RL
Polish (Poland)	
Slovakian (Slovakia)	

Valid for markets	
<input checked="" type="checkbox"/>	Austria (GS0008201)
<input checked="" type="checkbox"/>	Belgium
<input checked="" type="checkbox"/>	Czech Republic
<input checked="" type="checkbox"/>	France (GS0008204)
<input checked="" type="checkbox"/>	Germany (GS0008205)
<input type="checkbox"/>	Hungary
<input checked="" type="checkbox"/>	Ireland
<input checked="" type="checkbox"/>	Italy (GS0008206)
<input checked="" type="checkbox"/>	Luxembourg
<input type="checkbox"/>	Netherlands (GS0008208)
<input type="checkbox"/>	Poland
<input checked="" type="checkbox"/>	Slovakia
<input type="checkbox"/>	Spain (GS0008209)
<input type="checkbox"/>	Switzerland (GS0008211)
<input checked="" type="checkbox"/>	United Kingdom (GS0008212)

Figure 34: DLG\_SalesTypeDetail for passenger cars

Language	Translation
German (Germany)	Viano TREND cdi 2.2 kompakt
German (Austria)	Viano TREND cdi 2.2 kompakt
German (Switzerland)	
German (Belgium)	Viano TREND cdi 2.2 kompakt
German (Luxembourg)	Viano TREND cdi 2.2 kompakt
German (Netherlands)	
English (United Kingdom)	Viano TREND cdi 2.2 plane van
English (Ireland)	Viano TREND cdi 2.2 plane van
English (Hungary)	
English (Netherlands)	
French (France)	
French (Belgium)	
French (Luxembourg)	
Italian (Italy)	
Spanish (Spain)	
Czech (Czech Republic)	
Polish (Poland)	
Slovakian (Slovakia)	

Valid for markets

- Austria (GS0008201)
- Belgium
- Czech Republic
- France (GS0008204)
- Germany (GS0008205)
- Hungary
- Ireland
- Italy (GS0008206)
- Luxembourg
- Netherlands (GS0008208)
- Poland
- Slovakia
- Spain (GS0008209)
- Switzerland (GS0008211)
- United Kingdom (GS0008212)

Save      Cancel

Figure 35: DLG\_SalesTypeDetail for transporters

### 3.1.4.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Edit mode: Load the sales type and its translations. Merge the translations with all maintainable languages available in the application configuration. Also load all available markets and sort them by OrgUnit.Name.  New mode: Load the maintainable languages from the application configuration and generate the table rows. Also load the available markets and sort them by OrgUnit.Name.

Linked label / button labeling	Type	Action description
“Save”	Button	Save the sales type (→ see AF_SaveSalesType) and switch to DLG_SalesTypesOverview (→ see chapter 3.1.3).
“Cancel”	Button	Discard the changes and switch to DLG_SalesTypesOverview (→ see chapter 3.1.3).

Table 25: Buttons and functions (DLG\_SalesTypeDetail)

### 3.1.4.2 Form fields and front-end data objects

Linked Label	Type	Details / Default	Business Object. Attribute
Baumuster	Textbox	The baumuster of the sales type.	SalesType.Baumuster
NST code	Textbox	The national sales type.	SalesType.NST
SR1	Textbox	The SR1 of the sales type.	SalesType.SR1
ModelNsrExtension	Textbox	The ModelNsrExtension of the sales type.	SalesType.ModelNsrExtension
Model Series	Dropdown	The model series the sales type is assigned to.	ModelSeries.modelSeriesId
Body Type	Dropdown	The body type the sales type is assigned to.	BodyTypeDescription.description
Maintained in SOE	Label	Is the sales type exclusively maintained in SOE or was it imported from AMDS.	Text: If SalesType.maintainedInSOE = "true", then display: "This sales type is exclusively maintained in SOE." Otherwise, i.e. if SalesType.maintainedInSOE = "false", then display: "This model series was imported from AMDS."
Table column “Language”	Label	Locale of the maintained translation.	-
Table column “Translation”	Textbox	The (localized) description of the sales type.	SalesTypeDescription.description
Valid for markets	Check-box list	Optionally assign the sales type to specific markets (this information will be ignored by the rules evaluation).	Texts: OrgUnit.Name (OrgUnit.OutletId)  Checked: The markets the sales type is valid for as given by association "is valid for".  Available values: Calls IIF_GetMarketsForContinent to retrieve the available markets.

Table 26: Form fields and front-end data objects (DLG\_SalesTypeDetail)

### 3.1.4.3 Dialog field validation

Linked Field	Validation	Error message
Baumuster	A value must be provided.	VEHPRO_008. Use the screen field name as parameter 1.
	The Baumuster must be provided in the format xxx.xxx.x. X must be numeric.	VEHPRO_009
	The leading characters of the Baumuster must match the chosen model series.	VEHPRO_024
SR1	One or both of these values must be provided iff the selected model series has the product group 'T' (transporter).	VEHPRO_025. Use "SR1" and "ModelNsrExtension" as parameters of the error message.
ModelNsrExtension		

Table 27: Dialog field validations (DLG\_SalesTypeDetail)

### 3.1.4.4 Configurability (incl. settings for roles)

None.

### 3.1.4.5 Dialog Elements States

Linked Label	Type	State Description
Baumuster	Textbox	Visible: Always Enabled: The Textbox is only editable in new mode.
NST	Textbox	Visible: If a model series is selected and its product group is 'P'. Enabled: The Textbox is only editable in new mode.
SR1	Textbox	Visible: If a model series is selected and its product group is 'T'. Enabled: The Textbox is only editable in new mode.
ModelNsrExtension	Textbox	Visible: If a model series is selected and its product group is 'T'. Enabled: The Textbox is only editable in new mode.
Model Series	Listbox	Visible: Always Enabled: The Listbox is only editable in new mode.
Body Type	Listbox	Visible: Always Enabled: The Listbox is only editable in new mode.
ReadOnlyInfo	Label	Visible: if dialog is in read-only mode Invisible: if dialog is in edit mode Enabled: -
All other elements except for "Cancel"	Varies	Visible: Always Enabled: if dialog is in edit mode Disabled: if dialog is in read-only mode
"Cancel"	Varies	Visible: Always Enabled: Always

Table 28 Dialog Elements State (DLG\_SalesTypeDetail)

## 3.1.5 DLG\_BodyTypesOverview

This dialog shows a list of the body types available in SOE. Body types are imported from source system AMDS and, if necessary, can be maintained in SOE. Starting from this dialog, body types can be added, edited or deleted.

Body Types Overview						
	ID	Product group	Description	Maintained in SOE	Opened Sessions	Action
<input type="checkbox"/>	0	P	MPV	No		<button>Edit</button>
<input type="checkbox"/>	1	P	Cabriolet/Roadster	No		<button>View</button>
<input type="checkbox"/>	3	P	Coupé	No		<button>Edit</button>
<input type="checkbox"/>	4	P	Saloon	No		<button>Edit</button>
<input type="checkbox"/>	5	P	Estate	No		<button>Edit</button>
<input type="checkbox"/>	6	P	SUV	No		<button>Edit</button>
<input type="checkbox"/>	7	P	Shooting Brake	No		<button>Edit</button>
<input type="checkbox"/>	FKA	T	Kastenwagen	Non		<button>Edit</button>

Figure 36: DLG\_BodyTypesOverview

### 3.1.5.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all body types and sort them by BodyType.ID.
"Add new Body Type"	Button	Switch to dialog DLG_BodyTypeDetail (see chapter 0) for creation of a new body type.
"Delete"	Button	After a confirmation popup, delete the selected body types by calling AF_DeleteBodyTypes (see chapter 3.5.14). Display error messages in the according message popup.
"Edit"	Button	Switch to dialog DLG_BodyTypeDetail (see chapter 0) to edit the body

Linked label / button labeling	Type	Action description
		type. The dialog is opened in edit mode.
“View”	Button	Switch to dialog DLG_SalesTypeDetail (see chapter 1.1.4) to view the sales type. The dialog is opened in read-only mode.

Table 29: Buttons and functions (DLG\_BodyTypesOverview)

### 3.1.5.2 Form fields and front-end data objects

Field	Type	Details / Default	Business Object. Attribute
Table column “ID”	Label	The ID of the body type.	BodyType.bodyTypId
Table column “Product Group”	Label	The product group of the body type.	BodyType.ProductGroup
Table column “Description”	Label	The description of the body type (localized to the user’s language).	BodyTypeDescription.description
Table column “Maintained in SOE”	Label	Was the body type imported from source system AMDS (“No”) or created in SOE (“Yes”).	BodyType.MaintainedInSOE
Table Column “Opened Sessions”	Label/Icon	Is the body type being edited in a change session.	-

Table 30: Form fields and front-end data objects (DLG\_BodyTypesOverview)

### 3.1.5.3 Dialog field validation

None.

### 3.1.5.4 Configurability (incl. settings for roles)

None.

### 3.1.5.5 Dialog Elements States

Linked Label	Type	State Description
Add new Model Series, Delete	Button	<u>Visible:</u> Always <u>Enabled:</u> If there is an open change session related to the current user.
All other elements	Varies	<u>Visible:</u> Always <u>Enabled:</u> Always

Table 31 Dialog Element States (DLG\_BodyTypeOverview)

## 3.1.6 DLG\_BodyTypeDetail

This dialog allows to create (add) a new body type or edit or view the details of an existing body type.

When creating or editing, the user enters a bodyTypId.

Further, the translations of body type descriptions can be maintained using this dialog.

Body Type ID\*  Product group\*

This Body Type is exclusively maintained in SOE.

Description	
Language	Translation
<b>German (Germany)</b>	T-Modell
German (Austria)	
German (Switzerland)	
<b>English (United Kingdom)</b>	Estate
English (Switzerland)	
<b>French (France)</b>	Commerciale
French (Belgium)	
French (Switzerland)	
<b>Italian (Italy)</b>	Familiare

Figure 37: DLG\_BodyTypesDetail

### 3.1.6.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Edit mode: Load the body type and its translations. Merge the translations with all maintainable languages available in the application configuration.  New mode: Load the maintainable languages from the application configuration and generate the table rows.
“Save”	Button	Save the body type (see AF_SaveBodyType) and switch to DLG_BodyTypesOverview (see chapter 3.1.5).
“Cancel”	Button	Discard the changes and switch to DLG_BodyTypesOverview (see chapter 3.1.5).

Table 32: Buttons and functions (DLG\_BodyTypesDetail)

### 3.1.6.2 Form fields and front-end data objects

Linked Label	Type	Details / Default	Business Object. Attribute
Body Type ID	Textbox	The ID of the body type.	BodyType.bodyTypeld
Table column "Product Group"	Listbox	The product group of the body type.	BodyType.ProductGroup
Table column "Language"	Label	Locale of the maintained translation.	-
Table column "Translation"	Textbox	The (localized) description of the body type.	BodyTypeDescription.description

Table 33: Form fields and front-end data objects (DLG\_BodyTypesDetail)

### 3.1.6.3 Dialog field validation

Linked Field	Validation	Error message
Body Type ID	A value must be provided.	VEHPRO_008. Use the screen field name as parameter 1.
Product Group	A value must be provided.	VEHPRO_008. Use the screen field name as parameter 1.

Table 34: Dialog field validations (DLG\_BodyTypesDetail)

### 3.1.6.4 Configurability (incl. settings for roles)

None.

### 3.1.6.5 Dialog Elements States

Linked Label	Type	State Description
"Body Type ID"	Textbox	Visible: Always Enabled: The Textbox is only editable in new mode.
ReadOnlyInfo	Label	Visible: if dialog is in read-only mode Invisible: if dialog is in edit mode Enabled: -
All other elements except for "Cancel"	Varies	Visible: Always Enabled: if dialog is in edit mode Disabled: if dialog is in read-only mode
"Cancel"	Varies	Visible: Always Enabled: Always

Table 35 Dialog Elements States

## 3.1.7 DLG\_EquipmentCodesOverview

This dialog lists all MBconnect relevant equipment codes for SOE. For each equipment code the dialog displays the code id and the equipment code description (localized to the user's language).

Additionally the dialog gives the information if an equipment code is relevant for maintaining any service assignment rule and if the code is exclusively maintained in SOE or imported by AMDS.

The dialogue allows viewing details of each available equipment code.

<div style="text-align: center;"> <input type="button" value="Add new Equipment"/> </div>								
<input type="button" value="Delete"/>			Items per page <input type="button" value="10"/> Page <a href="#">1</a> <a href="#">2</a> <a href="#">3</a>					
	Code	Product group	Description	Relevant for service assignment rules	Maintained in SOE	Opened Sessions	Action	
<input type="checkbox"/>	510	P	Audio 20 CD/Radio - integrated 6 disc CD changer	Yes	No		<input type="button" value="Edit"/>	
<input type="checkbox"/>	523	P	Audio 20 CD/Radio - single CD drive (MP3 compatible)	Yes	No		<input type="button" value="View"/>	
<input type="checkbox"/>	512	P	COMAND Online system with Media Interface and 6-disc changer	Yes	No		<input type="button" value="Edit"/>	
<input type="checkbox"/>	H12	P	Holzausfuehrung Wurzelnuss hell	Yes	Yes		<input type="button" value="Edit"/>	
<input type="checkbox"/>	H12	T	Warwasserzusatzeitung	Yes	Yes		<input type="button" value="Edit"/>	

<div style="text-align: center;"> <input type="button" value="Delete"/> </div>								
<div style="text-align: center;">         Items per page    <input type="button" value="10"/>    Page <a href="#">1</a> <a href="#">2</a> <a href="#">3</a> </div>								
<div style="text-align: center;"> <input type="button" value="Add new Equipment"/> </div>								

Figure 38: DLG\_EquipmentCodesOverview

### 3.1.7.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all available equipment codes together with the permitted actions for each loaded code by calling the loading algorithm described in Chapter 2.3.14.2 ("Loading of master data elements") (see details in subchapter "Dialog Elements States") and sort them by Equipment.Code.
"Add new Equipment"	Button	Switches to the dialog DLG_EquipmentCodeDetail (→ see chapter 0) to create a new equipment code.
"Delete"	Button	After displaying a confirmation popup delete the selected equipment codes by calling the AF_DeleteEquipmentCodes (→ see chapter 3.5.16). Display potential error messages in the according message popup.
"Edit" (1..n)	Button	Switches to the dialog DLG_EquipmentCodeDetail (→ see chapter 0) to edit or edit the equipment code in detail. The dialog is opened in edit mode.
"View" (1..n)	Button	Switches to the dialog DLG_EquipmentCodeDetail (→ see chapter 0) to view the equipment code in detail. The dialog is opened in read-only

Linked label / button labeling	Type	Action description
		mode.

Table 36: Buttons and functions

### 3.1.7.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Table column "Code"	Label	The equipment code.	Equipment.Code
Table column "Product group"	Label	The product group of the respective equipment.	Equipment.ProductGroup
Table column "Description"	Label	The description of the equipment code (localized to the user's language).	EquipmentDescription.Description
Table column "Relevant for service assignment rule"	Label	States if the equipment code is relevant for maintaining any service assignment rule.	Equipment.RelevantForServiceAssignmentRule
Table column "Maintained in SOE"	Label	States if the equipment code is exclusively maintained in SOE or has been imported by AMDS.	Equipment.MaintainedInSOE
Table Column "Opened Sessions"	Label/Icon	Indicates whether the entity Equipment Codes is edited inside a change session.	-

Table 37: Form fields and front-end data objects

### 3.1.7.3 Dialog field validation

None.

### 3.1.7.4 Configurability (incl. setting for roles)

None.

### 3.1.7.5 Dialog Elements States

Linked Label	Type	State Description
Add new Model Series, Delete	Button	<u>Visible:</u> Always <u>Enabled:</u> If there is an open change session related to the current user.
All other elements	Varies	<u>Visible:</u> Always <u>Enabled:</u> Always

Table 38 Dialog Element States (DLG\_EquipmentCodesOverview)

## 3.1.8 DLG\_EquipmentCodeDetail

This dialog allows the user to create, view or edit an equipment code in detail.

The dialog shows the translations of the equipment code descriptions. Only these translation texts can be modified that have not been imported by AMDS but are maintained in SOE.

Additional a checkbox can be selected if the equipment code is relevant for maintaining the service assignment rules.

Equipment codes, for which the checkbox is selected, are available for maintaining the service assignment rules and are displayed on DLG\_ServiceAssignmentRulesDetails.

The screenshot shows a dialog box titled 'DLG\_EquipmentCodeDetail'. At the top, there are two input fields: 'Code\*' containing 'H12' and 'Product group\*' containing 'P' with a dropdown arrow. Below these is a note: 'This equipment code has been imported by AMDS'. A checkbox labeled 'This equipment code is relevant for maintaining SOE Service Assignment Rules' is checked. Under the heading 'Description', there is a table with two columns: 'Language' and 'Translation'. The table lists translations for German (Germany), German (Austria), German (Switzerland), English (United Kingdom), English (Switzerland), French (France), French (Belgium), French (Switzerland), Italian (Italy), and Italian (Switzerland). At the bottom of the dialog are 'Save' and 'Cancel' buttons.

Language	Translation
German (Germany)	Holzausführung Wurzelnuss hell
German (Austria)	
German (Switzerland)	
English (United Kingdom)	
English (Switzerland)	
French (France)	
French (Belgium)	
French (Switzerland)	
Italian (Italy)	
Italian (Switzerland)	

Figure 39: DLG\_EquipmentCodeDetail

### 3.1.8.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		<p><u>Edit mode:</u></p> <p>Load the equipment code and all available translations of the code descriptions. Use the loading algorithm described in Chapter "Loading of master data elements". Merge the available translations with all maintainable languages, which are available in the application configuration.</p> <p><u>New mode:</u></p> <p>Load the maintainable languages from the application configuration and generate the table rows.</p>

Linked label / button labeling	Type	Action description
“Save”	Button	Save the currently created equipment code (new mode) or the current changes of the equipment code by calling AF_SaveEquipment (→ see chapter 3.5.15) and navigate back to DLG_EquipmentCodeDetail (→ see chapter 0). Display potential error messages in the according message popup.
“Cancel”	Button	Discard all changes and navigate back to DLG_EquipmentCodesOverview (→ see chapter 0).

Table 39: Buttons and functions

### 3.1.8.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Code	Textbox	The equipment code.	Equipment.Code
Product group	Listbox	The product group of the equipment.	Equipment.ProductGroup
Table column “Language”	Label	Locale of the maintained translation.	-
Table column “Translation”	Textbox	The description of the equipment (localized).	EquipmentDescription.Description
Relevant for service assignment rules	Checkbox	States if the equipment code is relevant for maintaining any service assignment rule.	Equipment.RelevantForServiceAssignmentRules
Maintained in SOE	Label	Information if an equipment code is exclusively maintained in SOE or has been imported by AMDS.	Text: If Equipment.maintainedInSOE="true" display: “This equipment code is exclusively maintained in SOE.” Otherwise if Equipment.maintainedInSOE="false" display: “This equipment code has been imported by AMDS.”
ReadOnlyInfo	Label	Presents this information: “The dialog is opened in read-only mode.”	-

Table 40: Form fields and front-end data objects

### 3.1.8.3 Dialog field validation

Linked Field	Validation	Errormessage
Code	A value must be provided.	VEHPRO_008. Use the screen field name as parameter 1.
	The code must be alpha numeric	VEHPRO_010.
Product group (only in new mode)	A value must be provided.	VEHPRO_008. Use the screen field name as placeholder 1.

Table 41: Dialog field validations

### 3.1.8.4 Configurability (incl. setting for roles)

None.

### 3.1.8.5 Dialog Elements States

Linked Label	Type	State Description
Code	Textbox	<u>Editable:</u> The Textbox is only editable if Equipment.MaintainedInSOE="true" and in new mode.
Product group	Listbox	<u>Visible:</u> Always <u>Enabled:</u> The Textbox is only editable in new mode.
Translation	Textbox	<u>Editable:</u> The textbox of a localized description is only editable if EquipmentDescription.MaintainedInSOE="true" and in new mode.
ReadOnlyInfo	Label	<u>Visible:</u> if dialog is in read-only mode <u>Invisible:</u> if dialog is in edit mode
all other elements except for "Cancel"	Varies	<u>Visible:</u> Always <u>Enabled:</u> if dialog is in edit mode <u>Disabled:</u> if dialog is in read-only mode
All other elements	Varies	<u>Visible:</u> Always <u>Enabled:</u> Always

Table 42 Dialog Elements State (DLG\_EquipmentCodesDetail)

## 3.2 External View - Offered Interfaces

None.

## 3.3 External View - Consumed Interfaces

### 3.3.1 IF\_AMDS\_GetSalesTypeGroups

**Communication type:** Synchronously

This interface calls AMDS in order to retrieve all available sales type groups for the given market.

#### 3.3.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
market number	Mand.	String	3	-	Example: "571" for CH	The unique market number of the market for which the export is generated. A matching between market number and market can be found in the application configuration.
ProductGroup	Mand.	Product GroupEnum		-	P, T	The product group of the sales types groups to be retrieved.

Table 43: External interface input (IF\_AMDS\_GetSalesTypeGroups)

#### 3.3.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of Sales type groups (all following entries exist per returned sales type group)</b>						

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
salesTypeGroupValue	Mand.	String	5	ModelSeries.ModelSeriesID/BodyType.BodyTypeID	Example: „C“ oder „204“	The code of the sales type group.
sparte	Opt.	String	1	-	Example: „0“ for PKW/Smart	The division of the sales type group.
class	Mand.	String	5	-	Example: „BR“ for model series or “VKLA” for “Verkaufsklasse”	The class of the sales type group.
• List of sales type group Descriptions (inner list per modelSeriesID)						
description	Opt.	String	100	-	Example: „C-Class“ (for VKLA)	The localized description of the sales type group.
lang	Mand.	String	2	-	Example: „de“	The ISO-code for the language of the localized description.
• List of SalesTypes (inner list per modelSeriesID)						
Baumuster	Mand.	BaumusterDT		SalesType.Baumuster	Example: „2042312“	The uniquebaumuster id of the sales type.
NST	Opt.	NstDT		SalesType.Nst	Example: „CH1“	The NST code of the sales type (for passenger cars, i.e. product group ‘P’)
SR1	Opt.	Sr1DT		SalesType.Sr1	Example: “MG3+VP1+XZ1”	The SR1 of the sales type (for transporters, i.e. product group ‘T’).
ModelNsrExtension	Opt.	ModelNsrExtensionDT		SalesType.ModelNsrExtension	„BAD“	The ModelNsrExtension of the sales type (for transporters, i.e. product group ‘T’).

Table 44: External interface output (IF\_AMDS\_GetSalesTypeGroups)

### 3.3.1.3 Exceptions

None.

## 3.3.2 IF\_AMDS\_GetSalesTypes

**Communication type:** Synchronous

This interface calls AMDS in order to retrieve all available sales types and their descriptions for the given market.

### 3.3.2.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
market number	Mand.	String	3	-	Example: “571” for CH	The unique market number of the market for which the export is generated. A matching between

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						market number and market can be found in the application configuration.
ProductGroup	Mand.	ProductGroupEnum		-	P, T	The product group the sales types groups to be retrieved.

Table 45: External interface input (IF\_AMDS\_GetSalesTypes)

### 3.3.2.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of SalesTypes</b>						
Baumuster	Mand.	BaumusterDT		SalesType.Baumuster	Example: „2042312“	The unique baumuster id of the sales type.
NST	Opt.	NstDT		SalesType.Nst	Example: „CH1“	The NST code of the sales type (for passenger cars, i.e. product group ‘P’).
SR1	Opt.	Sr1DT		SalesType.Sr1	Example: “MG3+VP1+XZ1”	The SR1 of the sales type (for transporters, i.e. product group ‘T’).
ModelNsrExtension	Opt.	ModelNsrdExtensiionDT		SalesType.ModelNsrExtension	„BAD“	The ModelNsrExtension of the sales type (for transporters, i.e. product group ‘T’).
sparte	Opt.	String	1	-	Example: „0“ for PKW/Smart	The division of the sales type group.
<b>• List of SalesTypeDescriptions (for each SalesType)</b>						
description	Opt.	String	100	SalesTypeDescription.description	Example: „C-Class“ (for VKLA)	The localized description of the sales type group.
lang	Mand.	String	2		Example: „de“	The ISO-code for the language of the localized description.

Table 46: External interface output (IF\_AMDS\_GetSalesTypes)

### 3.3.2.3 Exceptions

None.

## 3.3.3 IF\_AMDS\_getCodes

**Communication type:** Synchronously

This interface calls AMDS in order to retrieve all available equipment codes for the given market.

### 3.3.3.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Market number	Mand.	String	3	-	Example: “571” for CH	The unique market number of the market for which the

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						export is generated. A matching between market number and market can be found in the application configuration.
ProductGroup	Mand.	Product GroupEnum	-	P, T	The product group the sales types groups to be retrieved.	

Table 47: External interface input (IF\_AMDS\_getCodes)

### 3.3.3.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of Equipment (all following entries exist per returned equipment code)</b>						
codeId	Mand.	String	3	Equipment.Code	Example: „05U“	The unique id of the equipment code.
division	Mand.	String	1	-	Example: „0“ for PKW/Smart	The division of the equipment code.
codeCategory	Mand.	String	2	-	Example: „SA“	The equipment code category.
productGroup	Opt.	String	2	-	Example: „P“ for PKW	The product group of the code.
<b>• List of Code Descriptions (inner list per codeId)</b>						
description	Mand.	String	100	EquipmentDescription.Description	Example: „Life Traffic“	The localized description of the equipment code.
language	Mand.	String	2	-	Example: „de“	The ISO-code for the language of the localized description.

Table 48: External interface output (IF\_AMDS\_getCodes)

### 3.3.3.3 Exceptions

None.

## 3.3.4 IF\_ODC\_GetVehicleData

### 3.3.4.1 General Description

**Communication type:** Synchronously

This interface returns the vehicle data including description texts for the given list of vehicles. The vehicle data comprises the configurations and technical information for the vehicles. Additionally, descriptive texts are returned for each vehicle's configuration. Each vehicle is identified by its FIN or VIN. The language of the descriptions is determined based on the locale and country that are given as input parameters.

This interface is implemented by AF\_GetVehicleData.

### 3.3.4.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Locale	M	String	5	As defined by calling AF.	Language and country	Locale to retrieve the vehicle and equipment texts in.

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
					ISO code, e.g. de_DE	
Country	M	String	2	As defined by calling AF.	Country ISO code, e.g. DE, GB, IS...	Country to retrieve the vehicle and equipment texts for. Background: The texts maintained for a vehicle for the locale "de_DE" in the country "DE" might be different than the texts for "de_DE" in the country "CH".
RequestedData	M	Enum	-	-	ENUM with possible values: TECHNICAL, SALES, BOTH, ANY	Indicates what kind of data is requested (TECHNICAL=data from VeDoc, SALES=data from UVS, BOTH=data from both systems, ANY=data from any of the two systems)
<b>List of vehicles to retrieve data.</b>						
FinOrVin	M	String	17	As defined by calling AF.	e.g. „WDD2043311G122127“	The vehicle identifier number. May be a European FIN or the American VIN.

Table 49: IF\_ODC\_GetVehicleData input

### 3.3.4.3 Output

The output is structured in three sections, each coloured differently as described in Table 50:

Colour	Description
Light Blue	Section containing vehicle information available in both systems, UVS and VeDoc.
Light Red	Section containing sales-related data only available in UVS.
Light Green	Section containing technical data solely available in VeDoc.

Table 50: Colour coding of the output table

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
finOrVin	M	String	17	-	e.g. „WDD20433111G122127“	Search strings that the consumer sent in as "finOrVin". Needed for the interface consumer to match the results with his request.
Common vehicle information per vehicle (only provided if sales-related or technical data regarding the vehicle was found)						
FIN	M	String	17	VehicleCoreData.FIN	e.g. „WDD20433111G122127“	The FIN of the requested vehicle.
VIN	O	String	17	VehicleCoreData.VIN	e.g. „WDDGJ3BB9DG122127“	The VIN of the requested vehicle (if available).
NST	O	String	3	ODC::VehicleConfiguration.NST	e.g. "DE1"	The NST code.
ModelYearCode	O	String	5	ODC::VehicleConfiguration.ModelYearCode	Example: "804"	The code of the model year. <u>Mapping:</u> Internally, add this code to the set of equipments of the vehicle.
List of change year codes per vehicle (there can be 0 to n change year codes) – optional						
ChangeYear-Code	M	String	5	ODC::VehicleConfiguration.ChangeYe	e.g. "056"	Code (partially) describing a vehicle's change year.

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
				arCode		
Origin	O	Enum	-	-	Example: "SALES"	Only available for a merged list of change year codes. Value set to: "SALES" if code only available in system providing sales data, "TECHNICAL" if code only available in system providing technical data "BOTH" if code available in both systems.
List of Equipment per vehicle (there can be 0 to n equipment codes).						
Please note that the equipment codes for paint, upholstery and line are not available if sales data is requested/returned. In this case, information regarding paint, upholstery and line are only available via the dedicated attributes Paint, Line and Upholstery which are available as part of the sales-related data.						
Code	M	String	5	ODC::VehicleConfiguration.Equipment.Code	Example: "527"	The equipment code.
Origin	O	Enum	-	-	Example: "SALES"	Only available for a merged list of equipment codes. Value set to: "SALES" if code only available in system providing sales data, "TECHNICAL" if code only available in system providing technical data "BOTH" if code available in both systems.
Description	O	String	254	ODC::VehicleConfiguration.Equipment.Description	Example: "COMAND Online system with Media Interface"	The description of the equipment in the requested locale, if available. Only available if sales-related data is returned.
List of TechnicalData per vehicle (there can be 0 to n codes)						
Omit the technical data						
Sales-related vehicle information per vehicle (only provided if sales-related information was found)						
Baumuster	O	String	7	ODC::VehicleConfiguration.Baumuster	e.g. "123.456.7"	The baumuster.
BaumusterDescription	O	String	254	ODC::VehicleConfiguration.Description	"E 270 CDI Limousine ELEGANCE"	The baumuster description (localized).
Omit final inspection date						
SalesDateTime-stamp	O	TimeStampDT	-	-	-	Timestamp indicating if and when the sales-related vehicle data was fetched. The timestamp is comprised of date and time information: YYYY-MM-DD HH:MM:SS
Omit "salesDataAvailable"						
Line						
Omit the line						
Paint						
Paint1Code	O	String	5	ODC::VehicleConfiguration.Paint1.Code	e.g. „123“	The code of paint 1.
Paint1Description	O	String	254	ODC::VehicleConfiguration.Paint1.Description	e.g. "Black metallic"	The description of paint 1 (localized).
Paint2Code	O	String	5	ODC::VehicleConfiguration.Paint2.Code	e.g. „123“	The code of paint 2.
Paint2Description	O	String	254	ODC::VehicleConfigu	e.g. "Black"	The description of paint 2

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
on				guration.Paint2.Description	metallic”	(localized).
<b>Upholstery</b>						
UpholsteryCode	O	String	5	ODC::VehicleConfiguration.Upholstery.Code	e.g. „123“	The code of the upholstery.
UpholsteryDescription	O	String	254	ODC::VehicleConfiguration.Upholstery.Description	e.g. “Grey fabric”	The description of the upholstery (localized)
Technical vehicle information per vehicle (only provided if technical information was found)						
ConsumerCountry	O	String	2	ODC::VehicleConfiguration.ConsumerCountry	Example: “DE”. Values are stored according to ISO-3166 AL-PHA-2.	The country the vehicle is built for.
TechnicalData-Timestamp	M	TimeStampDT	-	-		Timestamp indicating if and when the technical vehicle data was fetched. The timestamp is comprised of date and time information: YYYY-MM-DD HH:MM:SS
Omit all other technical vehicle information						
List of serial numbers (0..*)						
Omit the serial numbers						
List of control units (0..*)						
Omit the control units						

Table 51: IF\_ODC\_GetVehicleData output

### 3.3.4.4 Exceptions

None.

## 3.3.5 IF\_ODC\_GetVehicleDataWithoutLocale

### 3.3.5.1 General Description

**Communication type:** Synchronous

This interface returns the vehicle data excluding descriptions for the given list of vehicles, i.e. codes describing their model and equipment. Each vehicle is identified by its FIN or VIN.

This interface is implemented by AF\_FetchVehicleDataWithoutLocale.

### 3.3.5.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of vehicles to retrieve data.</b>						
FinOrVin	M	String	17	As defined by calling AF.	e.g. „WDD2043311G122127“	The vehicle identifier number. May be a European FIN or the American VIN.
RequestedData	M	Enum	-	-	ENUM with possible values: TECHNICAL, SALES, BOTH, ANY	Indicates what kind of data is requested (TECHNICAL=data from VeDoc, SALES=data from UVS, BOTH=data from both systems, ANY=data from any of the two systems)

Table 52: IF\_ODC\_GetVehicleDataWithoutLocale input

### 3.3.5.3 Output

The output is structured in three sections, each coloured differently as described in Table 53:

Colour	Description
Light Blue	Section containing vehicle information available in both systems, UVS and VeDoc.
Light Red	Section containing sales-related data only available in UVS.
Light Green	Section containing technical data solely available in VeDoc.

Table 53: Colour coding of the output table

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
finOrVin	M	String	17	-	e.g. „WDD2043311 G122127“	Search strings that the consumer sent in as “finOrVin”. Needed for the interface consumer to match the results with his request.
Common vehicle information per vehicle (only provided if sales-related or technical data regarding the vehicle was found)						
FIN	M	String	17	VehicleCoreData.FIN	e.g. „WDD2043311 G122127“	The FIN of the requested vehicle.
VIN	O	String	17	VehicleCoreData.VIN	e.g. “WDDGJ3BB 9DG122127”	The VIN of the requested vehicle (if available).
NST	O	String	3	ODC::VehicleBasic Configuration.NST	e.g. “DE1”	The NST code.
ModelYearCode	O	String	5	ODC::VehicleBasic Configuration.ModelYearCode	Example: “804”	The code of the model year. <u>Mapping:</u> Internally, add this code to the set of equipments of the vehicle.
List of change year codes per vehicle (there can be 0 to n change year codes) – optional						
ChangeYear-Code	M	String	5	ODC::VehicleConfiguration.ChangeYearCode	e.g. “056”	Code (partially) describing a vehicle’s change year.
Origin	O	Enum	-	-	Example: “SALES”	Only available for a merged list of change year codes. Value set to: “SALES” if code only available in system providing sales data, “TECHNICAL” if code only available in system providing technical data “BOTH” if code available in both systems.
List of Equipment per vehicle (there can be 0 to n equipment codes).						
Please note that the equipment codes for paint, upholstery and line are not available if sales data is requested/returned. In this case, information regarding paint, upholstery and line are only available via the dedicated attributes Paint, Line and Upholstery which are available as part of the sales-related data.						
Code	M	String	5	ODC::VehicleBasic Configuration.Equipment.Code	Example: “527”	The equipment code.
Origin	O	Enum	-	-	Example: “SALES”	Only available for a merged list of equipment codes. Value set to: “SALES” if code only available in system providing sales data, “TECHNICAL” if code

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						only available in system providing technical data “BOTH” if code available in both systems.
List of TechnicalData per vehicle (there can be 0 to n codes)						
Omit the technical data						
Sales-related vehicle information per vehicle (only provided if sales-related information was found)						
Omit final inspection date						
Baumuster	O	String	7	ODC::VehicleBasic Configuration.Baumuster	e.g. „123.456.7”	The baumuster.
SalesDateTime-Stamp	O	TimeStampDT	-	-		Timestamp indicating if and when the sales-related vehicle data was fetched. The timestamp is comprised of date and time information: YYYY-MM-DD HH:MM:SS
Omit “salesDataAvailable”						
Line						
Omit the line						
Paint						
Paint1Code	O	String	5	ODC::VehicleBasic Configuration.Paint1.Code	e.g. „123“	The code of paint 1.
Paint2Code	O	String	5	ODC::VehicleBasic Configuration.Paint2.Code	e.g. „123“	The code of paint 2.
Upholstery						
UpholsteryCode	O	String	5	ODC::VehicleBasic Configuration.Upholstery.Code	e.g. „123“	The code of the upholstery.
Technical vehicle information per vehicle (only provided if technical information was found)						
ConsumerCountry	O	String	2	ODC::VehicleBasic Configuration.ConsumerCountry	Example: “DE”. Values are stored according to ISO-3166 AL-PHA-2.	The country the vehicle is built for.
TechnicalData-Timestamp	O	TimeStampDT	-	-		Timestamp indicating if and when the technical vehicle data was fetched. The timestamp is comprised of date and time information: YYYY-MM-DD HH:MM:SS
Omit all other technical vehicle information						
List of serial numbers (0..*)						
Omit the serial numbers						
List of control units (0..*)						
Omit the control units						

Table 54: IF\_ODC\_GetVehicleDataWithoutLocale output

### 3.3.5.4 Exceptions

None.

## 3.4 Internal View - Offered Interfaces

This chapter contains interfaces which are offered for other components within the SOE system (so called internal interfaces).

---

### **3.4.1 IIF\_GetModelSeries**

Internally calls AF\_GetModelSeries (→ see chapter 3.5.1) to retrieve all existing ModelSeries entities.

### **3.4.2 IIF\_GetSalesTypes**

Internally calls AF\_GetSalesTypes (→ see chapter 3.5.2) to retrieve all existing SalesType entities.

### **3.4.3 IIF\_GetBodyTypes**

Internally calls AF\_GetBodyTypes (→ see chapter 3.5.3) to retrieve all existing BodyType entities.

### **3.4.4 IIF\_GetEquipment**

Internally calls AF\_GetEquipment (→ see chapter 3.5.3) to retrieve all existing Equipment entities.

### **3.4.5 IIF\_UpdateModelSeries**

Internally calls AF\_UpdateModelSeries (→ see chapter 3.5.5) to update all ModelSeries entities.

### **3.4.6 IIF\_UpdateSalesTypes**

Internally calls AF\_UpdateSalesTypes (→ see chapter 3.5.6) to update all SalesType entities.

### **3.4.7 IIF\_UpdateBodyTypes**

Internally calls AF\_UpdateBodyTypes (→ see chapter 3.5.7) to update all BodyType entities.

### **3.4.8 IIF\_UpdateEquipment**

Internally calls AF\_UpdateEquipment (→ see chapter 3.5.8) to update all Equipment entities.

### **3.4.9 IIF\_FetchVehicleData**

Retrieves the vehicle data for a given FIN/VIN. This includes the descriptions in the given language. Internally calls AF\_FetchVehicleData (→ section 3.5.19) in order to retrieve a vehicle's configuration.

### **3.4.10 IIF\_FetchVehicleDataWithoutLocale**

Retrieves the vehicle data for a given FIN/VIN.

Internally calls AF\_FetchVehicleDataWithoutLocale (→ section 3.5.20) in order to retrieve a vehicle's configuration.

### **3.4.11 IIF\_UpdateFirstRegistrationDate**

Updates the first registration date of the vehicle data with the given FIN/VIN.

---

Internally calls AF\_UpdateFirstRegistrationDate (→ section 3.5.21) in order to update the first registration date.

### **3.4.12 IIF\_ResolveModelSeriesForSalesType**

Returns the model series of given sales type.

Internally calls AF\_ResolveModelSeriesForSalesType (→ section 3.5.22).

### **3.4.13 IIF\_FilterAvailableModelSeries**

This internal interface offers a service that filters the given list of model series and returns the model series available for the given market and not out of production.

This interface is implemented by AF\_FilterAvailableModelSeries (→ section 3.5.24).

### **3.4.14 IIF\_GetSalesTypesForCondition**

This internal interface offers a service that determines the sales types of a given model series that meet the given sales type condition comprising a baumuster wildcard and an optional list of NST codes.

This interface is implemented by AF\_GetSalesTypesForCondition (→ section 3.5.25).

### **3.4.15 IIF\_GetSalesTypesForModelSeriesId**

This internal interface offers the service to determine and return all sales types of a model series. The model series is specified by its ID, which is given as input.

This internal interface is implemented by AF\_GetSalesTypesForModelSeriesId (→ section 3.5.26).

## **3.5 Implementation**

### **3.5.1 AF\_GetModelSeries**

#### **3.5.1.1 General Description**

This application function provides all existing model series entities.

#### **3.5.1.2 Sequence Description**

Determine all instances of entity <ModelSeries> and information from related entities that are available in SOE and return the data.

#### **3.5.1.3 Input**

None.

#### **3.5.1.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;ModelSeries&gt;: List of all ModelSeries entities</b>		
modelSeriesID	ModelSeries.modelSeriesID	The ID of a Model Series.
ProductGroup	ModelSeries.ProductGroup	The product group of the model series.
maintainedInSoe	ModelSeries.maintainedInSoe	This flag is set to true if the ModelSeries is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.

Parameter Name	Type / Length / BOM	Description
outOfProduction	ModelSeries.outOfProduction	This flag is set to true if the sales type is out of production. The flag is false by default.
- Inner List of <MPC>		
gemsOutledID	MPC.gemsOutledID	The GemsOutledID of a MPC.

Table 55: AF\_GetModelSeries output

### 3.5.1.5 Exceptions

None.

## 3.5.2 AF\_GetSalesTypes

### 3.5.2.1 General Description

This application function provides all existing sales type entities.

### 3.5.2.2 Sequence Description

Determine all instances of entity <SalesType> and information from related entities that are available in SOE and return the data.

### 3.5.2.3 Input

None.

### 3.5.2.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;SalesType&gt;: List of all SalesType entities</b>		
Baumuster	SalesType.Baumuster	The baumuster of a sales type.
NST	SalesType.NST	The NST of a sales type.
SR1	SalesType.SR1	The SR1 of the sales type.
ModelNsrExtension	SalesType.ModelNsrExtension	The ModelNsrExtension of the sales type.
maintainedInSoe	SalesType.maintainedInSoe	This flag is set to true if the SalesType is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
- Inner List of <SalesTypeDescription>		
Locale	SalesTypeDescription.locale	The locale for which a SalesTypeDescription is valid.
description	SalesTypeDescription.description	The (localized) description of a sales type.
maintainedInSoe	SalesTypeDescription.maintainedInSoe	This flag is set to true if the SalesTypeDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
- Inner List of <MPC>		
gemsOutletID	MPC.gemsOutletID	The GemsOutletID of a MPC.
bodyTypeID	BodyType.bodyTypeID	The ID of a BodyType.
modelSeriesID	ModelSeries.modelSeriesID	The ID of a ModelSeries.

Table 56: AF\_GetSalesTypes output

### 3.5.2.5 Exceptions

None.

---

### **3.5.3 AF\_GetBodyTypes**

#### **3.5.3.1 General Description**

This application function provides all existing body type entities.

#### **3.5.3.2 Sequence Description**

Determine all instances of entity <BodyType> and information from related entities that are available in SOE and return the data.

#### **3.5.3.3 Input**

None.

#### **3.5.3.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;BodyType&gt;: List of all BodyType entities</b>		
bodyTypeID	BodyType.BodyTypeID	The ID of a BodyType.
ProductGroup	BodyType.ProductGroup	The product group of the body type.
maintainedInSOE	BodyType.maintainedInSoe	This flag is set to true if the BodyType is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
- Inner List of <BodyTypeDescription>		
Locale	BodyTypeDescription.locale	The locale for which a BodyTypeDescription is valid.
description	BodyTypeDescription.description	The (localized) description of a body type.
maintainedInSoe	BodyTypeDescription.maintainedInSoe	This flag is set to true if the BodyTypeDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.

Table 57: AF\_GetBodyTypes output

#### **3.5.3.5 Exceptions**

None.

### **3.5.4 AF\_GetEquipment**

#### **3.5.4.1 General Description**

This application function provides all existing equipment entities.

#### **3.5.4.2 Sequence Description**

Determine all instances of entity <Equipment> with <Equipment.relevantForServiceAssignmentRule> = TRUE and information from related entities that are available in SOE and return the data.

#### **3.5.4.3 Input**

None.

#### **3.5.4.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;Equipment&gt;: List of all Equipment entities</b>		
Code	Equipment.Code	The Code of an Equipment.
ProductGroup	Equipment.ProductGroup	The product group of the equipment code.
maintainedInSoe	Equipment.maintainedInSoe	This flag is set to true if the Equipment is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.

Parameter Name	Type / Length / BOM	Description
		ble in AMDS and is exclusively maintained in SOE. The flag is false by default.
relevantForServiceAssignmentRule	Equipment.relevantForServiceAssignmentRule	The flag is set to true if the equipment code is relevant for maintaining the service assignment rules. Only equipment codes with this flag set to true are available on DLG_ServiceAssignmentRuleDetails. The flag is false by default.
- Inner List of <EquipmentDescription>		
Locale	EquipmentDescription.locale	The locale for which an EquipmentDescription is valid.
description	EquipmentDescription.description	The (localized) description of an Equipment.
maintainedInSoe	EquipmentDescription.maintainedInSoe	This flag is set to true if the EquipmentDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.

Table 58: AF\_GetEquipment output

### 3.5.4.5 Exceptions

None.

## 3.5.5 AF\_UpdateModelSeries

### 3.5.5.1 General Description

This application function updates the model series entities.

### 3.5.5.2 Sequence Description

Compare the <ModelSeries> given as input parameter with the existing <ModelSeries> in SOE retrieved by calling AF\_GetModelSeries.

#### If the updateMode = ADD

Add all instances of <ModelSeries> which are given as input parameter and not parts of the existing <ModelSeries>.

#### If the updateMode = UPDATE

Update all instances of <ModelSeries> that are available in SOE which are not completely equal to the <ModelSeries> given as input parameter.

#### If the updateMode = DELETE

Delete all instances of <ModelSeries> that are available in SOE which are not parts of the <ModelSeries> given as input parameter.

### 3.5.5.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the model series.
<b>List&lt;ModelSeries&gt;: List of ModelSeries entities</b>		
modelSeriesID	ModelSeries.modelSeriesID	The ID of a Model Series.
ProductGroup	ModelSeries.ProductGroup	The product group of the model series.
maintainedInSoe	ModelSeries.maintainedInSoe	This flag is set to true if the ModelSeries is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
outOfProduction	ModelSeries.outOfProduction	This flag is set to true if the sales type is out of pro-

Parameter Name	Type / Length / BOM	Description
		duction. The flag is false by default.
- Inner List of <MPC>		
gemsOutledID	MPC.gemsOutledID	The GemsOutledID of a MPC.

Table 59: AF\_UpdateModelSeries input

### 3.5.5.4 Output

None.

### 3.5.5.5 Exceptions

None.

## 3.5.6 AF\_UpdateSalesTypes

### 3.5.6.1 General Description

This application function updates the SalesType entities.

### 3.5.6.2 Sequence Description

Compare the <SalesType> given as input parameter with the existing <SalesType> in SOE retrieved by calling AF\_GetSalesTypes.

#### If the updateMode = ADD

Add all instances of <SalesType> which are given as input parameter and not parts of the existing <SalesType>.

#### If the updateMode = UPDATE

Update all instances of <SalesType> that are available in SOE which are not completely equal to the <SalesType> given as input parameter.

#### If the updateMode = DELETE

Delete all instances of <SalesType> that are available in SOE which are not parts of the <SalesType> given as input parameter.

### 3.5.6.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the sales type entities.
<b>List&lt;SalesType&gt;: List of all SalesType entities</b>		
Baumuster	SalesType.Baumuster	The baumuster of a sales type.
NST	SalesType.NST	The NST of a sales type.
SR1	SalesType.SR1	The SR1 of the sales type.
ModelNsrExtension	SalesType.ModelNsrExtension	The ModelNsrExtension of the sales type.
maintainedInSoe	SalesType.maintainedInSoe	This flag is set to true if the SalesType is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
- Inner List of <SalesTypeDescription>		
Locale	SalesTypeDescription.locale	The locale for which a SalesTypeDescription is valid.
description	SalesTypeDescription.description	The (localized) description of a sales type.
maintainedInSoe	SalesTypeDescription.maintainedInSoe	This flag is set to true if the SalesTypeDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the sales type entities.
- Inner List of <MPC>		
gemsOutletID	MPC.GemsOutletId	The GemsOutletId of a MPC.
bodyTypeID	BodyType.bodyTypeID	The ID of a BodyType.
modelSeriesID	ModelSeries.modelSeriesID	The ID of a ModelSeries.

Table 60: AF\_UpdateSalesTypes input

### 3.5.6.4 Output

None.

### 3.5.6.5 Exceptions

None.

## 3.5.7 AF\_UpdateBodyTypes

### 3.5.7.1 General Description

This application function updates the BodyType entities.

### 3.5.7.2 Sequence Description

Compare the <BodyType> given as input parameter with the existing <BodyType> in SOE retrieved by calling AF\_GetBodyTypes.

#### If the updateMode = ADD

Add all instances of <BodyType> which are given as input parameter and not parts of the existing <BodyType>.

#### If the updateMode = UPDATE

Update all instances of <BodyType> that are available in SOE which are not completely equal to the <BodyType> given as input parameter.

#### If the updateMode = DELETE

Delete all instances of <BodyType> that are available in SOE which are not parts of the <BodyType> given as input parameter.

### 3.5.7.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the body type entities.
<b>List&lt;BodyType&gt;: List of all BodyType entities</b>		
bodyTypeID	BodyType.BodyTypeID	The ID of a BodyType.
ProductGroup	BodyType.ProductGroup	The product group of the body type.
maintainedInSOE	BodyType.maintainedInSoe	This flag is set to true if the BodyType is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
- Inner List of <BodyTypeDescription>		
locale	BodyTypeDescription.locale	The locale for which a BodyTypeDescription is valid.
description	BodyTypeDescription.description	The (localized) description of a body type.

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the body type entities.
maintainedInSoe	BodyTypeDescription.maintainedInSoe	This flag is set to true if the BodyTypeDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.

Table 61: AF\_UpdateBodyTypes input

### 3.5.7.4 Output

None.

### 3.5.7.5 Exceptions

None.

## 3.5.8 AF\_UpdateEquipment

### 3.5.8.1 General Description

This application function updates the Equipment entities.

### 3.5.8.2 Sequence Description

Compare the <Equipment> given as input parameter with the existing <Equipment> in SOE retrieved by calling AF\_GetEquipment.

#### If the updateMode = ADD

Add all instances of <Equipment> which are given as input parameter and not parts of the existing <Equipment>.

#### If the updateMode = UPDATE

Update all instances of <Equipment> that are available in SOE which are not completely equal to the <Equipment> given as input parameter.

#### If the updateMode = DELETE

Delete all instances of <Equipment> that are available in SOE which are not parts of the <Equipment> given as input parameter.

### 3.5.8.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the equipment entities.
<b>List&lt;Equipment&gt;: List of all Equipment entities</b>		
code	Equipment.Code	The Code of an Equipment.
ProductGroup	Equipment.ProductGroup	The product group of the equipment code.
maintainedInSoe	Equipment.maintainedInSoe	This flag is set to true if the Equipment is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
relevantForServiceAssignmentRule	Equipment.relevantForServiceAssignmentRule	The flag is set to true if the equipment code is relevant for maintaining the service assignment rules. Only equipment codes with this flag set to true are available on DLG_ServiceAssignmentRuleDetails. The flag is false by default.
- Inner List of <EquipmentDescription>		
locale	EquipmentDescription.locale	The locale for which an EquipmentDescription is

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the equipment entities.
		valid.
description	EquipmentDescription.description	The (localized) description of an Equipment.
maintainedInSoe	EquipmentDescription.maintainedInSoe	This flag is set to true if the EquipmentDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.

Table 62: AF\_UpdateEquipment input

### 3.5.8.4 Output

None.

### 3.5.8.5 Exceptions

None.

## 3.5.9 AF\_SaveModelSeries

This application function

- saves the current changes of an existing model series or
- conducts a duplicate check and creates a model series inside a change session.

### 3.5.9.1 Sequence Description

Verify if the model series is already inside a change session

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter 17.4.1). If the error SESSION\_001 is thrown execution stops here.
- Verify if the model series is *already inside a change session of another user*: if yes, throw error VEHPROSESSION\_001. Execution stops here.

If the entity ModelSeries should be updated:

Assign each given MPC entity to the current ModelSeries entity.

If the ModelSeries entity is assigned to further MPCs, delete these assignments. Additionally update the model series' attribute outOfProduction if required.

If the entity ModelSeries should be newly created:

#### Step 1: Duplicate check

Try to retrieve the model series from the database which match the given model series for the attribute ModelSeries.ModelSeriesID.

If such a model series exists (=either has a productive form or it is in the change session of the requesting user), abort with the error message VEHPRO\_013 with ModelSeries.ModelSeriesID for placeholder <1>.

#### Step 2: Create model series

---

Create the given model series and set the attribute ModelSeries.MaintainedInSOE to “true”.

Additionally create its relation to each attached MPC.

Set the model series’ attribute outOfProduction as given and bring the model series in the change session of the user (see general algorithm ApplyChangeSessionOnElement).

### 3.5.9.2 Input

Name	Type / Length / BOM	Description
modelSeries	ModelSeries	The model series to be created or changed.
MarketList	List of MPCs	The markets the model series is valid for.
outOfProduction	Boolean	The information if the model series is out of production

Table 63: Application function input

### 3.5.9.3 Output

None.

### 3.5.9.4 Exceptions

Error message VEHPRO\_013 in case a duplicate was detected.

If the model series is already being edited inside the change session of another user, the error VEHPROSESSION\_001 is thrown.

If the user doesn’t have an opened change session, then the error SESSION\_001 is thrown.

## 3.5.10 AF\_DeleteModelSeries

This AF deletes the given model series in a change session, if they are not used anymore by any service assignment rule or by any sales type and if they have been created in SOE (model series not imported by AMDS).

### 3.5.10.1 Sequence Description

#### Check change sessions for given objects (model series)

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter 17.4.1). If the error SESSION\_001 is thrown execution stops here.
- Verify, that none of the objects is inside a change session of another user. If it is, throw error VEHPROSESSION\_001. Execution stops here.

#### Validate objects and references

- If at least one model series was not created in SOE (ModelSeries.maintainedInSOE = “false”), abort the transaction for all given model series with error message VEHPRO\_015. Use < ModelSeries.modelSeriesID> (separated by commas) of these model series that were not created in SOE as placeholder <1>.

- If at least one model series is referenced by one or more ServiceAssignmentRules abort the transaction for all given model series with the error message VEHPRO\_007. Use < ModelSeries.modelSeriesID> as placeholder 1. As placeholder 2 use a list of all ServiceAssignmentRule.ServiceAssignmentruleId which reference to the ModelSeries.
- If at least one model series is referenced by one or more sales types, abort the transaction for all given model series with the error message VEHPRO\_022. Use < ModelSeries.modelSeriesID> as placeholder 1. As placeholder 2 use a list of the keys of all sales types that reference the ModelSeries.

#### **Mark objects as deleted**

- Bring the ModelSeries in the change session of the user (see general algorithm ApplyChangeSessionOnElement).
- Mark the ModelSeries as deleted (ChangeOperation.DELETE).

#### **3.5.10.2 Input**

Name	Type / Length / BOM	Description
ModelSeries	List of ModelSeries	The model series to be deleted.

Table 64: AF\_DeleteModelSeries Input

#### **3.5.10.3 Output**

None.

#### **3.5.10.4 Exceptions**

Error message VEHPRO\_007, in case one of the model series to be deleted is referenced by one or more service assignment rules.

Error message VEHPRO\_015, in case one of the model series to be deleted was not created in SOE.

Error message VEHPRO\_022, in case one of the model series to be deleted is referenced by one or more sales types.

If the model series is already being edited inside the change session of another user, the error VEHPROSESSION\_001 is thrown.

If the user does not have an opened change session, then the error SESSION\_001 is thrown.

### **3.5.11 AF\_SaveSalesType**

This application function saves a new sales type or saves updates to an existing body type inside a change session.

#### **3.5.11.1 Sequence Description**

##### **Check change sessions for given object (sales type)**

Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown, executions stops here.

---

Verify, that that sales type is not already inside a change session of another user. If it is, throw error VEHPROSESSION\_001. Execution stops here.

**If the sales type is to be created:**

Step 1: Check for duplicates

Check, if a sales type exists that has the same key as the sales type to be saved.

If such a sales type exists (=either has a productive form or is in the change session of the requesting user), abort with the error message VEHPRO\_018 with the key of this sales type for placeholder <1>.

Step 2: Create sales type

Create the given sales type and set the attribute SalesType.MaintainedInSOE to “true”.

**If the sales type is to be created or updated:**

Create or update the assignments of the sales type to the MPCs.

Create or update the assignments of the sales type to the model series.

Create or update the assignments of the sales type to the body type.

Bring the sales type into the change session of the user (see general algorithm **ApplyChangeSessionOnElement**).

### 3.5.11.2 Input

Name	Type / Length / BOM	Description
salesType	SalesType	The sales type to be created or changed.
MarketList	List of MPCs	The markets the sales type is valid for.
modelSeries	ModelSeries	The model series the sales type is to be assigned to.
bodyType	BodyType	The body type the sales type is to be assigned to.

Table 65: AF\_SaveSalesType input

### 3.5.11.3 Output

None.

### 3.5.11.4 Exceptions

Error message VEHPRO\_018 in case a duplicate was detected.

If the sales type is already being edited inside the change session of another user, the error VEHPROSESSION\_001 is thrown.

If the user does not have an opened change session, then the error SESSION\_001 is thrown.

## 3.5.12 AF\_DeleteSalesTypes

This AF deletes the given sales types, if they are not used by any service assignment rules.

### 3.5.12.1 Sequence Description

#### Check change sessions for given objects (sales types)

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown, execution stops here.
- Verify, that none of the objects is inside a change session of another user. If it is, throw error VEHPROSESSION\_001. Execution stops here.

### **Validate objects and references**

- If one or more sales types were not created in SOE (maintainedInSOE="false"), abort the transaction for all sales types with error message VEHPRO\_019. Use the key of each sales type not created in SOE as placeholder <1>.
- Check, if one or more sales types are referenced by one or more ServiceAssignmentRules in terms of SalesTypeConditions. Here, a sales type is referenced by a ServiceAssignmentRule, if the rule contains a SalesTypeCondition and useConditionBaumusterWildcards is set, and if the baumuster of the sales type matches the baumuster wildcard of the rule, and if useConditionNst is set and the NST of the sales type matches one of the NSTs of the rule. If in this sense sales types are referenced by rules, then abort the transaction for all sales types with the error message VEHPRO\_007. Use their keys as placeholder 1. As placeholder 2 use a list of all ServiceAssignmentRules.ServiceAssignmentRuleId which reference the sales type.

### **Mark objects as deleted**

- Bring the sales types into the change session of the user (see general algorithm **ApplyChangeSessionOnElement**).
- Mark the sales types as deleted (ChangeOperation.DELETE).

#### **3.5.12.2 Input**

Name	Type / Length / BOM	Description
SalesTypes	List of SalesTypes	The sales types to be deleted.

Table 66: AF\_DeleteSalesTypes Input

#### **3.5.12.3 Output**

None.

#### **3.5.12.4 Exceptions**

Error message VEHPRO\_007, in case one of the sales types to be deleted is referenced by other entities.

Error message VEHPRO\_019, in case one of the sales types to be deleted was not created in SOE.

#### **3.5.13 AF\_SaveBodyType**

This application function saves a new body type or saves updates to an existing body type inside a change session.

### **3.5.13.1 Sequence Description**

#### **Check change sessions for given object (body type)**

Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown, execution stops here.

Verify, that that sales type is not already inside a change session of another user. If it is, throw error VEHPROSESSION\_001. Execution stops here.

#### **If the body type is to be created:**

##### Step 1: Check for duplicates

Check, if a body type exists, that has the same value key as the body type to be saved.

If such a body type exists (=either has a productive form or is in the change session of the requesting user), abort with the error message VEHPRO\_020 with BodyType.ID for placeholder <1>.

##### Step 2: Create body type

Create the given body type and set the attribute BodyType.MaintainedInSOE to “true”.

Bring the body type into the change session of the user (see general algorithm **ApplyChangeSessionOnElement**).

### **3.5.13.2 Input**

Name	Type / Length / BOM	Description
bodyType	BodyType	The body type to be saved.

Table 67: Application function input

### **3.5.13.3 Output**

None.

### **3.5.13.4 Exceptions**

Error message VEHPRO\_020 in case a duplicate was detected.

If the object (body type) is already being edited by another user inside another change session, the error VEHPROSESSION\_001 is thrown.

If the user does not have an opened change session, then the error SESSION\_001 is thrown.

## **3.5.14 AF\_DeleteBodyTypes**

This AF deletes the given list of body types in a change session, if they are not used by any service assignment rules, and if they were created in SOE (and not imported from AMDS).

### **3.5.14.1 Sequence Description**

#### **Check change sessions for given objects (body types)**

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown, executions stops here.
- Verify, that none of the objects is inside a change session of another user. If it is, throw error VEHPROSESSION\_001. Execution stops here.

### **Validate objects and references**

- If one or more body types were not created in SOE (maintainedInSOE="false"), abort the transaction for all body types with error message VEHPRO\_021. Use <BodyType.ID> of the body types not created in SOE as placeholders <1>, respectively.
- If at least one body type is referenced by one or more sales types, abort the transaction for all given model series with the error message VEHPRO\_023. Use <ModelSeries.modelSeriesID> as placeholder 1. As placeholder 2 use a list of the keys of all sales types that reference the ModelSeries.

### **Mark objects as deleted**

- Bring the body types into the change session of the user (see general algorithm **ApplyChangeSessionOnElement**).
- Mark the body types as deleted (ChangeOperation.DELETE).

#### **3.5.14.2 Input**

Name	Type / Length / BOM	Description
bodyTypes	List of BodyType	The body types to be deleted.

Table 68: AF\_DeleteBodyTypes Input

#### **3.5.14.3 Output**

None.

#### **3.5.14.4 Exceptions**

Error message VEHPRO\_021, in case one of the body types to be deleted was not created in SOE.

If one of the body types is already being edited inside the change session of another user, the error VEHPROSESSION\_001 is thrown.

If the user does not have an opened change session, then the error SESSION\_001 is thrown.

Error message VEHPRO\_023, in case one of the body types to be deleted is referenced by one or more sales types.

#### **3.5.15 AF\_SaveEquipment**

This application function

- saves the current changes of an existing equipment code or
- conducts a duplicate check and creates an equipment code

---

inside a change session.

### 3.5.15.1 Sequence Description

Verify if the equipment is already inside a change session

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter **17.4.1**). If the error SESSION\_001 is thrown executions stops here.
- Verify if the equipment is *already inside a change session of another user*: if yes, throw error VEHPROSESSION\_001. Execution stops here.

If the entity Equipment should be updated:

#### Step 1: Update EquipmentDescriptions

For each given EquipmentDescription check if the entity already exists.

- If no such EquipmentDescription exists, create the given EquipmentDescription, set its attribute EquipmentDescription.maintainedInSOE="true" and assign it to the current Equipment entity.
- If such an EquipmentDescription entity exists and has changed, save the current changes of the existing entity.  
If there are no changes do nothing.

Bring the EquipmentDescription entity in the change session of the user (see general algorithm **ApplyChangeSessionOnElement**).

#### Step 2: Update relevance for specific check(s)

Update the Equipment entities' attribute relevantForServiceAssignmentRule if required.

If the entity Equipment should be newly created:

#### Step 1: Duplicate checks

Try to retrieve equipment codes from the database which match the given equipment code for the attributes Equipment.Code, Equipment.ProductGroup.

If such a equipment code exists (=either has a productive form or it is in the change session of the requesting user), abort with the error message VEHPRO\_012. Fill the placeholder as follows:

1. Equipment.Code
2. Equipment.ProductGroup
3. Equipment.Description

#### Step 2: Create equipment code

Create the given equipment code entity and set the attribute Equipment.maintainedInSOE to "true".

Create each given EquipmentDescription entity, set the attribute EquipmentDescription.maintainedInSOE="true" and assign them to the current equipment code.

Set the Equipment entities' attribute relevantForServiceAssignmentRule as given and bring the EquipmentDescription entity in the change session of the user ( see general algorithm **ApplyChangeSessionOnElement**).

### 3.5.15.2 Input

Name	Type / Length / BOM	Description
equipment	Equipment	The equipment code to be updated or created.
equipmentDescriptionList	List of EquipmentDescriptions	A list of descriptions given for the equipment code.
relevantForServiceAssignmentRules	Boolean	States if the equipment code is relevant for maintaining any service assignment rule.

Table 69: Application function input

### 3.5.15.3 Output

None.

### 3.5.15.4 Exceptions

Error message VEHPRO\_012 in case a duplicate was detected.

If the equipment is already being edited inside the change session of another user, the error VEHPROSESSION\_001 is thrown.

If the user doesn't have an opened change session, then the error SESSION\_001 is thrown.

## 3.5.16 AF\_DeleteEquipmentCodes

This AF deletes the given equipment codes if they are not used anymore by any service assignment rule and if they have not been created in SOE (equipment code not imported by AMDS) inside a change session.

### 3.5.16.1 Sequence Description

#### Verify if the equipment is already inside a change session

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter 17.4.1). If the error SESSION\_001 is thrown execution stops here.
- Verify if the equipment is already inside a change session of another user: if yes, throw error VEHPROSESSION\_001. Execution stops here.

#### Delete the given equipment

- If at least one equipment code has not been created in SOE (Equipment.maintainedInSOE ="false") abort the transaction for all given equipment codes with error message VEHPRO\_014. Use Equipment.Code as parameter <1>, Equipment.ProductGroup as parameter <2>, and Equipment.Description in the user's language as parameter <3>.
- If at least one equipment is still being referenced by one or more ServiceAssignmentRules, abort the transaction for all given equipment codes with the error message VEHPRO\_028. Use Equipment.Description in the user's language as parameter <1>, Equipment.Code as parameter <2>, Equipment.ProductGroup as parameter <3>, and as placeholder 4 use a list of all

---

ServiceAssignmentRule.ServiceAssignmentruleId which reference to the Equipment.

#### Mark objects as deleted

- Bring the EquipmentCodes in the change session of the user (see general algorithm ApplyChangeSessionOnElement).
- Mark them as deleted (ChangeOperation.DELETE).

#### 3.5.16.2 Input

Name	Type / Length / BOM	Description
Equipment	List of Equipment	The equipment to be deleted.

Table 70: AF\_DeleteEquipmentCodes Input

#### 3.5.16.3 Output

None.

#### 3.5.16.4 Exceptions

Error message VEHPRO\_007 in case one of the equipments to delete is still referenced by other entities.

Error message VEHPRO\_014 in case one of the model series to delete has not been created in SOE.

If the equipment is already being edited inside the change session of another user, the error VEHPROSESSION\_001 is thrown.

If the user doesn't have an opened change session, then the error SESSION\_001 is thrown.

### 3.5.17 AF\_ImportSalesTypesAndGroupsBatch

#### 3.5.17.1 General Description

This application function imports all sales types and related sales types groups - model series and body types - for Mercedes me connect relevant markets and product groups, as provided by source system AMDS:

- sales types, including the information which markets they are valid for ("assignment to markets"),
- multi-lingual descriptions for sales types,
- model series, and their assignments to markets,
- assignments of sales types to model series,
- body types,
- multi-lingual descriptions for body types,
- and the assignments of sales types to body types.

Generally, master data that was created manually in SOE (maintainedInSOE == true), and is (newly) provided by AMDS at this point, gets overwritten by this batch.

Whereas master data that was created manually in SOE, but is not provided by AMDS at this point, is not deleted by this batch.

To ensure data consistency, this batch technically is divided into transactions, for the given markets. The batch, however, does not run inside a change session.

---

When this batch finds pieces of master data that are held inside change sessions of SOE users, the batch will remove such master data from change sessions and overwrite the related data, if required.

### 3.5.17.2 Sequence Description

#### **Step 1: Retrieve markets from application configuration**

Retrieve the list of market numbers (VBETs - German: Vertriebseinheiten) used by AMDS from the application configuration (PROP\_MARKETS).

The batch executes steps 2 to 5 for each combination of VBET and Mercedes connect me relevant product group.

Determine the GSSN/GEMS outlet-outlet ID of the corresponding MPC for the given market number by calling IIF\_GetMPCByCountry.

**Background:** In SOE, outlet-outlet IDs are used to represent MPCs, and the several vehicle master data is imported and assigned to MPCs by this batch.

#### **Step 2a: Retrieve sales type groups (model series, body types) and assigned sales types**

The batch calls the external interface IF\_AMDS\_GetSalesTypeGroups to retrieve all sales type groups and all assigned sales types available for the given market and product group.

**Background:** Sales type groups exist for a number of classifications ("sales type classes") such as model series, body types and other classifications. For SOE, assignments of sales types to model series and to body types are relevant.

#### **Step 2b: Retrieve sales type descriptions**

The batch calls the external interface IF\_AMDS\_GetSalesTypes to retrieve all sales type descriptions available for the given market and in several languages.

#### **Step 3: Filter SOE relevant model series, body types and assigned sales types**

Only sales type groups that satisfy the following conditions are considered:

If the product group is 'P' (passenger cars):

- SOE relevant classes: Retrieve these classes from application configuration PROP\_AMDS\_SALESTYPEGROUPCLASSES. Currently, this list contains the sales type group classes "BR" and "BODY" providing model series and body types, respectively. Only sales types whose baumuster has 7 digits are considered. All other sales types are ignored.
- All model series listed in PROP\_AMDS\_MBCONNECT\_INCAPABLE\_MODELSERIES are ignored.

If the product group is "T" (transporters):

- SOE relevant classes: Retrieve these classes from application configuration PROP\_AMDS\_SALESTYPEGROUPCLASSES. Currently, this list contains the sales type group classes "BR" and "BODY" providing model series and body types, respectively.
- There is no restriction with respect to the size of the baumuster.
- All model series listed in PROP\_AMDS\_MBCONNECT\_INCAPABLE\_MODELSERIES are ignored.

---

### **Step 4a: Save model series**

For all returned model series (determined in Step 3) check if there already exists a ModelSeries entity in SOE (by comparing ModelSeries.ModelSeriesID).

#### Case 1: No such model series exists

- Create a new ModelSeries entity in SOE database
- Assign it to the MPC with the GSSN/GEMS outlet-outlet ID determined before.
- Set ModelSeries.MaintainedInSoe="false".

#### Case 2: Returned model series already exists in SOE

- Assign the ModelSeries entity to the MPC with the GSSN/GEMS outlet-outlet ID.
- Set ModelSeries.MaintainedInSoe="false".

Note: As the model series ID is unique across all product groups the attribute ModelSeries.ProductGroup is not required to check for duplicate model series.

### **Step 4b: Save body types**

Process all retrieved and filtered body types (Step 3):

Check, if the body type exists in SOE.

If the body type does not exist, create a new one, else take the existing one.

Set MaintainedInSoe to "false".

Save the descriptions (BodyTypeDescription) for the given body type and languages.

### **Step 5: Save sales types, descriptions and assignments to model series and body types**

For the SOE relevant ModelSeries and BodyTypes (Step 3) process all retrieved and filtered sales types:

Check, if the sales type already exists in SOE.

If the sales type does not exist, create a new one, else take the existing one.

Assign the SalesType to the MPC using the GSSN/GEMS outlet-outlet ID.

Assign the SalesType to the ModelSeries and to the BodyType.

Set MaintainedInSoe = "false".

Save the descriptions (SalesTypeDescription) for the given sales type, languages and market.

### **Step 6: Delete model series, body types and sales types that are not provided by AMDS anymore**

After processing data for each of the markets separately (steps 2 to 5), check, if there is data that was provided by AMDS before (MaintainedInSOE == false), but is not provid-

---

ed by AMDS anymore at this point for any of the markets. Such data was probably deleted in AMDS and needs to be deleted in SOE, here:

#### Model series

If there are ModelSeries entities in SOE that are not provided by AMDS anymore:

Delete these ModelSeries entities and all assigned SalesType entities, if

- the attribute ModelSeries.MaintainedInSOE = "false" AND
- this model series is not referenced by any service assignment rule. If this is the case, the application function logs the following warning message with  
<ModelSeries.ModelSeriesID> as placeholder <1> and  
<ServiceAssignmentRule.ID> as placeholder <2> and continues:

*"The model series <1> deleted in AMDS cannot be deleted in SOE because there it is still being referenced by one or more service assignment rules. The service assignment rules with the following Ids(s) have been identified as referencing to this item: <2>."*

#### Body types

Check, if there are body types in SOE that

- were not maintained in SOE (MaintainedInSOE == "false")
- and are not retrieved from AMDS at this point.

Delete these body types.

#### Sales types

Check, if there are sales types in SOE that

- were not maintained in SOE (MaintainedInSOE == "false")
- and are not retrieved from AMDS at this point
- and are not referenced by any service assignment rule.

Delete these sales types.

For sales types referenced by service assignment rules, log the following warning message *"The sales type <SalesTypeKey> not available from AMDS cannot be deleted in SOE because it is referenced by these service assignment rules: <ServiceAssignmentRule.ID>."*

### **Step 7: Trigger SOE Regions**

Trigger SOE Region instances of SOEMDM about changes to vehicle products master data that were carried out through this batch run, by triggering IIF\_TriggerReplications with MODELSERIES, SALESTYPE and BODYTYPE as input parameter <MasterDataReplicationTypeEnum>.

#### **3.5.17.3 Input**

None.

---

### **3.5.17.4 Output**

None.

### **3.5.17.5 Exceptions**

None.

## **3.5.18 AF\_ImportEquipmentCodesBatch**

### **3.5.18.1 General Description**

This application function imports all equipment codes including the available (localized) code descriptions present in the third party system AMDS.

Equipment or EquipmentDescription entities that are saved in SOE and that haven't been imported by AMDS are not deleted by this batch.

Equipment or EquipmentDescription entities that have been manually created in SOE are overwritten by corresponding entities imported by this batch.

*Note: A technical user needs to be defined to be used by this batch. This user is not visible in the UI.*

### **3.5.18.2 Sequence Description**

#### ***Step 1: Retrieve markets from Application Configuration***

Retrieve the list of market numbers (VBETs) used by AMDS from application configuration (PROP\_MARKETS).

The batch executes steps 2 and 3 for each combination of VBET and Mercedes connect me relevant product group.

#### ***Step 2: Retrieve equipment codes***

The external interface IF\_AMDS\_getCodes (see chapter 3.3.2) is called in order to import all Equipment codes including their descriptions.

Check for each returned Equipment code if there is already an Equipment entity existing in SOE.

- If no such entity exists, create a new one in SOE database in the change session of the user and set the attribute MaintainedInSOE="false".
- If the Equipment entity already exists, set its attribute MaintainedInSOE="false".

#### ***Step 3: Determine codes descriptions***

Consider for each returned Equipment code the included Equipment descriptions:

Case 1: EquipmentDescription entity already exists (The value determined by the given market number (VBET) and the returned parameter "language" corresponds to the attribute EquipmentDescription.Locale):

- Overwrite the existing entity by the imported one.
- Set its attribute MaintainedInSOE to "false".

---

Case 2: No corresponding EquipmentDescription entity exists in SOE (No entity with attribute EquipmentDescription.Locale corresponding to the value determined by the given market number (VBET) and the returned parameter “language”):

- Create a new EquipmentDescription entity
- Assign it to the currently considered Equipment entity
- The given market number (VBET) and the returned parameter “language” determine the attribute EquipmentDescription.Locale.

#### **Step 4: Delete equipment descriptions that are not imported from AMDS**

If there exist EquipmentDescription entities in SOE database that are not imported from AMDS, delete these entities only if the attribute Equipment.MaintainedInSOE="false".

#### **Step 5: Delete equipment codes that are not imported from AMDS**

If there exist Equipment entities in SOE database that are not imported from AMDS:

Delete these entities and all assigned EquipmentDescription entities if

- the attribute Equipment.MaintainedInSOE="false" AND
- this equipment code is not referenced by any service assignment rule. If this is the case, the application function logs the following warning message with the key of the equipment as parameter <1> and a list of <ServiceAssignmentRule.ID> referencing the respective equipment as placeholder <2>:

*“The equipment code <1> deleted in AMDS cannot be deleted in SOE because there it is still being referenced by one or more service assignment rules. The service assignment rules with the following Id(s) have been identified as referencing to this item: <2>.”*

#### **Step 6: Trigger SOE Regions**

Call IIF\_TriggerReplications with EQUIPMENT as input parameter  
<MasterDataReplicationTypeEnum>.

#### **3.5.18.3 Input**

None.

#### **3.5.18.4 Output**

None.

#### **3.5.18.5 Exceptions**

None.

### **3.5.19 AF\_FetchVehicleData**

#### **3.5.19.1 General Description**

Retrieves the vehicle data for a given FIN/VIN. This includes the descriptions of the vehicle, the model series, baumuster and equipment in the given language.

---

Note: Currently, this application function returns neither the SR1 context nor the ModelNsrExtension of a sales type since UVS does not provide these attributes.

### 3.5.19.2 Sequence Description

1. If no locale is given as input parameter, continue with the default locale “de\_DE”.  
If no country is given, use the country of the locale. If the locale contains no country part, proceed with the country “DE”.
2. Use the external interface IF\_ODC\_GetVehicleData in order to retrieve the given vehicle. Use the given input parameter “Locale” as parameter “Locale”, and “Country” as “Country” for the external interface. Besides, set the input parameter RequestedData to *BOTH*.
3. Pass in the given input parameter FinOrVin as parameter “finOrVin”.
4. If the answer from IF\_ODC\_GetVehicleData does not contain the desired vehicle configuration abort with the error message VEHPRO\_005 (this is the case if the returned list of vehicles’ entry for the requested vinOrVin does not contain any Vehicle Information). In case of technical errors abort with VEHPRO\_006.
5. If the corresponding VehicleCoreData entry does not exist yet, create one using the resolved data.
6. If the corresponding VehicleCoreData entry does already exist and contains a first registration date, add this date to the response data.
7. If the attribute modelSeries of the corresponding VehicleCoreData is not filled, call AF\_ResolveModelSeriesForSalesType with the SalesType.baumuster returned by IF\_ODC\_GetVehicleData and save the returned ModelSeries.modelSeriesID in VehicleCoreData.modelSeries.
8. Map the ODC response to the returned vehicle configuration as follows:
  - a. VehicleConfiguration.Baumuster = ODC.Baumuster
  - b. VehicleConfiguration.BaumusterDescription = ODC.BaumusterDescription
  - c. VehicleConfiguration.NST = ODC.NST
  - d. VehicleConfiguration.ChangeYear = First code of ODC.ListOfChangeYearCodes, if available
  - e. VehicleConfiguration.Equipment = ODC.ListOfEquipments.Code
  - f. VehicleConfiguration.EquipmentDescription = ODC.ListOfEquipments.Description
  - g. VehicleConfiguration.ConsumerCountry = ODC.ConsumerCountry
  - h. VehicleConfiguration.Paint1Code = ODC.Paint1Code
  - i. VehicleConfiguration.Paint1Description = ODC.Paint1Description
  - j. VehicleConfiguration.Paint2Code = ODC.Paint2Code
  - k. VehicleConfiguration.Paint2Description = ODC.Paint2Description
  - l. VehicleConfiguration.UpholsteryCode = ODC.UpholsteryCode
  - m. VehicleConfiguration.UpholsteryDescription = ODC.UpholsteryDescription
9. Return the vehicle configuration as return parameter “Vehicle”

### 3.5.19.3 Input

Name	Type / Length / BOM	Description
FinOrVin	String	The vehicle identifier number. May be a European FIN or the American VIN.

Locale	String	Optional. The locale the vehicle's texts shall be translated in. If not given, a potentially already cached translation or a default translation will be returned.
Country	String	Optional. The country that shall be used to translate the vehicle's texts. If not given, a potentially already cached translation or a default translation will be returned.

Table 71: AF\_FetchVehicleData Input

### 3.5.19.4 Output

Name	Type / Length / BOM	Description	Changes
VehicleConfiguration	ODC:VehicleConfiguration	The vehicle's configuration, including the related SalesType, ModelYear, Equipment, ChangeYear(s) (if available), ConsumerCountry (if available), Upholstery (if available) and Paint (if available).	Changed (attribute-description only)

Table 72: AF\_FetchVehicleData Output

### 3.5.19.5 Exceptions

- Error message VEHPRO\_005 if the given vehicle cannot be found on UVS
- Error message VEHPRO\_006 if things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors)

## 3.5.20 AF\_FetchVehicleDataWithoutLocale

### 3.5.20.1 General Description

Retrieves the vehicle data for a given FIN/VIN.

Note: Currently, this application function returns neither the SR1 context nor the ModelNsrExtension extension of a sales type since UVS does not provide these attributes.

### 3.5.20.2 Sequence Description

1. Use the external interface IF\_ODC\_GetVehicleDataWithoutLocale in order to retrieve the given vehicle.
2. Pass in the given input parameter FinOrVin as parameter "finOrVin". Besides, set the input parameter RequestedData to *BOTH*.
3. If the answer from IF\_ODC\_GetVehicleDataWithoutLocale does not contain the desired vehicle configuration abort with the error message VEHPRO\_005 (this is the case if the returned list of vehicles' entry for the requested vinOrVin does not contain any Vehicle Information). In case of technical errors abort with VEHPRO\_006.
4. If the corresponding VehicleCoreData entry does not exist yet, create one using the resolved data.
5. If the corresponding VehicleCoreData entry does already exist and contains a first registration date, add this date to the response data.

- 
6. If the attribute modelSeries of the corresponding VehicleCoreData is not filled, call AF\_ResolveModelSeriesForSalesType with the SalesType.baumuster returned by IF\_ODC\_GetVehicleDataWithoutLocale and save the returned ModelSeries.modelSeriesID in VehicleCoreData.modelSeries.
  1. Map the ODC response to the returned vehicle configuration as follows:
    - a. VehicleConfiguration.Baumuster = ODC.Baumuster
    - b. VehicleConfiguration.NST = ODC.NST
    - c. VehicleConfiguration.ChangeYear = First code of ODC.ListOfChangeYearCodes, if available
    - d. VehicleConfiguration.Equipment = ODC.ListOfEquipments.Code
    - e. VehicleConfiguration.ConsumerCountry = ODC.ConsumerCountry
    - f. VehicleConfiguration.Paint1Code = ODC.Paint1Code
    - g. VehicleConfiguration.Paint2Code = ODC.Paint2Code
    - h. VehicleConfiguration.UpholsteryCode = ODC.UpholsteryCode
  7. Return the vehicle basic configuration as return parameter “Vehicle”

### 3.5.20.3 Input

Name	Type / Length / BOM	Description
FinOrVin	String	The vehicle identifier number. May be a European FIN or the American VIN.

Table 73: AF\_FetchVehicleDataWithoutLocale Input

### 3.5.20.4 Output

Name	Type / Length / BOM	Description
VehicleConfiguration	ODC:VehicleConfiguration	The vehicle’s configuration, including the related SalesType, ModelYear, Equipment, ChangeYear(s) (if available), ConsumerCountry (if available), ConsumerCountry (if available), Upholstery (if available) and Paint (if available).

Table 74: AF\_FetchVehicleDataWithoutLocale Output

### 3.5.20.5 Exceptions

- Error message VEHPRO\_005 if the given vehicle cannot be found on UVS
- Error message VEHPRO\_006 if things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors)

## 3.5.21 AF\_UpdateFirstRegistrationDate

Updates the first registration date of the VehicleCoreData with the given FIN/VIN.

### 3.5.21.1 Sequence Description

Retrieve the VehicleCoreData with the given input parameter FinOrVin. If the VehicleCoreData.FirstRegistrationDate is already set, check if the given input parameter FirstRegistrationDate differs from the current VehicleCoreData.FirstRegistrationDate and log a warning to the application log (e.g. “The first registration date of the vehicle <FinOrVin> is currently set to <VehicleCoreData.FirstRegistrationDate>, but is about to

---

be set to <FirstRegistrationDate>"). Explanation: The first registration date is set on the first registration and remains valid even in case the vehicle is re-sold.

### 3.5.21.2 Input

Name	Type / Length / BOM	Description
FinOrVin	String	The vehicle identifier number. May be a European FIN or the American VIN.
FirstRegistrationDate	Date	The new first registration date.

Table 75: AF\_UpdateFirstRegistrationDate Input

### 3.5.21.3 Output

None.

### 3.5.21.4 Exceptions

None.

## 3.5.22 AF\_ResolveModelSeriesForSalesType

Retrieves a matching model series for a given sales type. Uses direct search and a fall-back.

### 3.5.22.1 Sequence Description

Determine the SalesType entities for which the given input parameter Baumuster matches with SalesType.Baumuster.

- A. At least one matching SalesType entity can be found:

For the first matching Sales Type entity, determine the ModelSeries entity, this SalesType entity is assigned to and return it.

- B. No matching SalesType entity can be found or the SalesType has no associated ModelSeries:

Determine the first three digits of Baumuster. Determine the ModelSeries entity for which the attribute ModelSeries.ModelSeriesID is corresponding to these three digits.

- a. The matching ModelSeries entity can be found:

- i. Return the ModelSeries. The application function logs the following warning message with <Baumuster> as placeholder <1>, <Modelseries> as placeholder <2> and continues:

- ii. *"The requested sales type <1> hasn't been found in the set of sales types received from AMDS. The sales type is successfully assigned to model series <2> anyway."*

- b. No matching ModelSeries entity can be found:

- Log the following warning in the application log "Could not resolve model series for sales type with baumuster <SalesType.baumuster>".

- Return with an empty result.

### **3.5.22.2 Input**

Name	Type / Length / BOM	Description
Baumuster	String	The baumuster part of the sales type.

Table 76: AF\_ResolveModelSeriesForSalesType Input

### **3.5.22.3 Output**

Name	Type / Length / BOM	Description
ModelSeries	ModelSeries	The model series (if available).

Table 77: AF\_ResolveModelSeriesForSalesType Output

### **3.5.22.4 Exceptions**

None.

## **3.5.23 AF\_RetrieveModelSeriesForFIN**

Returns the ModelSeries of the VehicleCoreData with the given FIN/VIN.

### **3.5.23.1 Sequence Description**

Retrieve the VehicleCoreData with the given input parameter FinOrVin.

If there is no VehicleCoreData.ModelSeries given, call IF\_ODC\_GetVehicleData with the given input parameter FinOrVin.

Call AF\_ResolveModelSeriesForSalesType with the sales type (and NST) returned by IF\_ODC\_GetVehicleData and save the returned ModelSeries.modelSeriesID in VehicleCoreData.modelSeries.

Return the data saved in VehicleCoreData.ModelSeries.

### **3.5.23.2 Input**

Name	Type / Length / BOM	Description
FinOrVin	String	The vehicle identifier number. May be a European FIN or the American VIN.

Table 78: AF\_RetrieveModelSeriesForFIN Input

### **3.5.23.3 Output**

Name	Type / Length / BOM	Description
modelSeries	VehicleCoreData.modelSeries	The model series of the given vehicle.

Table 79: AF\_RetrieveModelSeriesForFIN Output

### **3.5.23.4 Exceptions**

None.

---

### **3.5.24 AF\_FilterAvailableModelSeries**

This application function filters the given list of model series and returns the model series available for the given market and not out of production.

#### **3.5.24.1 Sequence Description**

For each model series from the given list, check, if the model series is available for the given market, and if it is not out of production (ModelSeries.outOfProduction = false).

Return a list of these model series.

#### **3.5.24.2 Input**

Name	Type / Length / BOM	Description
ModelSeriesList	List of <ModelSeries>	The list of model series to be filtered.
Market	MPC	The market to be used for filtering.

Table 80: AF\_FilterAvailableModelSeries Input

#### **3.5.24.3 Output**

Name	Type / Length / BOM	Description
ModelSeriesList	List of <ModelSeries>	The filtered list of model series. It contains those input model series that are available for the given market and not out of production.

Table 81: AF\_FilterAvailableModelSeries Output

#### **3.5.24.4 Exceptions**

None.

### **3.5.25 AF\_GetSalesTypesForCondition**

This application function determines the sales types of a given model series that meet the given sales type condition comprising a baumuster wildcard and an optional list of NST codes.

#### **3.5.25.1 Sequence Description**

Get all sales types available in the vehicle master data.

For each sales type, check, if its model series matches the one given as input. Furthermore, check if it matches the given sales type condition. The sales type condition is met, if SalesTypeCondition.useConditionBaumusterWildcards is set, and the Baumuster wildcard is matched by the baumuster of the given sales type, and, if SalesTypeCondition.useConditionNst is set, the list of NST codes contains the NST of the given sales type.

Return a list of the sales types which satisfy all of these conditions.

### 3.5.25.2 Input

Name	Type / Length / BOM	Description
ModelSeries	ModelSeries	The model series a sales type must have to be included in the output.
SalesTypeCondition	SalesTypeCondition	The sales type condition comprising a baumuster wildcard and optional list of NST codes to be matched by the sales types to be returned.

Table 82: AF\_GetSalesTypesForCondition Input

### 3.5.25.3 Output

Name	Type / Length / BOM	Description
SalesTypesList	List of <SalesType>	The sales types that meet the given sales type condition.

Table 83: AF\_GetSalesTypesForCondition Output

### 3.5.25.4 Exceptions

None.

## 3.5.26 AF\_GetSalesTypesForModelSeriesId

This application function determines and returns all sales types of a model series. The model series is specified by its ID, which is given as input.

### 3.5.26.1 Sequence Description

- Initialize an empty list salesTypesList.
- Get all sales types.
- For each sales type:
  - Check if the ID of its model series matches modelSeriesID.
  - If the IDs match, add the sales type to salesTypesList.
- Return salesTypesList.

### 3.5.26.2 Input

Name	Type / Length / BOM	Description
modelSeriesID	ModelSeries.modelSeriesId	The ID of the model series of the requested sales types'.

Table 84: AF\_GetSalesTypesForModelSeriesId Input

### 3.5.26.3 Output

Name	Type / Length / BOM	Description
salesTypesList	List<SalesType>	All sales types of the model series specified by the model series ID given as input.

Table 85: AF\_GetSalesTypesForModelSeriesId Output

### **3.5.26.4 Exceptions**

None.

## **3.5.27 AF\_GetModelYearsByProductGroup**

### **3.5.27.1 General Description**

This application function retrieves all model year codes stored in SOE and returns them.

### **3.5.27.2 Sequence Description**

### **3.5.27.3 Input**

Name	Type / Length / BOM	Description
ProductGroup	ProductGroupEnum	The product group.

Table 86: AF\_GetModelYear Input

### **3.5.27.4 Output**

Name	Type / Length / BOM	Description
List of <ModelYear>		
ModelYear	ModelYear	The model year entity.

Table 87: AF\_GetModelYear Input

### **3.5.27.5 Exceptions**

None.

## **3.6 Batches**

Batchname	Called Application Function	Description
Batch "Import Sales Types and groups"	AF_ImportSalesTypesAndGroupsBatch	<u>Execution time:</u> night time <u>Execution frequency:</u> daily This batch can also be started manually via the SOE batch admin dialogue.
Batch "Import Codes"	AF_ImportEquipmentCodesBatch	<u>Execution time:</u> night time <u>Execution frequency:</u> daily This batch can also be started manually via the SOE batch admin dialogue.

Table 88: Batches

## **3.7 Error Messages**

Message Id	Fault Title	Fault Message
VEHPRO_005	Vehicle not found	The given vehicle could not be found on the Daimler vehicle database.
VEHPRO_006	Error retrieving data from UVS	An unexpected error occurred when trying to retrieve the requested vehicle from the UVS system.
VEHPRO_007	Cannot delete item	The item <1> could not be deleted because it is still being referenced by one or more service assignment rules. Please go to the service assignment rule maintenance screen and remove the respective item from the service

		assignment rule(s) first. The service assignment rule(s) with the following ID(s) have been identified as referencing to this item: <4>.
VEHPRO_008	Mandatory field missing	The mandatory field "<1>" is not provided. Please fill this field. Hint: Mandatory fields are marked with an asterisk ("*").
VEHPRO_009	Baumuster Format	The Baumuster must be provided in the format xxx.xxx.x, e.g. "170.447.1".
VEHPRO_010	Equipment code format	The equipment code must be numeric. Please enter a numeric value.
VEHPRO_011	No country number found	No matching country number for the country <Country> can be found in the application configuration.
VEHPRO_012	Duplicate equipment code detected	The equipment code <1> - <2> (<3>) you are trying to save already exists.
VEHPRO_013	Duplicate model series detected	The model series ("<1>") you are trying to save already exists.
VEHPRO_014	Cannot delete code(s)	The equipment code(s) <1> - <2> (<3>) cannot be deleted because it is/they are not maintained in SOE.
VEHPRO_015	Cannot delete model series	The model series <1> cannot be deleted because it is/they are not maintained in SOE.
VEHPRO_016	ModelSeriesID Format	The Model series id must be provided in the format xxx, e.g. "204".
VEHPRO_017	No market selected.	Please select at least one market the model series is valid for.
VEHPRO_018	Duplicate sales type detected	The sales type "<1>" you are trying to save already exists.
VEHPRO_019	Cannot delete sales type	The sales type <1> cannot be deleted, because it is not maintained in SOE.
VEHPRO_020	Duplicate body type detected	The body type ("<1>") you are trying to save already exists.
VEHPRO_021	Cannot delete body type	The body type <1> <2> cannot be deleted, because it is not maintained in SOE.
VEHPRO_022	Cannot delete model series, because referenced by sales types	The model series <1> could not be deleted, because it is referenced by one or more sales types: <2>.
VEHPRO_023	Cannot delete body type, because referenced by sales types	The body type <1> could not be deleted, because it is referenced by one or more sales types: <2>.
VEHPRO_024		The Baumuster must match the model series in order to save the sales type.
VEHPRO_025	One or both values must be provided	One or both of the values <1>, <2> must be provided.
VEHPRO_028	Cannot delete equipment code.	The equipment code <1> (<2>) of product group <3> could not be deleted because it is still being referenced by one or more service assignment rules. Please go to the service assignment rule maintenance screen and remove the respective equipment code from the service assignment rule(s) first. The service assignment rule(s) with the following ID(s) have been identified as referencing to this item: <4>.
VEHPROSESSION_001	<VehicleProduct Element> is locked by another user	The element <VehicleProduct Element> you are trying to modify is already being edited by user <user>. Changes are not possible and will be discarded automatically.

Table 89: Messages of the component Vehicle Products



## 4 Component “Services”

### 4.1 Dialogs

#### 4.1.1 DLG\_ServiceCategoryOverview

This dialog shows a list of the service categories available in SOE. Service categories can be maintained in SOE. Starting from this dialog, service categories can be added, edited or deleted.

The screenshot shows a dialog titled "Add new Service Category". At the top right are buttons for "Delete", "Items per page" (set to 10), and "Page" (set to 1 of 3). Below is a table with columns: ID, Sort Order, Name, Opened Session, and Action. The table contains three rows:

	ID	Sort Order	Name	Opened Session	Action
<input type="checkbox"/>	101	1	Standard Services		<input type="button" value="Edit"/>
<input type="checkbox"/>	102	3	Vehicle Setup		<input type="button" value="View"/>
<input type="checkbox"/>	103	2	Vehicle Monitoring		<input type="button" value="Edit"/>

Figure 40: DLG\_ServiceCategoryOverview

##### 4.1.1.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all body types and sort them by BodyType.ID.
“Add new Service Category”	Button	Switch to dialog DLG_ServiceCategoryDetail (see chapter 0) for creation of a new service category.
“Delete”	Button	After a confirmation popup, delete the selected service categories by calling AF_DeleteServiceCategories (see chapter 3.5.14). Display error messages in the according message popup.
“Edit”	Button	Switch to dialog DLG_ServiceCategoryDetail (see chapter 0) to edit the service category. The dialog is opened in edit mode.
“View”	Button	Switch to dialog DLG_ServiceCategoryDetail (see chapter 0) to view the

---

<b>Linked label / button labeling</b>	<b>Type</b>	<b>Action description</b>
		service category. The dialog is opened in read-only mode.

Table 90: Buttons and functions (DLG\_ServiceCategoryOverview)

#### 4.1.1.2 Form fields and front-end data objects

<b>Field</b>	<b>Type</b>	<b>Details / Default</b>	<b>Business Object. Attribute</b>
Table column "ID"	Label	The ID of the service category.	ServiceCategory.serviceCategoryId
Table column "Sort Order"	Label	The sort order of the service category.	ServiceCategory.sortOrder
Table column "Name"	Label	The name of the service category (in the language of the user).	ServiceCategory.name
Table Column "Opened Sessions"	Label/Icon	Is the service category being edited in a change session.	-

Table 91: Form fields and front-end data objects (DLG\_ServiceCategoryOverview)

#### 4.1.1.3 Dialog field validation

None.

#### 4.1.1.4 Configurability (incl. settings for roles)

None.

#### 4.1.1.5 Dialog Elements States

None.

### 4.1.2 DLG\_ServiceCategoryDetail

This dialog allows to create (add) a new service category or edit or view the details of an existing service category.

Further, the translations of service category names can be maintained using this dialog.

<b>Service Category ID</b>	101
<b>Sort Order*</b>	1
Language	Translation
<b>German (Germany)</b>	Basisdienste
German (Austria)	
German (Switzerland)	
<b>English (United Kingdom)</b>	Standard Services
English (Switzerland)	
<b>French (France)</b>	Services essentiels
French (Belgium)	
French (Switzerland)	
<b>Italian (Italy)</b>	Servizi di Base
<b>Save</b>	<b>Cancel</b>

Figure 41: DLG\_ServiceCategoryDetail

#### 4.1.2.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Edit mode: Load the service category and its translations, i.e. load the maintainable languages defined in the application configuration and the available translations.  New mode: Load the maintainable languages from the application configuration and generate the table rows.
“Save”	Button	Save the service category (see AF_SaveServiceCategory) and switch to DLG_ServiceCategoryOverview (see chapter 3.1.5).
“Cancel”	Button	Discard the changes and switch to DLG_ServiceCategoryOverview (see chapter 3.1.5).

Table 92: Buttons and functions (DLG\_ServiceCategoryDetail)

#### 4.1.2.2 Form fields and front-end data objects

Linked Label	Type	Details / Default	Business Object. Attribute
Service Category ID	Label	The ID of the service category. Not editable. Generated by the system during.	ServiceCategory.serviceCategoryId
Sort Order	Textbox	The sort order of the service category, i.e. the position at which this category appears in the sequence of categories, when displayed by frontend systems.	ServiceCategory.sortOrder
Table column “Language”	Label	Locale of the maintained translation.	-
Table column “Translation”	Textbox	The localized name of the service category.	ServiceCategory.name

Table 93: Form fields and front-end data objects (DLG\_ServiceCategoryDetail)

#### 4.1.2.3 Dialog field validation

Linked Field	Validation	Error message
Sort Order	A value must be provided.	SERMAN_020. Use the screen field name as parameter 1.
Sort Order	The value for sort order is already assigned to another service category.	SERMAN_022.

Table 94: Dialog field validations (DLG\_ServiceCategoryDetail)

#### 4.1.2.4 Configurability (incl. settings for roles)

None.

#### 4.1.2.5 Dialog Elements States

None.

### 4.1.3 DLG\_ServiceAssignmentRulesOverview

#### 4.1.3.1 General Description

This dialog is the entry point for maintaining the available service assignment rules. It displays a list of all currently maintained service assignment rules and allows the creation, deletion or editing the available rules.

<div style="text-align: center; margin-bottom: 5px;"> <input type="button" value="Delete"/> <input type="button" value="Add new Service Assignment Rule"/> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>Items per page</span> <span>10</span> <span>Page 1   2   3</span> </div>																					
	Id	Services / Validity	Model Series / Sales Types	Year Code Combinations / Equipment Condition	Consumer Countries	Opened Sessions	Action														
	1	<p><b>Description</b> Model Series 205 2014 mit Audio 20+Bluetooth</p> <p><b>Services</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Service-Name</td> <td style="padding: 2px; text-align: right;">V S-ID</td> </tr> <tr> <td style="padding: 2px;">Accident Management</td> <td style="padding: 2px; text-align: right;">1 1</td> </tr> <tr> <td style="padding: 2px;">Breakdown Management</td> <td style="padding: 2px; text-align: right;">1 2</td> </tr> <tr> <td style="padding: 2px;">Maintenance &amp; Predictive ...</td> <td style="padding: 2px; text-align: right;">1 3</td> </tr> <tr> <td style="padding: 2px;">Vehicle Finder</td> <td style="padding: 2px; text-align: right;">1 5</td> </tr> <tr> <td style="padding: 2px;">Vehicle Tracker / Geofencing</td> <td style="padding: 2px; text-align: right;">1 6</td> </tr> <tr> <td style="padding: 2px;">...</td> <td style="padding: 2px; text-align: right;">1 7</td> </tr> </table>	Service-Name	V S-ID	Accident Management	1 1	Breakdown Management	1 2	Maintenance & Predictive ...	1 3	Vehicle Finder	1 5	Vehicle Tracker / Geofencing	1 6	...	1 7	<p><b>Model Series</b> 205</p> <p><b>Wildcard / NST</b> 205.0* / -</p>	<p>Model Year (+ Change Year)</p> <p>804</p> <p>805</p> <p>Equipment</p> <p>Basic Services (06U) and MBconnect Remote (05U) and Standheizung (228) and Live Traffic (B54)</p>	<p><b>Available in:</b></p> <p>Belgien (BE)</p> <p>Deutschland (DE)</p> <p>Frankreich (FR)</p> <p>Großbritannien (GE)</p> <p>Irland (IE)</p> <p>Italien (IT)</p> <p>Japan (JP)</p> <p>Luxemburg (LU)</p>		<input type="button" value="Edit"/>
Service-Name	V S-ID																				
Accident Management	1 1																				
Breakdown Management	1 2																				
Maintenance & Predictive ...	1 3																				
Vehicle Finder	1 5																				
Vehicle Tracker / Geofencing	1 6																				
...	1 7																				
	2	<p><b>Description</b> Model Series 205, 212 2014 ohne Sonderausstattung</p> <p><b>Services</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Service-Name</td> <td style="padding: 2px; text-align: right;">V S-ID</td> </tr> <tr> <td style="padding: 2px;">Accident Management</td> <td style="padding: 2px; text-align: right;">2 11</td> </tr> <tr> <td style="padding: 2px;">Breakdown Management</td> <td style="padding: 2px; text-align: right;">2 12</td> </tr> <tr> <td style="padding: 2px;">Maintenance &amp; Predictive ...</td> <td style="padding: 2px; text-align: right;">2 13</td> </tr> </table>	Service-Name	V S-ID	Accident Management	2 11	Breakdown Management	2 12	Maintenance & Predictive ...	2 13	<p><b>Model Series</b> 205</p> <p><b>Wildcard / NST</b> - / -</p>	<p>Model Year (+ Change Year)</p> <p>805 + 055</p> <p>806</p> <p>Equipment</p> <p>(Hemes (360) or Hemes LTE (362) ) and Code A (C01) and not Code B (C02)</p>	<p><b>Available in:</b></p> <p>Belgien (BE)</p> <p>Deutschland (DE)</p> <p>Frankreich (FR)</p> <p>Großbritannien (GE)</p> <p>Irland (IE)</p> <p>Italien (IT)</p> <p>Japan (JP)</p> <p>Luxemburg (LU)</p>		<input type="button" value="View"/>						
Service-Name	V S-ID																				
Accident Management	2 11																				
Breakdown Management	2 12																				
Maintenance & Predictive ...	2 13																				

Figure 42: DLG\_ServiceAssignmentRulesOverview

#### 4.1.3.2 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all available service assignment rules together with the permitted actions for each loaded service assignment rule by calling the loading algorithm described in Chapter “ <b>Loading of master data elements</b> ”. Sort them by ServiceAssignmentRule.ServiceAssignmentRuleId.
“Add new Service Assignment Rule”	Button	Switches to the dialog <b>DLG_ServiceAssignmentRuleDetail</b> (→ see chapter 0) to create a new service assignment rule. The dialog is opened in new mode.
“Delete”	Button	After displaying a confirmation popup delete the selected service assignment rules.
“Edit”	Button	Switches to the dialog <b>DLG_ServiceAssignmentRuleDetail</b> (→ see chapter 0) to edit the service assignment rule. The dialog is opened in new mode.
“View”	Button	Switches to the dialog <b>DLG_ServiceAssignmentRuleDetail</b> (→ see chapter 0) to edit the service assignment rule. The dialog is opened in read-only mode.

Table 95: Buttons and functions

#### 4.1.3.3 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Id	Label	The id of the service assignment rule.	ServiceAssignmentRule.ServiceAssignmentRuleId
Description	Label	The description of the service assignment rule.	ServiceAssignmentRule.Description
Service Table		The services covered by this service assignment rule.  Sorted by: Service.ServiceID	
Service-Name	Column	Lists the service names of all related services.	ServiceMaster.Name
Version	Column	Lists the versions of all related services.	Service.versionNumber
Service-ID	Column	Lists the service ids of all related services.	Service.serviceID
Services	Label	The services covered by this service assignment rule.	Service.Name + "(" + Service.ServiceID + ")" of all related services. Sort by Service.ServiceID.
Model Series	Label	Lists the model series this rule applies to.	ModelSeries.modelSeriesID of all related model series.
Sales Type Condition	Label	The wildcard for baumusters and the NST as defined by the optional sales type condition of this rule.	SalesTypeCondition.BaumusterWildcard + "/" + SalesTypeCondition.NST
Model Years	Label	Lists the model years this rule applies to.	ModelYear.Description + "(" + ModelYear.Code + ")" of all related model years. Sort by ModelYear.Code.
Equipment	Label	Lists the equipment codes needed or not allowed for this rule to apply in form of logical expression.	Boolean expression of Equipment Codes as defined using DLG_ServiceAssignmentRuleDetail. The boolean expression consists of Equipment Codes, displayed as EquipmentDescription.description + "(" + Equipment.Code + ")" connected by "or" or "and", marked by "not" operators, and grouped in brackets (see the example in the figure above).
Consumer Countries	Label	Lists the consumer countries for this rule to apply	<ul style="list-style-type: none"> <li>- If no ConsumerCountryCondition is associated to the ServiceAssignmentRule then write "ALL"</li> <li>- Otherwise list all countries referenced by the ConsumerCountryCondition: List of &lt;MBcCountry.name + "(" + MBcCountry.countryCode + ")"&gt;. Sort by MBcCountry.name.</li> </ul>
Table Column "Opened Sessions"	Label/Icon	Indicates whether the entity Document is edited inside a change session.	-

Table 96: Form fields and front-end data objects

#### 4.1.3.4 Dialog field validation

None.

#### 4.1.3.5 Configurability (incl. settings for roles)

None.

#### 4.1.3.6 Dialog Elements States

Linked Label	Type	State Description
Add new Model Series, Delete	Button	<u>Visible:</u> Always <u>Enabled:</u> If there is an open change session related to the current user
Wildcard / NST (Sales Type Condition)	Label	<u>Visible:</u> For each rule the NST part (SalesType.NST) is visible only if SalesTypeCondition.UseConditionNST = true. <u>Enabled:</u> -
All other elements	Varies	<u>Visible:</u> Always <u>Enabled:</u> Always

Table 97 DLG\_ServiceAssignmentRulesOverview

#### 4.1.4 DLG\_ServiceAssignmentRuleDetail

##### 4.1.4.1 General Description

This is the main dialog of the component Services. It allows to define service assignment rules by defining i) conditions to be fulfilled by vehicle configurations, and ii) services that are available for vehicles, when the conditions are fulfilled. For the definition of the conditions, several vehicle master data can be combined. For the definition of available services, these services can be selected. To this, the dialog consists of two sections in which the following vehicle and service master data can be combined for the definition service assignment rules:

1. Vehicle Conditions:

- A model series
- An optional condition on sales types.
- Year Code Combinations (each including a model year and an optional change year)
- Equipment codes
- Consumer countries

2. Rule Effects:

- Services - The services selected here are available for vehicles that match the conditions.

Note: As described in **Component „Vehicle Products“** (→ see chapter 3), the model series, equipment codes and services used in a service assignment rule cannot be deleted.

Hint: The validation described here is not new. Only the documentation is moved from here to section “Dialog field validation” below.

Service Assignment Rule ID: 1

Description:

Long description:

The vehicle must be of the following selected model series:

The vehicle must be of one of the following sales types:

ALL sales types whose baumuster starts with   Currently, the following sales types are selected by the above wildcard definition:

The year code combination (model year + optional change year) must be one of the selected:

Select/Deselect all  
 800  
 801  
 + optional Change Year:   
 802  
 803  
 804  
 805

The vehicle must fulfill the following equipment conditions

not   or  not   or   
 and  
 not   or  
 and  
 not   or  
 and

Consider the following consumer countries for this rule  
 Select/Deselect all  
 Austria (AT)  
 Belgium (BE)  
 Denmark (DK)  
 France (FR)  
 Germany (DE)

If the above conditions are fulfilled, all of the following selected services are available for the vehicle:

Accident Management - Version 1 (ID 1)  
 Accident Management - Version 2 (ID 11)  
 Breakdown Management - Version 1 (ID 2)  
 Breakdown Management - Version 2 (ID 12)  
 Maintenance & Predictive Diagnosis - Version 1 (ID 3)

Figure 43: DLG\_ServiceAssignmentRuleDetail for passenger cars

#### 4.1.4.2 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		<p><u>Edit, Read-only mode:</u> Load the service assignment rule and all related model series, year code combinations (including model years and optional year code), equipment codes and services. Also load all other available model series, year code combinations, equipment and services and display them. Mark those that are already related to the service assignment rule as checked.</p> <p><u>New mode:</u> Load all available model series, year code combinations and services. Additionally load these Equipment entities with Equipment.RelevantForServiceAssignmentRules="true". The service assignment rule id will be empty until service assignment rule is saved.</p> <p>Consider the loading algorithm described in Chapter "<b>Loading of master data elements</b>"</p>
“Save”	Button	<p>Save the currently maintained service assignment rule by calling AF_SaveServiceAssignmentRule (→ see chapter 4.5.22). and navigate back to DLG_ServiceAssignmentRulesOverview (→ see chapter 4.1.3).</p> <p>Bring the currently maintained ServiceAssignmentRule in the change session of the user. (see general algorithmApplyChangeSessionOnElement)</p>
“Cancel”	Button	Discard all changes and navigate back to DLG_ServiceAssignmentRulesOverview (→ see chapter 4.1.3).
“check/uncheck all”	Button	Selects or deselects all check boxes in the section.
Vehicle conditions / Boolean Term / Equipment Condition / NEGATE Button	ToggleButton	<p>Include or exclude (toggle) the equipment code in the given Equipment Condition.</p> <p>Note here that “Equipment Condition” denotes the group of the user interface elements “Negate button”, “Equipment code selector” and “Delete button”, a new instance of which is displayed when an OR or AND button is clicked.</p> <p>If the button is pressed, the inclusion of the selected equipment code is explicitly negated. Thus, to match this part of Boolean expression a vehicle must not have this equipment.</p>
Vehicle conditions / Boolean Term / Equipment Condition / Delete Button	Button	Delete selected equipment code from equipment condition. It allows correcting the expression of the Equipment Condition by deleting the selection at any position of expression.
Vehicle conditions / Boolean Term / OR Button	Button	Add additional equipment code selector to the right with its Negate, Delete and new OR buttons. The clicked OR button will be deactivated. All selected equipment codes lying in the same line are connected by OR-operator.
Vehicle conditions / Boolean Term / AND Button	Button	Add another Equipment Condition (equipment code selector, Delete and Negate buttons) and new AND Button below of the existing Equipment Conditions. It allows adding another equipment code or a set of them connected by OR operator to the existing Boolean expression. This additional Equipment Condition added by AND button is connected by the AND-operator to the rest of Boolean expression. The clicked AND button will be deactivated.
Vehicle conditions / Model Series List	Dropbox	On selection of a model series, if the rule contains a Sales Type condition, a warning message ("You are about to change the selected model series. This will reset the Sales Type condition. Ok?") is shown,

<b>Linked label / button labeling</b>	<b>Type</b>	<b>Action description</b>
		and, if confirmed, section "Sales Type Condition" is reset.
Vehicle conditions / Sales Type Condition / Baumuster Wildcard Checkbox	Checkbox	Activates the section for the Sales Type Condition.
Checkbox for NSTs (Vehicle conditions / Sales Type Condition)	Checkbox	Optionally activatable if the selected model series' product group is 'P' (passenger cars), if NSTs are (to be) defined as conditions for sales types, using checkbox list "NST List". For transporters (if the selected model series' product group is 'T') this checkbox is <u>always</u> unchecked, i.e. SalesTypeCondition.UseConditionNST=false).
"Preview"	Button	Determine the sales types of the specified model series that meet the sales type condition defined by the baumuster wildcard and the optional list of NST codes by calling IIF_GetSalesTypesForCondition of component "Vehicle Products", and show these sales types in the Preview List.
Consumer Country Condition	Checkbox	Activates the section for the consumer countries.
Consumer countries	Checkbox	Optionally activated, if consumer countries are (to be) defined as conditions for service availability in those countries

Table 98: Buttons and functions

#### 4.1.4.3 Form fields and front-end data objects (AS08 reference)

<b>Linked Label</b>	<b>Type</b>	<b>Details / Default</b>	<b>Name Business Object. Attribut (AS08 reference)</b>
ServiceAssignmentRuleId	Label	The id of the service assignment rule.	ServiceAssignmentRule. ServiceAssignmentRuleId
Description	Textbox	The description of the service assignment rule.	ServiceAssignmentRule.Description
LongDescription	Textarea	The long description of the service assignment rule.	ServiceAssignmentRule.LongDescription
Vehicle conditions / Model Series List	Dropbox	The list of available model series	ModelSeries.ModelSeriesId  <u>Selected:</u> The model series that is related to the service assignment rule.  <u>Available values:</u> All available Model Series
Vehicle conditions / Sales Type Condition / Baumuster Wildcard Checkbox	Checkbox	Checked, if a sales type condition exists and is to be used.	SalesTypeCondition.UseConditionBaumusterWildcard  <u>Checked:</u> Iff SalesTypeCondition.UseConditionBaumusterWildcard.
Vehicle conditions /	Textbox	The wildcard	SalesTypeCondition.BaumusterWildcard

Sales Type Condition / Baumuster Wildcard		definition. Here: "204.0**" The wildcard as part of a sales type condition defines the set of sales types (baumusters) to be considered by this assignment rule.	This element is active, only if UseConditionBaumusterWildcard is set (see definition of dialog element states).
Vehicle conditions / Sales Type Condition / NST Checkbox	Checkbox	Checked, if a sales type condition exists and contains a list of NST codes to be used.	<p>SalesTypeCondition.UseConditionNST</p> <p><u>Checked:</u> Iff SalesTypeCondition.UseConditionNST.</p> <p>This element is active, only if UseConditionBaumusterWildcard is set (see definition of dialog element states).</p> <p>For transporters (i.e. iff the selected model series' product group is 'T') holds SalesTypeCondition.UseConditionNST = false. Hence, this checkbox is constantly unchecked.</p>
Vehicle conditions / Sales Type Condition / NST List	Checkbox list	List of NST codes available for the selected model series (here: 204). NST codes can be selected as part of a sales type condition to define conditions for the sales types to be selected.	<p><u>Checked:</u> The NST-Codes related to the service assignment rule.</p> <p><u>Active:</u> Iff (SalesTypeCondition.UseConditionNST AND )</p> <p><u>Available values:</u> All NST codes available for the model series selected in the Model Series List.</p> <p>This element is active, only if UseConditionBaumusterWildcard and UseConditionNST are set (see definition of dialog element states).</p> <p>For transporters (i.e. iff the selected model series' product group is 'T') holds SalesTypeCondition.UseConditionNST = false. Hence, this element is constantly disabled and all NSTs are unchecked in this case.</p>
Vehicle conditions / Sales Type Condition / Preview List	List	List of sales types that meet the sales type conditions as defined by the baumuster wildcard and optional list of NST codes.	SalesTypeDescription.description + " (" + SalesType.baumuster + " " + SalesType.NST + ")"
Vehicle conditions / Year Code Combination List / Model Year Code List	Checkbox list	The list of available model years.	<p>ModelYear.Code</p> <p><u>Checked:</u> The model year codes that are related to the service assignment rule (YearCodeCombination.ModelYearCode).</p>

			<p><u>Available values:</u> Call IIF_GetModelYearsByProductGroup to get all available ModelYears that have the same product group as the selected model series.</p> <p><u>Sorted by:</u> ModelYear.Code</p>
Vehicle conditions / Year Code Combination List / Change Year Code	Textbox	The optional change year code for each ModelYear.code	<p>Only active if corresponding Model Year Code is checked.</p> <p>For each selected Model Year Code (YearCodeCombination.ModelYearCode) the corresponding (optionally filled) YearCodeCombination.ChangeYearCode.</p>
Vehicle conditions / Boolean Term / Equipment Condition / NEGATE Button	ToggleButton	Allows activating of the including (initial) or excluding (negated) status of selected equipment code.	<p>OrTerm.Negated</p> <p><u>Pressed:</u> If OrTerm.Negated is true. (To match the Equipment condition of service assignment rule vehicle must not have the selected equipment code)</p> <p><u>Not pressed (initial state):</u> If OrTerm.Negated is false. (The vehicle must have the selected equipment to fulfill the Equipment condition of service assignment rule)</p>
Vehicle conditions / Boolean Term / Equipment Condition / Equipment List	Dropbox	The list of available equipment. The name/description is displayed localized to the user's locale. There may be several of these dropdowns in use within one Boolean expression.	<p>EquipmentDescription.Description + "(" + Equipment.Code + ")"</p> <p><u>Selected:</u> The equipment code that is part of the Boolean expression related to the service assignment rule</p> <p><u>Available values:</u> All available Equipment entities with Equipment.RelevantForServiceAssignmentRules="true" that have the same product group as the selected model series.</p> <p><u>Sorted by:</u> EquipmentDescription.Description</p>
Consumer Country Condition	Checkbox	Checked, if a consumer country condition exists and is to be used.	Label: "Consider the following consumer countries for this rule"
Consumer country List	Checkbox List	A list of countries is defined as condition for service availability in those countries	<p><u>Checked:</u> The consumer country related to the service assignment rule.</p> <p><u>Active:</u> If (ServiceAssignmentRule.useConsumerCountryCondition is true )</p> <p><u>Available values:</u> All MBcCountry returned by the call <b>IIF_GetMBconnectCountries</b></p>
Rule Effects / Service List	Checkbox list	The services that will be available if the rule applies.	ServiceMaster.Name + " - Version " + Service.versionNumber + " (" + "ID " + Service.ServiceId + ")"

		<p>The name is displayed localized to the user's locale.</p>	<u>Checked:</u> The services that are related to the service assignment rule.  <u>Available values:</u> All available Services  <u>Sorted by:</u> Service.ServiceId
ReadOnlyInfo	Label	<p>Presents this information:            "The dialog is opened in read-only mode."</p>	-

Table 99: Form fields and front-end data objects

#### 4.1.4.4 Dialog field validation

Linked Field	Validation	Errormessage
Model Series List, Model Year List and Service List	In all of the lists at least one item must be selected.	SERMAN_007
Model Series List, BaumusterWildcard	The first three digits of the baumuster wildcards have to match the model series in order to save the service assignment rule	SERMAN_020
Equipment List, OR Button, AND Button	When an OR or an AND button is clicked, validate if a code is selected in <i>one of the last</i> dropdown in the Equipment Condition.	SERMAN_021

Table 100: Dialog field validations

#### 4.1.4.5 Configurability (incl. settings for roles)

None.

#### 4.1.4.6 Dialog Elements States

Linked Label	Type	State Description
ReadOnlyInfo	Label	<u>Visible:</u> if dialog is in read-only mode <u>Invisible:</u> if dialog is in edit mode <u>Enabled:</u> -
<all elements in section Sales Type Condition except NST List, NST Checkbox>	Varies	<u>Visible:</u> If SalesTypeCondition.UseConditionBaumusterWildcard <u>Enabled:</u> If SalesTypeCondition.UseConditionBaumusterWildcard and the dialog is in edit mode
NST Checkbox	Checkbox	<u>Visible:</u> If SalesTypeCondition.UseConditionBaumusterWildcard and the selected model series' product group is 'P' <u>Enabled:</u> If SalesTypeCondition.UseConditionBaumusterWildcard and the selected model series' product group is 'P' and the dialog is in edit mode
NST list	Checkbox list	<u>Visible:</u> SalesTypeCondition.UseConditionBaumusterWildcard and UseConditionNST are set and the selected model series' product group is 'P' <u>Enabled:</u> If SalesTypeCondition.UseConditionBaumusterWildcard and UseConditionNST are set and the selected model series' product group is 'P'

Preview	Button	<u>Visible:</u> SalesTypeCondition.UseConditionBaumusterWildcard <u>Enabled:</u> If SalesTypeCondition.UseConditionBaumusterWildcard
all other elements except for "Cancel"	Varies	<u>Visible:</u> Always <u>Enabled:</u> if dialog is in edit mode <u>Disabled:</u> if dialog is in read-only mode
"Cancel"	Varies	<u>Visible:</u> Always <u>Enabled:</u> Always

Table 101: Dialog Elements States

## 4.1.5 DLG\_ServiceAssignmentRulesTestSimulation

### 4.1.5.1 General Description

This dialog allows the user to test the MBconnect service assignment rules. It allows testing for the available services for:

- a specific vehicle
- on a specific date
- on a specific customers address country
- on a specific consumer country
- from a specific business area
- on a specific source: either productive data or data from the users change session.

Moreover it also includes testing for a specific vehicle with the VIN of a vehicle or with model series, year code combination (including model year and optional change year) and MBconnect equipment of a specific vehicle.

Additionally, the possibility to test and visualize from different sources is given:

- "Productive": The testing and visualizing refers to the data from the productive system.
- "My Change Session": The testing and visualizing refers to the data from the personal change session and the productive system. Even the input of the vehicle data (Model Series, Year Code Combination (Model Year and optional Change Year) MBconnect equipment) will be loaded from the personal change session and the productive system.

*Note:* The SOE test simulator is not able to test if a vehicle technically supports a service. Because it is only possible to test vehicle and service master data inside the SOE, the output will display only the possible services from a sales perspective.

If a FIN is given, then the <consumer country> will when computing the available services. Currently the system ODC is not able to deliver the <consumer country> for a specific FIN.

Testing Criteria

Source*	Productive
Date (service availability)* (DD.MM.YYYY)	13.06.2014
Address Country*	Germany
Business Area *	All

Vehicle Data

<input checked="" type="radio"/> VIN available	VIN* WDD1690711K23J
<input type="radio"/> VIN not available (BTO vehicle)	
Model Series*	
Sales Type*	
Year Code Combination*	
Consumer Country	Germany
MBconnect equipment	<input type="checkbox"/> 06U MBconnect Basisdienste <input type="checkbox"/> 05U MBconnect SA Remote Online <input type="checkbox"/> 350 MB eCall Europa <input type="checkbox"/> 228 Standheizung <input type="checkbox"/> B54 Live Traffic <input type="checkbox"/> B57 ZEV Services

\* Mandatory field(s)

Figure 44 DLG\_TestServiceAssignmentRules

#### 4.1.5.2 Buttons and functions

Linked Label / Button Labeling	Type	Action Description
<init>		Initialize the fields below as follows in 4.1.5.3.
“Show Results”	Button	Calls AF: AF_TestServiceAssignmentRules (→ see chapter 4.5.31) with parameters <ul style="list-style-type: none"> <li>- productiveStatus (from Source: If “Productive” is selected, then return true, else return false)</li> <li>- evaluationDate (from Date (service availability))</li> <li>- BusinessArea</li> <li>- addressCountry (from Country)</li> <li>- FIN/VIN or VehicleConfiguration (from Vehicle Data)</li> <li>- Locale (from the language of the dialog)</li> </ul> and create a Timestamp for DLG_ServiceAssignmentRulesTestResults.
“Clear Input”	Button	Clear Input fields and reinitialize DLG_ServiceAssignmentRulesTestSimulation.

Table 102: Buttons and functions

#### 4.1.5.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Source	Drop-Down	The list of available data sources.  Default: empty	<u>Available values:</u> {"“Productive”, “My Change Session”}
Date (service availability)	Textbox	Specifies the date to be considered when testing the service availability  Default: “TODAY”	- (DataType: DateDT)
AddressCountry	Drop-Down	Specifies with which addresscountry the input is tested.  Default: empty	<u>Selected value:</u> MbcCountry.countryCode  <u>Available values:</u> All countries that are supported by MBconnect. For a list of countries, <b>IIF_GetMBconnectCountries</b> is called
Business Area	Drop-Down	Filters the available services by their business area.  Default: “All”.	<u>Selected value:</u> BusinessArea.  <u>Available values:</u> “B2C”, “B2B” and “All”.
VIN available	Radio button	Selection of this radio button for a test with a given VIN needed.	-
VIN/FIN	Textbox	The FIN or the VIN of a specific vehicle.	-
VIN not available	Radio button	Selection of this radio button for a test with a (freely chosen) vehicle configuration needed.  Default: selected	-
Model Series List	Drop-Down	The list of available model series.	<u>ModelSeries.ModelSeriesId</u>  <u>Selected:</u> The model series that is related to the tested vehicle (ModelSeries.ModelSeriesId).  <u>Available values:</u> All available Model Series depending on the context: - If the Source is “Productive” then only the productive form of the available model series master data is loaded. - If the Source is “My change session” then change and productive form of the available model series master data is loaded.  <u>Sorted by:</u> ModelSeries.ModelSeriesId
Sales Type List	Drop-Down	The list of available sales types.	<u>SalesTypeDescription.description + “(“ + Key of SalesType “)”</u>  For “Key of Sales Type” print only those attributes which are not empty.  <u>Selected:</u> The sales type of the vehicle to be tested.  <u>Available values:</u> - If Source is “Productive”, then only the productive sales types are available. - If Source is “My change session”, then changed and

			productive sales types are available.  <u>Sorted by:</u> SalesTypeDescription.description
Year Code Combination List	Drop-Down	The list of available year code combinations.	If YearCodeCombination.ChangeYearCode is not empty: YearCodeCombination.ModelYear-Code + "}" + YearCodeCombination.ChangeYearCode Else YearCodeCombination.ModelYearCode  <u>Selected:</u> The year code combination that is related to the tested vehicle.  <u>Available values:</u> All available distinct Year Code Combinations.  <u>Sorted by:</u> YearCodeCombination.ModelYearCode
ConsumerCountry	Drop-Down	the consumer country of the vehicle for which the availability of services is to be tested  Default: empty	<u>Selected value:</u> MbcCountry.countryCode  <u>Available values:</u> All countries that are supported by MBconnect. For a list of countries, <b>IIF_GetMBconnectCountries</b> is called
Equipment List	Checkbox list	The list of available equipment.	Equipment.Code + EquipmentDescription.Description  <u>Checked:</u> The equipment that are related to the tested vehicle (Equipment.Code).  <u>Available values:</u> All available Equipment entities with Equipment.RelevantForServiceAssignmentRules="true" that have the same product group as the selected model series depending on the context: - If the Source is "Productive" then only the productive form of the available model series master data is loaded. - If the Source is "My change session" then change and productive form of the available model series master data is loaded.  <u>Sorted by:</u> Equipment.Code

Table 103: Form fields and front-end data objects

#### 4.1.5.4 Dialog field validation

Linked Field	Validation	Errormessage
Date (service availability)	The "Date (service availability)" has to be today or in the future. The "Date (service availability)" isn't allowed to be in the past.	SERMAN_016
Source Date (service availability) Country VIN available > VIN/FIN VIN not available > Model Series > Year Code Combination	"Source", "Date (service availability)", "Country" cannot be left blank.  If "VIN available" is selected "VIN/FIN" cannot be left blank either. If "VIN not available" is selected, "Model Series" and "Year Code Combination" cannot be left blank either.	SERMAN_017

Table 104: Dialog field validations

---

#### 4.1.5.5 Configurability (incl. settings for roles)

None.

#### 4.1.5.6 Dialog Elements State

Linked Label	Type	State Description
VIN/FIN	Textbox	<u>Visible:</u> Always <u>Enabled:</u> The Textbox is only editable if the checkbox "VIN available" is selected.
Model Series List	Drop-Down	<u>Visible:</u> Always <u>Enabled:</u> The Drop-Down is only editable if the checkbox "VIN not available" is selected.
Sales Type List	Drop-Down	<u>Visible:</u> Always <u>Editable:</u> The Drop-Down is only editable if the checkbox "VIN not available" is selected.
Year Code Combination List	Drop-Down	<u>Visible:</u> Always <u>Editable:</u> The Drop-Down is only editable if the checkbox "VIN not available" is selected.
ConsumerCountry List	Drop-Down	<u>Visible:</u> Always <u>Editable:</u> The Drop-Down is only editable if the checkbox "VIN not available" is selected.
Equipment List	Checkbox List	<u>Visible:</u> Always <u>Editable:</u> The Checkbox List is only editable if the checkbox "VIN not available" is selected.

Table 105: Dialog Elements State

### 4.1.6 DLG\_ServiceAssignmentRulesTestResults

#### 4.1.6.1 General Description

The result dialog visualizes the input data from the test simulation dialog and visualizes the vehicle configuration even if the VIN/FIN was entered as vehicle data. The main part of the dialog is the visualization of the supported Mercedes Connect me services as the result of the test in a table, which is shown in **Error! Reference source not found..**

If there are more than one service assignment rules and corresponding equipment codes match a service, these service assignment rules and corresponding equipment codes are visualized as a sub table for the service. That means, one row of service in the result table can have more than one rows of service assignment rules and corresponding equipment codes.

Testing Input

Test Date/Time: 2015-12-14 - 14:31  
 Source: Productive  
 Date (service availability): 2015-12-14  
 Address Country: Germany  
 Consumer Country: Germany  
 FIN: WDD2074731F312768

Vehicle Data

Model Series: 207  
 Sales Type: 207.473.1 CH1  
 Year Code Combination: 806  
 MBconnect equipment:

- 05U MBconnect SA Remote Online
- 06U MBconnect Basisdienste
- 228 Standheizung
- 350 MB eCall Europa
- B54 Live Traffic

Supported MBconnect services

S-ID	Service-Name	SAR	Ma.-Codes
1	Live Traffic Information	6078	B54 + 06U
2	Mercedes-Benz emergency call system	6079	350
10	Parked Vehicle Locator	6081	05U
21	Remote Retrieval of Vehicle Status	6081	05U
22	Programming of Auxiliary Heating	6082	05U + 228
30	Maintenance Management	6080	06U
31	Telediagnostics	6080	06U
32	Accident Management	6080	06U
33	Breakdown Management	6080	06U
34	Vehicle Tracker	6081	05U

[Go back for a new test simulation](#)

Figure 45 DLG\_ServiceAssignmentRulesTestResults

#### 4.1.6.2 Buttons and functions

Linked Label / Button Labeling	Type	Action Description
<init>		
"Go back for a new test simulation"	Button	Clear Output fields and reinitialize DLG_ServiceAssignmentRulesTestSimulation.

Table 106: Buttons and functions

#### 4.1.6.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Testing input: Test Date/Time	Label	Lists the date and the time the user pushed the "Show Results" Button.	Initialize with the value of the linked timestamp from "Show Results".
Testing Input: Source	Label	Lists the source of the tested data.	Initialize with the value of the linked selected value "Source".
Testing Input: Date (service availability)	Label	Lists the date on which the selected vehicle configuration was tested.	Initialize with the value of the linked selected value "Test Date".
Testing Input: Address Country	Label	Lists the country in which the selected vehicle configuration was tested.	Initialize with the value of the linked selected value "Test Address Country".
Testing Input: FIN	Label	If a FIN was given as input, the FIN is displayed here.	The FIN of the vehicle (if given as input).
Testing Input: Consumer Count- ry	Label	Lists the consumer country for which the selected vehicle configuration was tested.	Initialize with the value of the linked selected value "Test Consumer Country".
Model Series	Label	Lists the model series the selected vehicle configuration applies to.	If "VIN available" is selected: ModelSeries.modelSeriesID returned from AF_TestServiceAssignmentRules. If "VIN not available" is selected: Initialize with the value of the linked selected value "Model Series List".
Sales Type	Label	Displays the key of the Sales Type.	If "VIN available" is selected: The concatenated attributes of the sales type forming the key (only the attributes which are not empty) returned from AF_TestServiceAssignmentRules. If "VIN not available" is selected: Initialize with the value of the linked selected value "Sales Type List".

Year Code Combination	Label	<p>Lists the year code combination the selected vehicle configuration applies to.</p>	<p>If “VIN available” is selected and ChangeYearCode is not empty: ModelYear.Code + “+” + ChangeYearCode returned from AF_TestServiceAssignmentRules</p> <p>else if “VIN available” is selected: ModelYearCode returned from AF_TestServiceAssignmentRules</p> <p>else if “VIN not available” is selected: Initialize with the value of the linked selected value “Year Code Combination” List.”</p>
Consumer Country	Label	<p>Lists the consumer country for which the selected vehicle configuration was tested.</p>	<p>If “VIN available” is selected: ConsumerCountryCode returned from AF_TestServiceAssignmentRules.</p> <p>If “VIN not available” is selected: Initialize with the value of the linked selected value “Test Consumer Country”.</p>
MBconnect equipment	Label	<p>Lists the equipment the selected vehicle configuration applies to.</p>	<p>If “VIN available” is selected: Equipment.Code + EquipmentDescription.description. Sort by Equipment.Code returned from AF_TestServiceAssignmentRules.</p> <p>If “VIN not available” is selected: Initialize with the value of the linked selected value “Equipment List”.</p>
<b>Supported MBconnect Services Table</b>			If AF_TestServiceAssignmentRules returns an empty list of services, then return “no services available”.  <u>Sorted by:</u> Service.ServiceID
Service-ID	Column	<p>Lists the service ids by the required services.</p>	Service.ServiceID returned from AF_TestServiceAssignmentRules.
Service-Name	Column	<p>Lists the service names by the required services.</p>	ServiceMaster.Name returned from AF_TestServiceAssignmentRules.
Version	Column	<p>Lists the version numbers by the required services.</p>	Service.versionNumber returned from AF_TestServiceAssignmentRules
BusinessArea	Column	<p>Lists the business area of the corresponding service master.</p>	ServiceMaster.BusinessArea returned from AF_TestServiceAssignmentRules.
SAR-ID	Column	<p>Lists the service assignment rule ids required by the services.</p>	RuleID returned from AF_TestServiceAssignmentRules.
MatchedCodes	Column	<p>Lists the matched equipment codes required by the service assignment rules.</p>	MergedMatchCodes returned from AF_TestServiceAssignmentRules.

Table 107: Form fields and front-end data objects

#### 4.1.6.4 Dialog field validation

None.

#### 4.1.6.5 Configurability (incl setting for roles)

None.

#### 4.1.6.6 Dialog Elements State

Linked Label	Type	State Description
Supported MBconnect Services Table	Table	<p><u>Visible:</u> The Table is only visible if AF_TestServiceAssignmentRules returns a non-empty list of services.</p> <p><u>Enabled:</u> -</p>
Testing Input: FIN	Label	<p><u>Visible:</u> If a FIN was given as input of the test simulation.</p> <p><u>Enabled:</u> -</p>

Table 108: Dialog Elements State

## 4.2 External View - Offered Interfaces

### 4.2.1 IF\_SOE\_CheckVehicle

#### 4.2.1.1 General Description

**Communication type:** Synchronously

This interface determines the set of services supported by a given vehicle or vehicle configuration for a given customer. A vehicle is given by its FIN or VIN, a vehicle configuration by its baumuster, NST, year code combination (including a model year and an optional change year), equipment, and the country it is produced for. Regarding the customer, it is sufficient to consider his or her address country for the purposes of this interface. In the FIN-case, the vehicle's configuration, i.e. baumuster, year code combination (including a model year and an optional change year) and equipment, optionally including descriptions is determined additionally.

Furthermore, this interface identifies the vehicle type of the vehicle or configuration given as input. The vehicle type indicates if it supports any services, and in case it does which class of services, i.e. adapter- or Mercedes connect me-services.

Moreover, this interface optionally allows to compare the current vehicle or vehicle configuration in a given address country to a second "old" vehicle or vehicle configuration in another address country with respect to the supported services. For this purpose, the interface returns a set of services that is additionally supported by the current vehicle or configuration in comparison to the old vehicle or configuration, and a set of services that is supported by the old vehicle or configuration but not by the current vehicle or configuration.

This interface is implemented by AF\_CheckVehicle.

#### 4.2.1.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Address countries</b>						
addressCountry	Mand	String	2	MbcCountry.countryCode	Examples: "DE", "CH", "UK", ...	The address country of the user. Based on this country, the list of available services is determined.
oldAddressCountry	Opt.	String	2	MbcCountry.countryCode	Examples: "DE", "CH", "UK", ...	Based on this address country the list of services that are additionally available /removed in

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						addressCountry is determined.
<b>Language information</b>						
locale	Opt.	String	5	Used as input to determine localized descriptions (from ODC).	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the retuned vehicle- or equipment- descriptions.
Vehicle for which the serviceList needs to be determined						
finOrVin	Mand. If baumuster not given	String	17	-	Example: WDD1690341 J764507	The given VIN/FIN.
baumuster	Mand. If finOr Vin not given	BaumusterDT		SalesType.baumuster	Example: "2050041"	The baumuster as part of the sales type.
NST	Opt.	NstDT		SalesType.Nst	Example: „CH1“	The NST code (for passenger cars, i.e. product group ‘P’) of the sales type.
ModelYearCode	Mand. If finOr Vin not given	String	5	YearCodeCombination.ModelYearCode	Example: "804"	The code of the model year. Mapping: Add the YearCodeCombination.ModelYearCode to the set of equipments.
ChangeYearCode	Opt.	String	5	YearCodeCombination.ChangeYearCode	Example: "054"	The code of the change year.
ConsumerCountryCode	Mand. If finOr Vin not given	String	2		AT, DE, CH	The ISO code of the consumer country.
List of MBconnect relevant equipment codes (only given, if finOrVin is not given) (0 to n equipment codes)						
Code	Mand.	String	5	Equipment.Code	Example: "527"	The equipment code of the given vehicle configuration.
Old vehicle – based on this the delta of the services needs to be determined (optional)						
oldFinOrVin	Mand. If oldBaumuster not given.	String	17	-	Example: WDD1690341 J764507	The old VIN/FIN.
oldBaumuster	Mand. If oldFinOrVin is not given	String	7	SalesType.baumuster	Example: "2050041"	The baumuster as part of the sales type of the given old vehicle configuration.
oldNST	Opt.	NstDT		SalesType.NST	Example: "DE1"	The NST code (for passenger cars, i.e. product group ‘P’) of the sales type.
oldModelYearCode	Mand. If	String	5	YearCodeCombination.ModelYearCode	Example: "804"	The code of the model year of the given old vehicle

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
	oldFinOrVin is not given					configuration. Mapping: Add the YearCodeCombination.ModelYearCode to the set of equipments.
oldChangeYearCode	Opt.	String	5	YearCodeCombination.ChangeYearCode	Example: "054"	The code of the change year of the given old vehicle configuration.
List of MBconnect relevant equipment codes (only given, if oldFinOrVin is not given) (0 to n equipment codes)						
oldCodeList	Mand.	String	5	Equipment.Code	Example: "527"	The equipment code of the given old vehicle configuration.

Table 109: IF\_SOE\_CheckVehicle Input

#### 4.2.1.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Vehicle</b>						
vehicleType	Mand.	Enum		VehicleTypeEnum	ADAPTER_VEHICLE, CONNECT_VEHICLE, NONE	Indicates the type of service the given vehicle or vehicle configuration supports. Mercedes connect me-vehicles are vehicles that have a communication module. Adapter-vehicles are vehicles that do not have a communication module but an OBD II-adapter interface. Type NONE: Vehicles that neither have a communication module nor an OBD II-adapter interface do not support any services integrated in Mercedes connect me.
<b>Vehicle details (only returned if VIN/FIN was given)</b>						
FIN	Mand.	String	17	-	Example: WDD1690341J764507	The FIN of the given vehicle.
VIN	Opt.	String	17	-		The VIN of the given vehicle.
<b>• Sales Type</b>						
Baumuster	Mand.	String	9	SalesType.Baumuster	Example: „204.231.2“	The baumuster id of the sales type.
NST	Opt.	NstDT		SalesType.NST	Example: „CH1“	The NST code (for passenger cars, i.e. product group 'P') of the sales type.
BaumusterDescription	Mand.	String	256	SalesType.Description (localized)	Example: „C 180 BlueEFFICIENCY mystar Estate“	The description of the baumuster in the requested locale. If no translation is available, the value from the field Baumuster is returned.
<b>• List of MBconnect relevant equipment codes for the vehicle (there can be 0 to n equipment codes)</b>						
Code	Mand.	String	5	Equipment.Code	Example: "527"	The equipment code.
CodeDescription	Mand.	String	256	Equipment.Description (localized)	Example: “COMAND Online system with Media Interface”	The description of the equipment in the requested locale. If no translation is available, the value from the field Code is returned.
<b>• Upholstery</b>						

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
UpholsteryCode	Opt.	String	5	Equipment.Code	Example: "123"	The code of the Upholstery.
Upholstery-Description	Opt.	String	256	Equipment.Description (localized)	Example: "Leder Alcantara"	The description of the Upholstery in the requested locale as returned from UVS.
<b>• Paint 1</b>						
Paint1-Code	Opt.	String	5	Equipment.Code	Example: "123"	The code of the Paint1.
Paint1-Description	Opt.	String	256	Equipment.Description (localized)	Example: "Brilliantgrau"	The description of the Paint1 in the requested locale as returned from UVS.
<b>• Paint 2</b>						
Paint2-Code	Opt.	String	5	Equipment.Code		The code of the Paint2.
Paint2-Description	Opt.	String	256	Equipment.Description (localized)		The description of the Paint2 in the requested locale as returned from UVS.
<b>• Registration Date</b>						
FirstRegistrationDate	Opt.	Date	-	VehicleConfiguration.VehicleCoreData.firstRegistrationDate	Example: "20130130"	The first registration date if available.
<b>List of MBconnect services supported by the given vehicle and available in the requested addressCountry (0...*)</b>						
ServiceID	Mand.	Int	-	Service.ServiceID	Examples: 1, 12356, 1234567890	The service ID identifies the specific service.
<b>Delta list of services (only returned if oldAddressCountry/old vehicle was given)</b>						
additionalServices	Opt.	List of Int.	-	List < Service.ServiceID >	Examples: 1, 12356, 1234567890	IDs of the services, that are additionally supported by the given vehicle in the requested addressCountry (referred to the oldAddressCountry/old vehicle)
removedServices	Opt.	List of Int.	-	List < Service.ServiceID >	Examples: 1, 12356, 1234567890	IDs of the services, that are not supported anymore by the given vehicle in the requested addressCountry (referred to the oldAddressCountry/old vehicle)

Table 110: IF\_SOE\_CheckVehicle Output

#### 4.2.1.4 Exceptions

Code	Message
ACCDAS_005	Action is not possible, because the customer is notified in an unsupported country.
VEHPRO_005	The given vehicle cannot be found.

Table 111: Table of exceptions (IF\_SOE\_CheckVehicle)

#### 4.2.2 IF\_SOE\_GetRequiredFieldsUsedByServices

##### 4.2.2.1 General Description

**Communication type:** Synchronously

---

This interface returns the required data fields for the given list of service IDs. The result contains account as well as vehicle related data fields.

Note:

- Additionally to the profile data fields, this interface returns structural information on the data fields. For further information see **Configuration of Profile Data Fields** (→ see chapter 2.3.19).

Internally, AF\_GetRequiredFieldsUsedByServices will be called.

#### 4.2.2.2 Input

Parameter Name	Mand./ Opt.	For- mat	Length	Data Model	Possible Values	Annotation
Locale	Opt.	String	5	Used as input parameter for AF_GetRequiredFieldsUsedByServices	Example: "de_De"	The locale is used to retrieve the internationalized country name. If not given a default will be used.
serviceIDs	Mand.	List of Integer	-	Used as input parameter for AF_GetRequiredFieldsUsedByServices	-	The service IDs

Table 112: IF\_SOE\_GetRequiredFieldsUsedByServices input

#### 4.2.2.3 Output

Parameter Name	Mand./ Opt.	For- mat	Length	Data Model	Possible Values	Annotation
<b>List of services (each of the following entries exists per service) (0...*)</b>						
serviceID	Mand.	Integer	-	Service.serviceID		-
serviceName	Mand.	String	-	ServiceMaster.name		The internationalized service name.
<b>• List of required items (inner list per service) (0...*)</b>						
relationshipType	Mand.	Enumeration	-	As returned by the underlying AF.	{GROUP, DATAFIELD}	Determines if structureItemID references to a data field or to a group.
itemID	Mand.	String / Enumeration	-	As returned by the underlying AF.		The id of a group field or a data field. In case itemID identifies a group its type is String. In case itemID identifies a data field, its type is Enumeration (an entry of).
fieldOwnerType	Mand.	Enumeration	-	As returned by the underlying AF.	{ACCOUNT, VEHICLE}	Specifies if the required field is account or vehicle related. In case of group, the field owner type will be empty.
childrenIDs	Opt.	List of Enumeration	-	As returned by the underlying AF.	{see list in chapter 2.3.18}	The IDs of the required dependent fields. <u>At least one</u> of the child profile data fields has to be filled.  For the available values please refer to <b>Handling of Profile Data within SOE</b> (→ see chapter 2.3.18).

Parameter Name	Mand./Opt.	For-mat	Length	Data Model	Possible Values	Annotation
• List of custom properties (inner list per service) (0...*)						
customProperty	Mand.	String		ServiceMaster.customProperty	-	The name of the custom property.

Table 113: IF\_SOE\_GetRequiredFieldsUsedByServices output

#### 4.2.2.4 Exceptions

If a serviceID is not found then exception SERMAN\_008 is thrown.

### 4.2.3 IF\_SOE.GetServiceCategories

#### 4.2.3.1 General Description

**Communication type:** Synchronously

This application function returns service categories available in SOE.

If a list of service category IDs is given, only the according service categories are returned. Otherwise all service categories are returned. If a locale is given, only the service category name available for this locale is returned. Otherwise the service category names for all available locales are returned.

This interface is implemented by AF.GetServiceCategories.

#### 4.2.3.2 Input

Parameter Name	Mand./Opt.	For-mat	Length	Data Model	Possible Values	Annotation
serviceCategoryIDs	Opt.	List of Strings	-	-	1, 2, 3	If a list of service category IDs is given, only the according service categories are returned.  Otherwise, i.e. if no service category ID is given, all service categories are be returned.
Locale	Opt.	String	5	-	Examples: “de_DE”, “de_AT”, “fr_CH”, “de_CH”, ...	If a locale (combination of language and country) is given, only the service category name available for this locale is returned.  Note here that if no name is available for the given locale, then the locale is mapped onto a locale for which a name is available, and this name is returned.  Otherwise, i.e. if no locale is given, the service category names for all available locales are returned.

Table 114: IF\_SOE.GetServiceCategories input

#### 4.2.3.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of Service Categories (each of the following parameters exist per service category)</b>						
serviceCategoryID	Mand.	Integer	-	ServiceCategory. ServiceCategoryID	1, 2, 3	The ServiceCategoryID identifies the service category the service is assigned to.
sortOrder	Mand.	Integer	-	ServiceCategory. SortOrder	-	The version of a service.
<b>• Inner List of service category names (each of the following entries exist per locale)</b>						
Locale	Mand.	String	5	-	Examples: de_DE, de_AT, fr_CH, de_CH, en, de, fr	Locale (combination of language and optionally country). Specifies the language of the service category name.
serviceCategoryName	Mand.	String	-	ServiceCategory.name		The localized service category name.

Table 115: IF\_SOE.GetServiceCategories output

#### 4.2.3.4 Exceptions

None.

### 4.2.4 IF\_SOE.GetServiceMasterData

#### 4.2.4.1 General Description

**Communication type:** Synchronously

This interface returns master data for services. If a list of service-IDs is given as input, the master data for the corresponding services is returned. If no service-ID is given, the master data of all services stored in SOE is returned.

This interface is implemented by AF.GetServiceMasterData.

Remark: Services that are currently not enabled in SOE are included in the output of this interface.

#### 4.2.4.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
serviceIDs	Opt.	List of String		Used as input parameter for "AF_GetServiceMasterData"	-	If a list of service IDs is given, only that list of services will be considered.  If no service ID is given, all services will be considered.

Table 116: IF\_SOE.GetServiceMasterData input

#### 4.2.4.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of Services (each of the following parameters exist per service)</b>						

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
serviceId	Mand.	Integer	-	Service.serviceID	-	The service id.
version	Mand.	Integer	-	Service.versionNumber	-	The version of a service.
serviceMasterId	Mand.	Integer	-	ServiceMaster.ServiceMasterID	-	The service master id.
activateAutomatically	Mand.	Boolean	-	ServiceMaster.activateAutomatically	-	This attribute determines whether the service should be activated automatically after event like: user agreement acceptance, vehicle registration
personalVerificationNeeded	Mand.	Boolean	-	ServiceMaster.personalVerificationNeeded	-	This attribute determines whether the customer has to be personally verified in order to use the service or not.
licenseRequired	Mand.	Boolean	-	Service.licenseRequired	-	This attribute determines whether a license is required in order to activate the service or if the service is available by default, e.g. eCall.
confirmationCodeNeeded	Mand.	Boolean	-	ServiceMaster.confirmationTokenNeeded	-	This attribute determines whether the customer has to enter a secret code in order to use the service or not.
trustLevel	Mand.	Integer		ServiceMaster.trustLevel	-	The attribute determines the minimal trust level of a vehicle registration in order to use the service.
technicalActivationPath	Mand.	Enum	-	Service.technicalActivationPath	"ACTIVATE_VIA_DAIVB", "ACTIVATE_VIA_	This attribute describes how the technical activa-

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
					FBS", "ACTIVATE_VIA_OMADM", "NO_ACTIVATION_NEED_ED"	tion of a service is done.
serviceCategoryID	Mand.	Integer		ServiceCategory.ServiceCategoryID	1, 2, 3	The ServiceCategoryID identifies the service category the service is assigned to.
• Inner List of service master names and service descriptions (each of the following entries exist per locale)						
Locale	Mand.	String	5	-	Examples: de_DE, de_AT, fr_CH, de_CH, en, de, fr	Locale (language in combination with or without a country). Specifies which language the service master name and the service description is supposed to be created in.
serviceName	Mand.	String	-	ServiceMaster.name		The localized service master name.
serviceDescription	Mand.	String	-	Service.description		The localized service description.
• Inner List of MBconnect countries, the service is available in						
countryCode	Mand.	String	-	MbcCountry.countryCode	Examples: DE, AT, CH	The country code of the MBconnect country the service is available in.

Table 117: IF\_SOE.GetServiceMasterData output

#### 4.2.4.4 Exceptions

None.

#### 4.2.5 IF\_SOE.GetServiceRelevantBodyTypesAndSalesTypes

**Communication type:** Synchronously

This interface returns lists of the body types and sales types available for the given locale, market, customers address country, consumer country and model series.

It is taken into account, whether the body types or sales types occur as constituents of service assignment rules.

**Note:** This interface serves to export potentially larger volumes of data. Consuming systems should consider caching mechanisms when retrieving data to ensure performance.

Internally, application function AF.GetServiceRelevantBodyTypesAndSalesTypes (see section 4.5.17) is called to get the body types and sales types.

#### 4.2.5.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Locale	Mand.	String	5	Used as input parameter for AF_GetModelSeries	Examples: "de_DE", "de_AT", "fr_CH", "de_CH", ...	Locale (either a language or a language in combination with a country). This field specifies in which locale the description fields of the returned master data are to be delivered.
MarketId	Mand.	String	9	Used as input parameter for AF_GetModelSeries	Example: "GS0008205"	The GEMS/GSSN outlet-ID of the MPC of the calling market. Supplying this parameter will exclude the model series and the sales types not valid for this market from the result list.
AddressCountryCode	Mand.	String	2		AT, DE, CH	The ISO code of the selected customer's address country.
ConsumerCountryCode	Mand.	String	2		AT, DE, CH	The ISO code of the consumer country.
ModelSeries	Mand.	String	3	ModelSeries.Mode ISeriesID	Example: "205"	A model series.

Table 118: IF\_SOE.GetServiceRelevantBodyTypesAndSalesTypes Input

#### 4.2.5.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>• A list of the body types:</b>						
BodyType	Mand.	String	256	BodyTypeDescription.description	Examples: "Cabriolet/Roadster", "Coupé", "Saloon", "Estate"	The body type.
<b>3. For each body type, a list of the according sales types:</b>						
Baumuster	Mand.	BaumusterDT	78	SalesType.Baumuster	Example: "205.004.1"	The baumuster of the sales type.
NST	Opt.	NstDT	3	SalesType.Nst	Example: "CH1"	The NSTet code of the sales type (for passenger cars, i.e. product group 'P').
SR1	Opt.	Sr1DT		SalesType.SR1	Example:	The SR1 of the sales type

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
					MG3+VP1+XZ1	(for transporters, i.e. product group 'T').
ModelNsrExtension	Opt.	ModelNsrExtensionDT		SalesType.ModelNsrExtension	Example: „BAD“	The ModelNsrExtension of the sales type (for transporters, i.e. product group 'T').
SalesTypeDescription	Mand.	String	256	SalesTypeDescription.Description (localized)	Example: "C220 BlueTEC"	The localized description of the sales type.

Table 119: IF\_SOE.GetServiceRelevantBodyTypesAndSalesTypes Output

#### 4.2.5.3 Exceptions

None.

### 4.2.6 IF\_SOE.GetServiceRelevantCodes

#### 4.2.6.1 General Description

**Communication type:** Synchronously

This interface returns lists of the year code combinations (each including a model year and an optional change year) and equipment codes available for the given locale, market, country, and sales type as defined by Baumuster and optionally NST code.

It is taken into account, whether the year code combinations and equipment codes occur as constituent of service assignment rules.

Note: This interface serves to export potentially larger volumes of data. Consuming systems should consider caching mechanisms when retrieving data to ensure performance.

Internally, application function AF.GetServiceRelevantYearAndEquipmentCodes (see section 4.5.19) is called to get the year code combinations and equipment codes.

#### 4.2.6.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Locale	Mand.	String	5	Used as input parameter for AF_GetModelSeries	Examples: "de_DE", "de_AT", "fr_CH", "de_CH", ...	Locale (either a language or a language in combination with a country). This field specifies in which locale the description fields of the returned master data are to be delivered.
MarketId	Mand.	String	9	Used as input parameter for AF_GetModelSeries	Example: "GS0008205"	The GEMS/GSSN outlet-ID of the MPC of the calling market. Supplying this parameter will exclude the model series not valid for

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						this market from the result list.
AddressCountryCode	Mand.	String	2		AT, DE, CH	The ISO code of the selected customer's address country.
ConsumerCountryCode	Mand.	String	2		AT, DE, CH	The ISO code of the consumer country.
Baumuster	Mand.	BaumusterDT	8	SalesType.Baumuster	Example: "205.004.1"	The baumuster of the sales type.
NST	Opt.	NstDT		SalesType.Nst	Example: „CH1“	The NST <sub>st</sub> code of the sales type (for passenger cars, i.e. product group 'P').
SR1	Opt.	Sr1DT		SalesType.SR1	Example: MG3+VP1+XZ1	The SR1 of the sales type (for transporters, i.e. product group 'T').
ModelNsrExtension	Opt.	ModelNsrExtensionDT		SalesType.ModelNsrExtension	Example: „BAD“	The ModelNsrExtension of the sales type (for transporters, i.e. product group 'T').

Table 120: IF\_SOE.GetServiceRelevantCodes Input

#### 4.2.6.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
• List of Mercedes connect me relevant year code combinations (model year + (opt.) change year)						
ModelYearCode	Mand.	String	5	YearCodeCombination.ModelYearCode	Example: "804"	The code of the model year.
ChangeYearCode	Opt.	String	5	YearCodeCombination.ChangeYearCode	Example: "054"	The code of the change year.
• For each year code combination (model year + (opt.) change year), a list of Mercedes connect me relevant equipment codes						
Code	Mand.	String	5	Equipment.Code	Example: "527"	The equipment code.
CodeDescription	Mand.	String	256	EquipmentDescription.Description (localized)	Example: "COMAND Online system with Media Interface"	The description of the equipment in the requested locale. If no translation is available, the value from the field Code is returned.

Table 121: IF\_SOE.GetServiceRelevantCodes Output

#### 4.2.6.4 Exceptions

None.

## 4.2.7 IF\_SOE.GetServiceRelevantModelSeries

**Communication type:** Synchronously

This interface returns a list of the model series available for the given market, customer's address country and consumer country of the vehicle.

It is taken into account, whether the model series occur as constituents of service assignment rules.

**Note:** This interface serves to export potentially larger volumes of data. Consuming systems should consider caching mechanisms when retrieving data to ensure performance.

Internally, application function AF\_GetServiceRelevantModelSeries is called to get the model series.

### 4.2.7.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
MarketId	Mand.	String	9	Used as input parameter for AF_GetModelSeries	Example: "GS0008205"	The GEMS/GSSN outlet-ID of the MPC of the calling market. Supplying this parameter will exclude the model series not valid for this market from the result list.
AddressCountryCode	Mand.	String	2		AT, DE, CH	The ISO code of the selected customer country.
ConsumerCountryCode	Mand.	String	2		AT, DE, CH	The ISO code of the consumer country.

Table 122: IF\_SOE.GetServiceRelevantModelSeries Input

### 4.2.7.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>• A list of the model series:</b>						
ModelSeries	Mand.	String	3	ModelSeries.ModelSeriesID	Example: „203“	The model series.

Table 123: IF\_SOE.GetServiceRelevantModelSeries Output

### 4.2.7.3 Exceptions

None.

## 4.2.8 IF\_SOE.GetServiceRightsByRole

### 4.2.8.1 General Description

**Communication type:** Synchronously

This interface returns the productive form of the authorization related master data for services and for a given list of roles. For a detailed description of the authorization roles and rights for services, see MBconnect role concept (→ see chapter 2.3.20).

Note: In case a list of roles is given, the interface will combine the rights for this list. If two or more roles have different rights regarding a specific service, the highest right for that service will be returned.

Internally, AF\_GetServiceRightsByRole is called.

### 4.2.8.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
roles	Mand.	List of Enumeration		ExternalEntityRoleEnumeration	-	The roles for which SOE has to return the associated rights.
serviceIDs	Opt.	List of String		Used as input parameter for "AF_GetServiceRightsByRole"	-	If a list of service IDs is given, only that list of services will be considered.  If no service ID is given, all services will be considered.

Table 124: IF\_SOE.GetServiceRightsByRole input

### 4.2.8.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of services (each of the following entries exist per service and resource)</b>						
serviceID	Mand.	String	-	Service.serviceID	-	-
right	Mand.	Enumeration	-	AccessRightAuthorizationAccessEnum	{NONE, READ, ACTIVATE, DEACTIVATE, WRITE}	Determines the right the given roles have for the given service. The highest right of each role will be returned for the list of roles.

Table 125: IF\_SOE.GetServiceRightsByRole output

### 4.2.8.4 Exceptions

If a serviceID is not found then exception SERMAN\_008 is thrown.

## 4.3 External View - Consumed Interfaces

### 4.3.1 IF\_CCM.GetServiceListRequest

**Communication type:** Synchronously

This interface calls CCM in order to retrieve all technically available Services for a vehicle with the given FIN or VIN.

#### 4.3.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
FinOrVin	Mand. ·	String	17		e.g. „WDD20433 1G122127“	The vehicle identifier number. May be a European FIN or the American VIN.

Table 126: IF\_CCM.GetServiceListRequest Input

#### 4.3.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
FinOrVin	Mand. ·	String	17	-	e.g. „ WDD204331 1G122127“	Search strings that the consumer sent in as "finOrVin". Needed for the interface consumer to match the results with his request.
Services	Mand. ·	List of Services		List < Service.ServiceID >	Examples: 1, 12356, 1234567890	IDs of the services which are available for the vehicle.

Table 127: IF\_CCM.GetServiceListRequest Output

#### 4.3.1.3 Exceptions

Code	Message
CCM_SYSTEM_ERROR	An "Internal System Error" occurred. Maybe something was changed on the vehicle.
CCM_VEHICLE_SYSTEM_ERROR	An "Internal System Error" occurred. Maybe something was changed on the vehicle.
VEHICLE_STOLEN	Vehicle is marked as "stolen". Service list is created anyway.
ERROR_VIN_FIN_CONFLICT	Vehicle could not be found in VeDoC.
VEHICLE_NOT_FOUND	Vehicle could not be found in VeDoC.

Table 128: Table of exceptions (IF\_CCM.GetServiceListRequest)

## 4.4 Internal View - Offered Interfaces

This chapter contains interfaces which are offered for other components within the SOE system (so called internal interfaces).

### 4.4.1 IIF\_DeleteServiceAssignmentRuleChangeLog

This internal interface deletes the ServiceAssignmentRuleChangeLog entry that corresponds to the given input data.

Internally calls AF\_DeleteServiceAssignmentRuleChangeLog (→ section 4.5.2).

---

#### **4.4.2 IIF\_DeleteServiceCountryAssignmentChangeLog**

This internal interface deletes the ServiceCountryAssignmentChangeLog entries that correspond to the given input data.

Internally, calls AF\_DeleteServiceCountryAssignmentChangeLog (→ section 4.5.4).

#### **4.4.3 IIF\_DeleteServiceMasters**

Internally, calls AF\_DeleteServiceMasters (→ see chapter 4.5.5) to delete the given ServiceMaster(s) if they are not used anymore by any service.

#### **4.4.4 IIF\_GetAllServices**

Internally, the **Error! Reference source not found.** is called.

#### **4.4.5 IIF\_GetAvailableServicesByConfiguration**

Internally, the **Error! Reference source not found.** is called.

#### **4.4.6 IIF\_GetMBcCountriesForServices**

This internal interface returns an aggregated list of unique MBcCountries, in which at least one of the given Services is available. Only enabled services in the given list of services are regarded.

Internally AF\_GetMBcCountriesForServices (→ section 4.5.8) will be called.

#### **4.4.7 IIF\_GetModelSeriesForService**

This internal interface returns each model series that is assigned to at least one of the given service assignment rules.

Internally calls AF\_GetModelSeriesForService (→ section 4.5.9).

#### **4.4.8 IIF\_GetServices**

Internally calls AF\_Getservices to retrieve all existing service entities.

#### **4.4.9 IIF.GetServiceAssignmentRules**

Internally calls AF.GetServiceAssignmentRules (→ see chapter 4.5.12) to retrieve all existing ServiceAssignmentRule entities.

#### **4.4.10 IIF.GetServiceAssignmentRulesForService**

This internal interface will retrieve the list of service assignment rules that cover the given service.

Internally calls AF.GetServiceAssignmentRulesForService (→ section 4.5.13).

---

#### **4.4.11 IIF\_GetServiceCategories**

Internally calls AF.GetServiceCategories to get necessary service categories.

#### **4.4.12 IIF\_GetServiceMasters**

Internally calls See external interface “IF\_SOE.GetServiceCategories”.  
AF.GetServiceMasters to retrieve all existing service master entities.

#### **4.4.13 IIF\_GetServiceMasterData**

Internally calls AF.GetServiceMasterData (→ section 4.5.16)

#### **4.4.14 IIF\_GetServiceRightsByRole**

Internally calls AF.GetServiceRightsByRole (→ see chapter 4.5.20) to retrieve the authorization related master data for given services and roles.

#### **4.4.15 IIF\_RetrieveCountryCodeForVehicleProductMasterData**

This internal interface will retrieve the list of countries that are affected by the changes brought to a ServiceAssignmentRule/Service.

Internally call AF\_RetrieveCountryCodeForVehicleProductMasterData (→ section 4.5.26).

#### **4.4.16 IIF\_RetrieveOldestServiceAssignmentRuleChangeLog**

This internal interface returns the oldest ServiceAssignmentRuleChangeLog entry. If there does not exist a ServiceAssignmentRuleChangeLog entry, the internal interface returns nothing.

Internally the internal interface calls  
AF\_RetrieveOldestServiceAssignmentRuleChangeLog (→ section 4.5.28).

#### **4.4.17 IIF\_RetrieveServiceCountryAssignmentChangeLog**

Based on the input parameter, this internal interface returns a list of ServiceCountryAssignmentChangeLog entries that either log new or deleted assignments between Services and MBcCountries.

If there does not exist any ServiceCountryAssignmentChangeLog entry with the given change operation, the internal interface returns an empty list.

Internally calls AF\_RetrieveServiceCountryAssignmentChangeLog (→ section 4.5.30).

#### **4.4.18 IIF\_SaveServiceDetail**

Internally calls to save the service information provided by the dialog DLG\_ServiceDetail.

---

#### **4.4.19 IIF\_SaveServiceMaster**

Internally calls AF\_SaveServiceMaster (→ see chapter 4.5.25) to save the given ServiceMaster.

#### **4.4.20 IIF\_UpdateServices**

Internally calls AF\_UpdateServices to update all service entities.

#### **4.4.21 IIF\_UpdateServiceAssignmentRules**

Internally calls AF\_UpdateServiceAssignmentRules (→ see chapter 4.5.33) to update all ServiceAssignmentRule entities.

#### **4.4.22 IIF\_UpdateServiceAssignmentRuleChangeLog**

Each time a service assignment rule has been newly created/deleted/changed and the appropriate change session has been released, this internal interface creates a new ServiceAssignmentRuleChangeLog entry.

Internally calls AF\_UpdateServiceAssignmentRuleChangeLog (→ section 4.5.34).

#### **4.4.23 IIF\_UpdateServiceCategories**

Internally calls AF\_UpdateServiceMasters to update all service master entities.

#### **4.4.24 IIF\_UpdateServiceCountryAssignmentChangeLog**

If the assignment between an enabled Service and a MBcCountry entity is added or deleted and the appropriate change session has been released, this application function creates or updates the corresponding ServiceCountryAssignmentChangeLog entry.

Internally calls AF\_UpdateServiceCountryAssignmentChangeLog (→ section 4.5.36).

#### **4.4.25 IIF\_UpdateServiceMasters**

Internally calls AF\_UpdateServiceMasters to update all service master entities.

### **4.5 Implementation**

#### **4.5.1 AF\_CheckVehicle**

##### **4.5.1.1 General Description**

See external interface **Error! Reference source not found..**

##### **4.5.1.2 Sequence Description**

**Step 1: Check if Mercedes connect me is supported in the customer's address country**

---

Call IIF\_IsMBconnectCountry with the requested MBcCountry (input parameter addressCountry).

If the result is true, continue. If the requested country is not supported by MBconnect, the user error “ACCDAS\_005” will be thrown and the application function quits.

*Remark: No check for MBcCountry, corresponding to input parameter oldAddressCountry required!*

## **Step 2: Determine the set of services supported by the current vehicle or vehicle configuration in the customer's address country**

- In the FIN-case, i.e. if <finOrVin> is given:

Call AF\_GetVehicleServiceInformationByFin with the given FIN or VIN - <finOrVin>, the business area “B2C”, the customer’s address country – <addressCountry>, and if given the locale - <locale> - in order to determine the vehicle’s MBconnect services, configuration including the FIN and VIN (if available), as well as the localized vehicle- and equipment descriptions.

Fill the list of supported services – list of <ServiceID>, the baumuster and corresponding baumuster description, the configuration of codes and the corresponding code descriptions - <Vehicle details>, as well as the vehicle’s vehicle type - <vehicleType> - with the results returned by AF\_GetVehicleServiceInformationByFin. The configuration of paint and upholstery including the descriptions is filled with the result returned by AF\_GetVehicleServiceInformationByFin, independently of the MBconnect relevance.

- In the configuration-case, i.e. if <baumuster> is given:

Call AF\_GetAvailableServicesByConfiguration with the given configuration, the business area “B2C”, the address country – <addressCountry>, and consumer country - <ConsumerCountry>, and fill the list of supported services – list of <ServiceID> - and the vehicle’s vehicle type - <vehicleType> - with the returned data.

If oldAddressCountry and/or an old vehicle (-configuration) is given as input, proceed with the steps described below (though the data determined in Step 1 is returned by this application function).

Otherwise the application quits and returns the data determined in Step 2.

## **Step 3: Determine the delta list of supported services**

1. Determine list of services additionally available/not available anymore

### Case 1: Old vehicle and old address country is given

With the given parameters of the old vehicle - <oldFinOrVin> -, the business area “B2C” and the address country corresponding to input parameter <oldAddressCountry>, call AF\_GetVehicleServiceInformationByFin (-> see chapter 3.2.4) if the old vehicle is given by FIN/VIN. If the old vehicle is given by a configuration, call AF\_GetAvailableServicesByConfiguration with the customer’s old address country - <oldAddressCountry>, as well as the vehicle’s configuration, the business area “B2C” and consumer country - <ConsumerCountry>. Only the returned list of services is extracted from the output data.

---

#### Case 2: Old vehicle and no old address country is given

With the FIN or VIN of the old vehicle - <oldFinOrVin> -, the business area “B2C” and the address country corresponding to input parameter <addressCountry>, call AF\_GetVehicleServiceInformationByFin if the old vehicle is given by FIN/VIN. If the old vehicle is given by a configuration, call AF\_GetAvailableServicesByConfiguration with the customer’s address country - <addressCountry>, as well as the vehicle’s configuration and consumer country - <ConsumerCountry> - as input. Only the returned list of services is extracted from the output data.

#### Case 3: No old vehicle, but old address country is given

With the FIN or VIN of the vehicle - <finOrVin> - (not the old one) and the address country corresponding to input parameter <oldAddressCountry>, call **Error! Reference source not found.** (-> see chapter **Error! Reference source not found.**) if the vehicle is given by FIN/VIN. If the old vehicle is given by a configuration, call AF\_GetAvailableServicesByConfiguration (-> see chapter 4.5.7) with the customer’s old address country - <oldAddressCountry>, as well as the vehicle’s configuration, the business area “B2C” and consumer country - <ConsumerCountry> - as input. Only the returned list of services is extracted from the output data.

## 2. Determine list of services additionally available/not available anymore

Compare the list of services, supported by the requested vehicle (determined in Step 2) with the list of services, supported by the old vehicle/and or in the old address country (determined in Step 3.1) by comparing the serviceIDs:

- The output list of additionally supported services is filled with these services that are in the list of services determined in Step 1 but not in the list determined in Step 3.1.
- The output list of removed services is filled with these services that are NOT in the list of services determined in Step 1 but are in the list determined in Step 3.1.
- For services that are in the list of services determined in Step 1 and in the list determined in Step 3.1, do nothing.

### **4.5.1.3 Input**

See external Interface “IF\_SOE\_CheckVehicle”.

### **4.5.1.4 Output**

See external Interface “IF\_SOE\_CheckVehicle”.

### **4.5.1.5 Exceptions**

See external Interface “IF\_SOE\_CheckVehicle”.

## **4.5.2 AF\_DeleteServiceAssignmentRuleChangeLog**

This application function deletes the ServiceAssignmentRuleChangeLog entity that corresponds to the given input data.

### **4.5.2.1 Sequence Description**

Delete the ServiceAssignmentRuleChangeLog entity that corresponds to the given input parameter.

### **4.5.2.2 Input**

Name	Type / Length / BOM	Description
ServiceAssignmentRuleChangeLog	ServiceAssignmentRuleChangeLog	The ServiceAssignmentRuleChangeLog to be deleted

Table 129: AF\_DeleteServiceAssignmentRuleChangeLog Input

### **4.5.2.3 Output**

None.

### **4.5.2.4 Exceptions**

None.

## **4.5.3 AF\_DeleteServiceCategories**

This AF deletes the given list of service categories in a change session, if they are not used by any service.

### **4.5.3.1 Sequence Description**

#### **Check change sessions for given objects (service categories)**

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown, execution stops here.
- Verify, that none of the objects is inside a change session of another user. If it is, throw error VEHPROSESSION\_001. Execution stops here.

#### **Validate objects and references**

- If at least one service category is referenced by one or more services (service masters), abort the transaction for all given service categories with the error message SERMAN\_021. Use <ServiceCategory>.ServiceCategoryID as placeholder 1. As placeholder 2 use a list of the services (service masters) that reference the ServiceCategory.

#### **Mark objects as deleted**

- Bring the service categories into the change session of the user (see general algorithm "ApplyChangeSessionOnElement").
- Mark the service categories as deleted (ChangeOperation.DELETE).

#### **4.5.3.2 Input**

Name	Type / Length / BOM	Description
serviceCategories	List of ServiceCategory	The service categories to be deleted.

Table 130: AF\_DeleteServiceCategories Input

#### **4.5.3.3 Output**

None.

#### **4.5.3.4 Exceptions**

If one of the service categories is already being edited inside the change session of another user, then throw error SRVSESSION\_003.

If the user does not have an opened change session, then throw error SESSION\_001.

If one of the service categories to be deleted is referenced by one or more services (service masters), then throw error SERMAN\_021.

### **4.5.4 AF\_DeleteServiceCountryAssignmentChangeLog**

This application function deletes each given ServiceCountryAssignmentChangeLog entity that corresponds to the given input data.

#### **4.5.4.1 Sequence Description**

Delete the ServiceCountryAssignmentChangeLog entities that correspond to the given input parameter.

#### **4.5.4.2 Input**

Name	Type / Length / BOM	Description
ServiceCountryAssignmentChangeLogList	(Can be empty) List of <ServiceCountryAssignmentChangeLog>	The list of ServiceCountryAssignmentChangeLog entities that need to be deleted.

Table 131: AF\_DeleteServiceCountryAssignmentChangeLog Input

#### **4.5.4.3 Output**

None.

#### **4.5.4.4 Exceptions**

None.

### **4.5.5 AF\_DeleteServiceMasters**

#### **4.5.5.1 General Description**

This AF deletes the given service masters if they are not used anymore by any service.

#### **4.5.5.2 Sequence Description**

Loop for all given service masters:

If at least one service master is still being referenced by one or more service abort with the error message SERMAN\_020:

- Use ServiceMaster.Name as placeholder 1.
- Optional: As placeholder 2 use a list of all Service.ServiceID which reference to the ServiceMaster.

Else delete the given ServiceMaster(s).

#### **4.5.5.3 Input**

Name	Type / Length / BOM	Description
ServiceMasters	List of <ServiceMaster>	The service masters to be deleted.

Table 132: AF\_DeleteServiceMasters input

#### **4.5.5.4 Output**

None.

#### **4.5.5.5 Exceptions**

Code	Message
SERMAN_020	The service master <1> could not be deleted because it is still being referenced by one or more services. Please go to the respective maintenance screen and remove the reference first. The service(s) with the following ID(s) have been identified as referencing to this item: <2>.

Table 133: Table of exceptions (AF\_DeleteServiceMasters)

### **4.5.6 AF\_GetAllServices**

Based on the input parameter, this application function provides

- either all existing Service entities or
- all enabled Service entities existing in SOE or
- all services available in the given country or
- all services that are enabled and available in the given country.

#### **4.5.6.1 Sequence Description**

- If the given input parameter onlyEnabledServices=false:

Determine all instances of <Service> that are available in SOE and return the data. In case that a business area other than "ALL" is given as input parameter, return the services where the ServiceMaster.BusinessArea of the corresponding service master matches the given business area. In case that a country is given as input parameter return the list of services available in SOE and available in that country.

- If the given input parameter onlyEnabledServices=true:

Determine all instances of <Service> that are available in SOE and for which <Service>.<enabledFrom> is given and is smaller or equal than today and <Service>.<enabledTo> is empty or bigger than today.

---

In case a business area other than “ALL” is given as input parameter, return the services where the ServiceMaster.BusinessArea of the corresponding service master matches the given business area.

If no country is given as input, return the data. If a country is given as input parameter, only return these Service entities that are available in that country.

#### 4.5.6.2 Input

Name	Type / Length / BOM	Description
onlyEnabledServices	Boolean	States if the internal interface returns each existing service (=false) or only services that are currently enabled (=true).
BusinessArea	Enum	The business area of the services, one of “B2C”, “B2B” or “ALL”. Only services from the given business area will be returned.
CountryCode	MbcCountry.countryCode	Optional: ISO Country Codes is used as a filter parameter in order to determine the list of services available in that country

Table 134 AF\_GetAllServices Input

#### 4.5.6.3 Output

Name	Type / Length / BOM	Description
List<Service> : List of either all Service entities or all enabled Service entities from the given business area existing in SOE and available in the country – if given		
Service	Service	Service entity.

Table 135: AF\_GetAllServices Output

#### 4.5.6.4 Exceptions

None.

### 4.5.7 AF\_GetAvailableServicesByConfiguration

#### 4.5.7.1 General Description

This AF returns all enabled Services which are configured to be available for the given configuration built for the requested consumer country and available in the customers address country. It also returns a list of distinct matched equipment codes belonging to the service rules. If no matching rule can be found, an empty list is returned (=no error). The calling AF has to decide, whether this is interpreted as an error or not.

#### 4.5.7.2 Sequence Description

Call AF\_RetrieveServicesAndMatchedRules the given VehicleConfiguration, BusinessArea, AddressCountry and ConsumerCountry and with productiveStatus=“TRUE” and evaluationDate=“TODAY”.

---

Return the lists of services and of distinct equipment codes contained in the rules returned by AF\_RetrieveServicesAndMatchedRules as output parameters <Services> and <MatchedCodes>.

Propagate the vehicle's vehicle type returned by AF\_RetrieveServicesAndMatchedRules to the output parameter <vehicleType>.

Note: The returned list of matched codes can be empty.

#### 4.5.7.3 Input

Name	Type / Length / BOM	Description
VehicleConfiguration	VehicleConfiguration	The configuration to find available services for.
BusinessArea	Enum	The business area, one of "B2C", "B2C" or "ALL". Only services from the given business area will be returned.
addressCountry (Opt.)	MbcCountry.countryCode	The address country of the customer. If the parameter is given, only B2C services that are available in this country will be returned. If the parameter is not given, the B2C services will not be filtered, i.e. all B2C services will be returned.  B2B services will never be filtered, i.e. all B2B services will be returned.
consumerCountry (Opt.)	MbcCountry.countryCode	(Optional) If given: Filters for the services available for the consumer country If not given: the consumer country will be ignored when computing the available services

Table 136: AF\_GetAvailableServicesByConfiguration Input

#### 4.5.7.4 Output

Name	Type / Length / BOM	Description
Services	List of Services	The available services.
MatchedCodes	List of Strings	The list of equipment codes which made the vehicle match the rule.
vehicleType	VehicleTypeEnum	Indicates the type of service the given vehicle or vehicle configuration supports. Mercedes connect me-vehicles have a communication module. Adapter-vehicles do not have a communication module but an OBD II-adapter interface. Vehicles that neither have a communication module nor an OBD II-adapter interface do not support any services integrated in Mercedes connect me.

Table 137: AF\_GetAvailableServiceByConfiguration Output

#### 4.5.7.5 Exceptions

None.

### 4.5.8 AF\_GetMBcCountriesForServices

This application returns an aggregated list of unique MBcCountries, in which at least one of the given services is available.

#### 4.5.8.1 Sequence Description

1. If the input parameter *Enabled* is given, filter the list of services given as input by their activation status (true or false).

- 
2. For each of the filtered services, determine the MBcCountries, the service is assigned to.
  3. Aggregate the lists of MBcCountries to a list of unique MBcCountries and return the data.

#### **4.5.8.2 Input**

Name	Type / Length / BOM	Description
ServiceList	List of <Service>	The list of services for which the MBcCountries need to be determined.
Enabled	Boolean	Optional

Table 138: AF\_GetMBcCountriesForServices Input

#### **4.5.8.3 Output**

Name	Type / Length / BOM	Description
MbcCountryList	List of MbcCountry	The MBcCountries, at least one of the given services are available in.

Table 139: AF\_GetMBcCountriesForServices Output

#### **4.5.8.4 Exceptions**

None.

### **4.5.9 AF\_GetModelSeriesForService**

This application returns an aggregated list of unique ModelSeries that are assigned to a ServiceAssignmentRule, that covers the given service.

#### **4.5.9.1 Sequence Description**

Determine each ServiceAssignmentRule that is assigned to the given ServiceID by calling AF.GetServiceAssignmentRulesForService.

For each returned ServiceAssignmentRule remember each related ModelSeries.

Aggregate the remembered ModelSeries to a list of unique ModelSeries and return the data.

#### **4.5.9.2 Input**

BO-Name	Type / Length / BOM	Description
Service	Service.ServiceID	The ID of the service the list of modelSeries are requested for.

Table 140: AF\_GetModelSeriesForService Input

#### **4.5.9.3 Output**

BO-Name	Type / Length / BOM	Description
ModelSeriesList	List of <ModelSeries>	List of model series assigned a service assignment rules, that covers the given service..

Table 141: AF\_GetModelSeriesForService Output

---

#### **4.5.9.4 Exceptions**

None.

### **4.5.10 AF\_GetRequiredFieldsUsedByServices**

#### **4.5.10.1 General Description**

This AF returns the master data of the required fields that are necessary to activate a particular service. The returned list covers account as well as vehicle related data fields.

#### **4.5.10.2 Sequence Description**

##### ***Step 1: Retrieve Structural Information On Profile Data Fields and Groups:***

Call IIF\_GetProfileDataFieldsAndGroups in order to get the information on profile data fields and groups needed in the following steps.

For each of the given services, do the following steps:

##### ***Step 2.1: Retrieve the Service Master***

Get the assigned ServiceMaster for each given Service.

##### ***Step 2.2: Retrieve Vehicle Related Required Profile Fields:***

For each entry in <ServiceMaster>.<requiredVehicleInformation>, where the item ID matches the datafieldID from a data field retrieved in step 1, add the required field to the output list of required items with relationshipType = DATAFIELD, itemID = fieldID, fieldOwnerType = fieldOwnerType from the response from step 1 and – if the field has dependentMandatoryFields – its child IDs.

For each entry in <ServiceMaster>.<requiredVehicleInformation>, where the item ID matches a groupID from a group retrieved in step 1, add the required field to the output list of required items with relationshipType = GROUP, itemID = groupID, fieldOwnerType = VEHICLE and – if the group has childrens – its child IDs.

##### ***Step 2.3: Retrieve Profile Data Fields:***

For each entry in <ServiceMaster>.<requiredCustomerInformation>, where the item ID matches the datafieldID from a data field retrieved in step 1, add the required field to the output list of required items with relationshipType = DATAFIELD, itemID = fieldID, fieldOwnerType = fieldOwnerType from the response from step 1 and – if the field has dependentMandatoryFields – its child IDs.

For each entry in <ServiceMaster>.<requiredCustomerInformation>, where the item ID matches a groupID from a group retrieved in step 1, add the required field to the output list of required items with relationshipType = GROUP, itemID = groupID, fieldOwnerType = ACCOUNT – if the group has childrens – its child IDs.

##### ***Step 2.4: Retrieve Custom Properties:***

Take all custom properties of the service master and add them as a list.

#### **4.5.10.3 Input**

See external interface “IF\_SOE\_GetRequiredFieldsUsedByServices”.

#### **4.5.10.4 Output**

See external interface “IF\_SOE\_GetRequiredFieldsUsedByServices”.

#### **4.5.10.5 Exceptions**

See external interface “IF\_SOE\_GetRequiredFieldsUsedByServices”.

### **4.5.11 AF\_GetServices**

#### **4.5.11.1 General Description**

This application function provides all existing Service entities.

#### **4.5.11.2 Sequence Description**

Determine all instances of entity <Service> and information from related entities that are available in SOE and return the data.

#### **4.5.11.3 Input**

None.

#### **4.5.11.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;Service&gt;: List of all Service entities</b>		
serviceID	Service.serviceID	The ID of a Service.
partNumber	Service.partNumber	The part number of a Service (“Sachnummer”)
versionNumber	Service.versionNumber	The version of a Service.
licenseRequired	Service.licenseRequired	Tells whether a license is required or not in order to use this service.
contractDuration	Service.contractDuration	<u>Optional:</u> Describes the contract duration in months, in case a contract for this service is established. This attribute is optional.
contractStartTrigger	Service.contractStartTrigger	Indicates what event triggers the begin of a contract related to this service.
enabledFrom	Service.enabledFrom	The start date of a service (including this day). From this date the service is available of the sales' point of view.
enabledTo	Service.enabledTo	The last date of a service (including this day). From this date the service is not available of the sales' point of view.
technicalActivationPath	Service.technicalActivationPath	Determines the technical activation path, i.e. the system, that has to be contacted in order to activate or deactivate the service in the vehicle.
serviceClass	Service.serviceClass	Mercedes connect me-services are available for Mercedes connect me-vehicles exclusively. Mercedes connect me-vehicles are vehicles that have a communication module. Adapter-services are available for adapter vehicles exclusively. Adapter-vehicles are vehicles that have an OBD II adapter-interface but no communication module.
serviceMasterID	ServiceMaster.serviceMasterID	The ID of the ServiceMaster.
- Inner List of <String(4096)>		

Parameter Name	Type / Length / BOM	Description
description	Service.description	Translation of a service's description (i18n)
- Inner List of <MbCountry>		
countryCode	MbcCountry.countryCode	The CountryCode of a Country.

Table 142: AF\_GetServices output

#### 4.5.11.5 Exceptions

None.

### 4.5.12 AF.GetServiceAssignmentRules

#### 4.5.12.1 General Description

This application function provides all existing ServiceAssignmentRule entities.

#### 4.5.12.2 Sequence Description

Determine all instances of entity <ServiceAssignmentRule> and information from related entities that are available in SOE and return the data.

#### 4.5.12.3 Input

None.

#### 4.5.12.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;ServiceAssignmentRule&gt;: List of all ServiceAssignmentRule entities</b>		
serviceAssignmentRuleId	ServiceAssignmentRule.serviceAssignmentRuleId	The ID of a ServiceAssignment Rule.
Description	ServiceAssignmentRule.description	A name for a rule. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
longDescription	ServiceAssignmentRule.longDescription	A comment on the rule that explains or summarizes the rule in greater detail. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
modelSeriesID	ModelSeries.modelSeriesID	The ID of a ModelSeries.
- Inner List of <YearCodeCombination>		
ModelYearCode	ModelYear.code	The Code of a ModelYear
ChangeYearCode	String	The Code of a ChangeYear
- Inner List of <AndTerm>		
AndTermOrder	AndTerm.displayOrder	The sequence number of OR terms within Boolean expression. Used to restore the OR terms in the sequence originally created.
o Inner List of <OrTerm>		
Code	OrTerm.code Equipment.code	The Code of an Equipment
isNegated	OrTerm.negated	Boolean flag that indicates whether the equipment code is negated within Boolean expression
OrTermOrder	OrTerm.displayOrder	The sequence number of equipment code within OR term of Boolean expression. Used to restore equipment codes in the OR term in the sequence originally created.
- SalesTypeCondition		
baumusterWild-card	SalesTypeCondition.baumusterWildcard	Search condition for a matching Baumuster. This pattern may contain wildcards.

Parameter Name	Type / Length / BOM	Description
○ Inner List of <NST>		
NST	SalesTypeCondition.NST	Search condition for a National Sales Type.
useConditionBaumuster-Wildcards	SalesTypeCondition.useConditionBaumusterWildcards	Boolean flag that indicates if the search condition under BaumusterWildcard shall be used during a search (TRUE) or ignored (FALSE).
useConditionNst	SalesTypeCondition.useConditionNst	Boolean flag that indicates if the search condition under Nst shall be used during a search (TRUE) or ignored (FALSE).
- ConsumerCountryCondition		
○ Inner List of <MbcCountry>		
countryCode	MbcCountry.countryCode	The CountryCode of a Country.
useCondition	ConsumerCountryCondition.useCondition	Tells whether the country condition shall be taken in account (true) or not (false) when the rule is evaluated.
- Inner List of <Service>		
ServiceID	Service.serviceID	The ID of a Service.

Table 143: AF\_GetServiceAssignmentRules output

#### 4.5.12.5 Exceptions

None.

### 4.5.13 AF.GetServiceAssignmentRulesForService

This application function will retrieve the list of service assignment rules that cover the given service.

#### 4.5.13.1 Sequence Description

Return each ServiceAssignmentRule that is assigned to the Service with the given ServiceID.

#### 4.5.13.2 Input

BO-Name	Type / Length / BOM	Description
Service	Service.ServiceID	The ID of the service the service assignment rule is requested for.

Table 144: AF\_GetServiceAssignmentRulesForService Input

#### 4.5.13.3 Output

BO-Name	Type / Length / BOM	Description
ServiceAssignmentRuleList	List of <ServiceAssignmentRule>	List of service assignment rules that cover the given service.

Table 145: AF\_GetServiceAssignmentRulesForService Output

#### 4.5.13.4 Exceptions

None.

---

## **4.5.14 AF\_GetServiceCategories**

### **4.5.14.1 General Description**

See external interface “IF\_SOE.GetServiceCategories”.

### **4.5.14.2 Sequence Description**

- If a list of service category IDs is given, determine the according service category. Otherwise determine all service categories available in SOE.
- If a locale is given, determine the names for the (one or more) service categories available for this locale. Here, if no name is available for the given category and locale, then map the locale onto a locale for which a name is available (fallback logics like for documents) and take this name. Otherwise, i.e. if no locale is given, take the service category names for all available locales.
- Return the (one or more) service categories and their localized (one or more) names.

### **4.5.14.3 Input**

See external interface “IF\_SOE.GetServiceCategories”.

### **4.5.14.4 Output**

See external interface “IF\_SOE.GetServiceCategories”.

### **4.5.14.5 Exceptions**

See external interface “IF\_SOE.GetServiceCategories”.

---

## **4.5.15 AF\_GetServiceMasters**

### **4.5.15.1 General Description**

This application function provides all existing ServiceMaster entities.

### **4.5.15.2 Sequence Description**

Determine all instances of entity <ServiceMaster> and information from related entities that are available in SOE and return the data.

### **4.5.15.3 Input**

None.

### **4.5.15.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;ServiceMaster&gt;: List of all ServiceMaster entities</b>		
serviceMasterId	ServiceMaster.serviceMasterId	The ID of the ServiceMaster.
serviceCategoryId	ServiceCategory.ServiceCategoryID	The ServiceCategoryID identifies the service category the service is assigned to.
activateAutomatically	ServiceMaster.activateAutomatically	States if service(s) assigned to the service master is/are activated automatically when the vehicle is registered to a customer.

Parameter Name	Type / Length / BOM	Description
confirmationTokenNeeded	ServiceMaster.confirmationTokenNeeded	If "TRUE" service(s) assigned to the service master need(s) a confirmed connection between the user and a vehicle through a verification token in. Such service(s) must be activated only after a successful verification.
personalVerificationNeeded	ServiceMaster.personalVerificationNeeded	If "TRUE" a retailer has to verify the identity of the customer in order to use service(s) assigned to the service master.
trustLevel	ServiceMaster.trustLevel	Determines the minimal trust level of a vehicle registration in order to use service(s) assigned to the service master.
- Inner List of <String(256)>		
name	ServiceMaster.name	Translation of a service master name (i18n)
- Inner List of custom properties		
customProperty	Service.customProperties	List of the custom properties required for the activation of service(s) assigned to the service master.
- Inner List of <ProfileDataFieldItem>		
requiredCustomerInformation	ServiceMaster.requiredCustomerInformation.fieldID	List with the mandatory fields for service(s) assigned to the service master that relate to attributes of the customer profile.
requiredVehicleInformation	ServiceMaster.requiredVehicleInformation.fieldID	List with the mandatory fields for service(s) assigned to the service master that relate to attributes of a vehicle.
- Inner List of <ProfileDataFieldGroup>		
requiredCustomerInformation	ServiceMaster.requiredCustomerInformation.groupID	List with the mandatory fields for service(s) assigned to the service master that relate to attributes of the customer profile.
requiredVehicleInformation	ServiceMaster.requiredVehicleInformation.groupID	List with the mandatory fields for service(s) assigned to the service master that relate to attributes of a vehicle.

Table 146: AF\_GetServiceMasters output

#### 4.5.15.5 Exceptions

None.

### 4.5.16 AF\_GetServiceMasterData

#### 4.5.16.1 General Description

This AF provides access to the service related master data.

If there is no list of services given as input parameter, the answer covers all services stored in SOE.

*Hint: Additionally services that are currently not enabled in SOE are returned by this application function.*

#### 4.5.16.2 Sequence Description

- If no service ID is given, consider all instances of <Service>.
- If a list of service IDs is given, consider those Services where <Service>.serviceID match the given service ID.

For all relevant instances of <Service>, return <Service>.serviceID, <Service>.versionNumber, <ServiceMaster>.ServiceMasterID, <ServiceMaster>.personalVerificationNeeded, <ServiceMaster>.confirmationTokenNeeded

---

(mapped to outputparameter <confirmationCodeNeeded>), <ServiceMaster>.<trustLevel>, <Service>.<technicalActivationPath>, <Service>.<licenseRequired> and <ServiceMaster>.<activateAutomatically> as an aggregated list.

For each service

- determine all available translations, and also attach the list of 5-digit locales (for example “de\_DE”) and localized service master name and service description to the service.
- additionally attach each locale, that is maintained in SOE as fallback, as 2-digit locale (only the language abbreviation, for example “de”) with the corresponding localized service master name and service description.
- determine all <MbcCountry> entities the service is assigned to and return the list of <MbcCountry>.<countryCode>.

Hint:

- The attribute <ServiceMaster>.<confirmationCodeNeeded> is mapped with <ServiceMaster>.<confirmationTokenNeeded> in the database.
- The service enablement dates (<Service>.<enabledFrom> and <Service>.<enabledTo>) are not returned by this application function.

#### **4.5.16.3 Input**

See internal interface “IF\_SOE.GetServiceMasterData”.

#### **4.5.16.4 Output**

See internal interface “IF\_SOE.GetServiceMasterData”.

#### **4.5.16.5 Exception**

See internal interface “IF\_SOE.GetServiceMasterData”.

### **4.5.17 AF\_GetServiceRelevantBodyTypesAndSalesTypes**

This application function returns lists of the body types and sales types available for the given locale, market, country, and model series.

It is taken into account, whether the body types or sales types occur as constituents of service assignment rules.

#### **4.5.17.1 Sequence Description**

Check, if the given model series is valid for the given market and not out of production by using IIF\_FilterAvailableModelSeries.

Select all service assignment rules that meet the following criteria:

1. The model series defined in the rule is matched by the given model series.
2. The rule contains a service that is
  - a.enabled on the given day (Service.enabledFrom is smaller than or equals Today, and Service.enabledTo is empty or greater than Today),
  - b.and available for the given country  
(ServiceAssignment.Service.MBcCountry.countryCode = CountryCode).

---

Call IIF\_GetSalesTypesForModelSeriesId with the given model series' ID as input. Select the distinct sales types that match one of the selected rules.

Filter out sales types which are not valid for the market given as input.

Group the sales types by their assignment to body types.

Use the translations for the descriptions of body types and sales types according to the given locale.

Return the lists of body types and sales types.

#### **4.5.17.2 Input**

See external interface IF\_SOE.GetServiceRelevantBodyTypesAndSalesTypes.

#### **4.5.17.3 Output**

See external interface IF\_SOE.GetServiceRelevantBodyTypesAndSalesTypes.

#### **4.5.17.4 Exceptions**

See external interface IF\_SOE.GetServiceRelevantBodyTypesAndSalesTypes.

### **4.5.18 AF\_GetServiceRelevantModelSeries**

This application function returns a list of the model series available for the given market, and country.

It is taken into account, whether the model series occur as constituents of service assignment rules.

#### **4.5.18.1 Sequence Description**

Select all service assignment rules that meet the following criteria:

- The rule contains a service that is
  - enabled on the given day (Service.enabledFrom is smaller than or equals Today, and Service.enabledTo is empty or greater than Today),
  - and available for the given:
    - customer address country  
(ServiceAssignmentRule.Service.MBcCountry.countryCode = AddressCountryCode).
    - given consumer country: the consumer country is found in the list of the MBcCountry associated to the ServiceAssignmentRule.ConsumerCountryCondition.  
In case the consumer country is not given, then skip this condition

Get the distinct model series contained in the selected rules.

Filter these model series to obtain the model series that are available for the given market and not out of production by using IIF\_FilterAvailableModelSeries.

Return this list of model series.

---

#### **4.5.18.2 Input**

See external interface IF\_SOE.GetServiceRelevantModelSeries.

#### **4.5.18.3 Output**

See external interface IF\_SOE.GetServiceRelevantModelSeries.

#### **4.5.18.4 Exceptions**

See external interface IF\_SOE.GetServiceRelevantModelSeries.

---

### **4.5.19 AF\_GetServiceRelevantModelYearsAndEquipmentCodes**

#### **4.5.19.1 General Description**

This application function returns lists of the year code combinations and equipment codes available for the given locale, market, country, and sales type as defined by Baumuster and optionally NST code.

It is taken into account, whether the sales type occurs as constituent of service assignment rules.

#### **4.5.19.2 Sequence Description**

Determine the model series of the given sales type by calling IIF\_ResolveModelSeriesForSalesType with SalesType.baumuster.

Check, if the model series is valid for the given market and not out of production by using IIF\_FilterAvailableModelSeries.

#### **Selection of relevant service assignment rules**

Select all service assignment rules that meet the following criteria:

- The model series from the rule is matched by the model series of the given sales type.
- If the rule contains a Sales Type Condition and UseConditionBaumusterWildcards is set: The Sales Type Condition is matched by the given sales type. This is the case, if the Baumuster wildcard is matched by the given Baumuster and, if useConditionNst is set, the NST list of the rule contains the given NST.
- The <consumer country> is found in the list of the MBcCountry associated to the ServiceAssignmentRule.ConsumerCountryCondition.  
In case the <consumer country> is not given, then skip this condition (=the condition doesn't apply)
- The rule contains a service that is
  - enabled on the given day (Service.enabledFrom is smaller than or equals Today, and Service.enabledTo is empty or greater than Today),
  - and available for the given address country (ServiceAssignment.Service.MBcCountry.countryCode = CountryCode).

#### **Selection of relevant year codes and equipment codes**

---

Get the distinct year code combinations (including model year and an optional change year) contained in these rules.

For each of these year code combinations, check the rules that meet the criteria specified above and select all rules that additionally meet the following criterion:

- The rule contains the given year code combination.

For each of these year code combinations, collect the distinct equipment codes contained in these rules, i.e. all ServiceAssignmentRule.AndTerm.OrTerm.code.

**Example:**

**Selection of relevant service assignment rules**

The following example illustrates the steps outlined above. Suppose IIF\_ResolveModelSeriesForSalesType determined, that the value of model series is 205 and IIF\_FilterAvailableModelSeries confirms that it is valid for the given market. By comparing model series values with the according values from the rules in the Table below, one finds that the four rules with the IDs 6028, 6029, 6032 and 6033 are matched.

Rule ID	Model Series	Sales Types (optional)	Year Code Combinations (Model Year + (opt.) Change Year)	Equipment Code Condition	Consumer Countries (optional)	Services
6028	205		805, 806, 807	06U and B54	„DE, FR, GB, ... (ECE15)“	1
6029	205		805, 806, 807	350 and not 05U	„DE, FR, GB, ... (ECE15)“	2
6032	205		805, 806, 807	05U and ( 228 or 230 )	„DE, FR, GB, ... (ECE15)“	22
6033	205		805	B57 and (06U or 05U) and 350 and not 228	„DE, FR, GB, ... (ECE15)“	23
6035	207		806, 807, 808	06U and B54	„DE, FR, GB, ... (ECE15)“	1
6036	207		806, 807, 808	not 350	„DE, FR, GB, ... (ECE15)“	2
6039	222		806, 807, 808	348	CN	35,36
6040	222		806, 807, 808	348	US	45,46

Table 147: Exemplary selection of service assignment rules

The result of step “Selection of relevant service assignment rules” is shown in Table

Rule ID	Model Series	Sales Types (optional)	Year Code Combinations (Model Year + (opt.) Change Year)	Equipment Code Condition	Consumer Countries (optional)	Services

6028	205		805, 806, 807	06U and B54	„DE, FR, GB, ... (ECE15)“	1
6029	205		805, 806, 807	350 and not 05U	„DE, FR, GB, ... (ECE15)“	2
6032	205		805, 806, 807	05U and ( 228 or 230 )	„DE, FR, GB, ... (ECE15)“	22
6033	205		805	B57 and (06U or 05U) and 350 and not 228	„DE, FR, GB, ... (ECE15)“	23

Table 148: Selected service assignment rules

### Selection of relevant year codes and equipment codes

After retrieving distinct year code combinations and distinct equipment codes contained in these four rules for each of these year code combinations, the following result is delivered:

Year code	Equipment code
805	06U, B54, 350, 05U, 228, 230, B57
806	06U, B54, 350, 05U, 228, 230
807	06U, B54, 350, 05U, 228, 230

Note: For the output of this application function only year code combination and equipment codes are relevant. The negation flag of equipment code (ServiceAssignmentRule.AndTerm.OrTerm.negated) is to ignore.

Use the translations for the descriptions of equipment codes according to the given locale.

Put the equipment codes, and their localized descriptions into a list, and return this list and the list of year code combinations (including model year and an optional change year).

#### 4.5.19.3 Input

See external interface IF\_SOE.GetServiceRelevantModelYearsAndEquipmentCodes.

#### 4.5.19.4 Output

See external interface IF\_SOE.GetServiceRelevantModelYearsAndEquipmentCodes.

#### 4.5.19.5 Exceptions

See external interface IF\_SOE.GetServiceRelevantModelYearsAndEquipmentCodes.

---

## 4.5.20 AF.GetServiceRightsByRole

### 4.5.20.1 General Description

This application function returns the authorization related master data for the given services and roles. For a detailed description of the authorization roles and rights for services, see MBconnect role concept (→ see chapter **2.3.20**).

### 4.5.20.2 Sequence Description

#### **Step 1: Retrieve Service Master(s)**

If no service ID is given, consider all instances of <ServiceMaster>.

If a list of serviceIDs is given, get the assigned <ServiceMaster> of each given serviceID.

#### **Step 2: Retrieve the right(s)**

For each service master determined in Step 1, call

**IIF\_GetRightsByServiceMasterAndRole** (→ see chapter **14.4.5**) with the serviceMasterID and the rules given as input parameter of this AF and remember the retrieved serviceMaster and retrieved right.

#### **Step 3: Return the service(s) and the associated right(s)**

If no service ID was given, return the service(s) assigned to the service master and the associated right remembered in Step 2.

Else return each service from Step 1 and the associated right remembered in Step 2.

### 4.5.20.3 Input

See external interface “IF\_SOE.GetServiceRightsByRole”.

### 4.5.20.4 Output

See external interface “IF\_SOE.GetServiceRightsByRole”.

### 4.5.20.5 Exceptions

See external interface “IF\_SOE.GetServiceRightsByRole”.

---

## 4.5.21 AF\_GetVehicleServiceInformationByFin

### 4.5.21.1 General Description

For vehicles identified by a VIN/FIN in a requested country, this AF returns an intersection of Services which are available on the sales side and of Services which are available on the technical side and the matching codes. If the locale is given as an input parameter, the configuration including the localized baumuster- and equipment descriptions is returned, too.

### 4.5.21.2 Sequence Description

#### **Step 1: Get the vehicle configuration**

- If the locale is given

- 
- Call IIF\_FetchVehicleData with the given FIN or VIN and the given locale in order to get the FIN and VIN (if available), the vehicle configuration including the localized baumuster- and equipment descriptions. Assign the baumuster-description returned by IIF\_FetchVehicleData to the output parameter - <BaumusterDescription>.
- If the locale is not given  
Call IIF\_FetchVehicleDataWithoutLocale with the given FIN or VIN to get the FIN and VIN (if available), and to retrieve the vehicle's configuration. Assign the baumuster to the baumuster-description - <BaumusterDescription>.  
If IIF\_FetchVehicleData or IIF\_FetchVehicleDataWithoutLocale aborts with an error, abort with the same error.

**Step 2: Get the currently available Mercedes connect me services for the vehicle**

**Step 2.1: Get the vehicle's Mercedes connect me services from the sales view**

Call AF\_RetrieveServicesAndMatchedRules (-> see chapter 4.4.12) to retrieve a service list and a list of equipment codes for each service. Fill the required parameters with productiveStatus="TRUE", evaluationDate="TODAY", the retrieved vehicle configuration the given BusinessArea and the customer's address country and the consumer country (contained in the vehicle configuration). In case the parameter consumer country is empty, log a WARNING and call AF\_RetrieveServicesAndMatchedRules with an empty consumer country parameter.

**Step 2.2: Get the vehicle's MBconnect services from the technical view**

If CCM\_ENABLED = "TRUE", do the following processe: Call IF\_CCM.GetServiceListRequest (-> see chapter 4.3.1) with the given VIN or FIN to retrieve the services which are technically possible.

If IF\_CCM.GetServiceListRequest (-> see chapter 4.3.1) returns an exception, ignore the error and write logs with the warning message : „An error occured when calling CCM for FIN <fin> So only the SOE view of service availability will be returned.“. Returns the vehicle's MBconnect services from the technical view same as the services from the sales view, which are got from step 2.1.

If CCM\_ENABLED = "FALSE", retrieved the services which are technically possible same as the services from the sales view (Step 2.1).

**Step 2.3: Get the intersection of the vehicle's MBconnect services from both views**

Compare the services returned from AF\_RetrieveServicesAndMatchedRules (-> see chapter 4.4.12) in Step 2.1 with the services returned from IF\_CCM.GetServiceListRequest (-> see chapter 4.3.1) in Step 2.2 and fill the list of supported services with the intersection of the returned services.

**Step 3: Get the equipment codes relevant for the intersection of the vehicle's MBconnect services**

**Step 3.1: Get the list of equipment codes for each MBconnect service from both views**

Remove each service from the set of supported services returned by AF\_RetrieveServicesAndMatchedRules which is not in the intersection determined in step 2.3.

**Step 3.2: Get the distinct equipment codes for the technical and sales available MBconnect services**

- If the locale is given:  
Fill the list of Mercedes connect me-relevant equipment codes with the distinct equipment codes left after Step 3.1 and fill the corresponding codes descriptions with the result returned by IIF\_FetchVehicleData.
- If the locale is not given:  
Fill the list of Mercedes connect me-relevant equipment codes with the distinct equipment codes left after Step 3.1. Furthermore, fill the corresponding code descriptions with the equipment codes.

#### **Step 4: Get optional service master data, independent of the MBconnect relevance**

- If the locale is given:  
Independent of the Mercedes connect me-relevance, the FIN and VIN (if available), the configuration of paint and upholstery including the localized paint- and upholstery descriptions are filled with the result returned by IIF\_FetchVehicleData.
- If the locale is not given:  
Independent of the Mercedes connect me-relevance, the FIN and VIN (if available), the configuration of paint and upholstery are filled with result returned by IIF\_FetchVehicleDataWithoutLocale and fill the corresponding descriptions with an empty string.

Propagate the vehicle's vehicle type returned by AF\_RetrieveServicesAndMatchedRules to the output parameter <vehicleType>.

#### **4.5.21.3 Input**

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
FinOrVin	Mand.	String	17		e.g. „WDD20433 11G122127“	The vehicle identifier number. May be a European FIN or the American VIN.
BusinessArea	Mand.	Enum		Enum	Example: “B2C”, “B2B”, “ALL”.	The business area, one of “B2C”, “B2B” or “ALL”. Only services from the given business area will be returned.
addressCountry	Opt.	String	2	MbcCountry.countryCode	Examples: “DE”, “CH”, “UK”, ...	<p>The address country of the customer.</p> <p>If the parameter is given, only B2C services that are available in this country will be returned.</p> <p>If the parameter is not given, the B2C services will not be filtered, i.e. all B2C services will be returned.</p> <p>B2B services will never be filtered, i.e. all B2B services will be returned.</p>
Locale	Opt.	String	5	Used as input to determine localized descriptions (from ODC).	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the returned vehicle- or equipment- descriptions.

Table 149: AF\_GetVehicleServiceInformation Input

#### 4.5.21.4 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
vehicleType	Mand.	Enum		VehicleTypeEnum	ADAPTER_VEHICLE, CONNECT_VEHICLE, NONE	Indicates the type of service the given vehicle or vehicle configuration supports. Mercedes connect me-vehicles have a communication module. Adapter-vehicles do not have a communication module but an OBD II-adapter interface. Vehicles that neither have a communication module nor an OBD II-adapter interface do not support any services integrated in Mercedes connect me.
FIN	Mand.	String	17	-	Example: WDD169034 1J764507	The FIN of the given vehicle.
VIN	Opt.	String	17	-	e.g. "WDDGJ3B B9DG12212 7"	The VIN of the requested vehicle (if available).
<b>Sales Type</b>						
Baumuster	Mand.	String	9	SalesType.Baumuster	Example: „204.231.2“	The baumuster id of the sales type.
NST-code	Opt.	String	3	SalesType.NST	Example: „CH1“	The NST code of the sales type.
BaumusterDescription	Opt.	String	256	SalesType.Description (localized)	Example: „C 180 BlueEFFICIENCY mystar Estate“	The description of the baumuster in the requested locale. If no translation is available, the value from the field Baumuster is returned.
<b>List of MBconnect relevant equipment codes for the vehicle (there can be 0 to n equipment codes)</b>						
Code	Mand.	String	5	Equipment.Code	Example: „527“	The equipment code.
CodeDescription	Opt.	String	256	Equipment.Description (localized)	Example: “COMAND Online system with Media Interface”	The description of the equipment in the requested locale. If not translation is available, the value from the field Code is returned.
<b>Upholstery</b>						
UpholsteryCode	Opt.	String	5	Equipment.Code	Example: „123“	The code of the Upholstery.
Upholstery-Description	Opt.	String	256	Equipment.Description (localized)	Example: “Leder Alcantara”	The description of the Upholstery in the requested locale as returned from UVS.
<b>Paint 1</b>						
Paint1-Code	Opt.	String	5	Equipment.Code	Example: „123“	The code of the Paint1.
Paint1-Description	Opt.	String	256	Equipment.Description (localized)	Example: “Brilliantgrau”	The description of the Paint1 in the requested locale as returned from UVS.
<b>Paint 2</b>						
Paint2-Code	Opt.	String	5	Equipment.Code		The code of the Paint2.
Paint2-	Opt.	String	256	Equip-		The description of the

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Description				ment.Description (localized)		Paint2 in the requested locale as returned from UVS.
<b>Registration Date</b>						
FirstRegistrationDate	Opt.	Date	-	VehicleConfiguration.VehicleCoreData.firstRegistrationDate	Example: "20130130"	The first registration date if available.
ConsumerCountryCode	Opt.	String	2	MbcCountry.countryCode	Example: AT, DE, CH	The ISO code of the consumer country.
<b>List of MBconnect services supported by the given vehicle and available in the requested addressCountry (0...*)</b>						
ServiceID	Mand.	Int	-	Service.ServiceID	Examples: 1, 12356, 1234567890	The service ID identifies the specific service.

Table 150: AF\_GetVehicleServiceInformationByFin Output

#### 4.5.21.5 Exceptions

None.

### 4.5.22 AF\_SaveServiceAssignmentRule

#### 4.5.22.1 General Description

Conducts a duplicate check and then creates or updates the given service assignment rule inside a change session.

#### 4.5.22.2 Sequence Description

##### *Verify if service is already inside a change session*

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter 17.4.1). If the error SESSION\_001 is thrown executions stops here.
- Verify if service is *already inside a change session of another user*: if yes, throw error SRVSESSION\_001. Execution stops here.

##### Step 1: Perform duplicate check(s)

Try to retrieve service assignment rules from the database which match ALL of the following conditions:

1. Contains the model series of the given rule
2. Contains a SalesTypeCondition that matches the one of the given rule. This is the case, if useConditionBaumusterWildcard are set for both, and if the baumuster wildcards match, and if useConditionNst are set for both, and if the list of NST codes contains at least one of the NST codes of the given rule.
3. Contains AT LEAST one of the year code combinations (including model year and an optional change year) of the given rule
4. Contains the same Boolean expression of equipment codes (i.e. the same codes connected in the same way by OR-, AND- and marked by NOT-operators) as the given rule.

5. Contains a ConsumerCountryCondition that matches one of the given rule. This is the case if “useCondition” is set for both and the list of referenced MBcCountry contains at least one common consumer country.
6. Is NOT the current rule

If such a rule exists (=either has a productive form or it is in the change session of the requesting user), abort with the error message SERMAN\_003. Fill the placeholders as follows:

4. ServiceAssignmentRule.Description + “(“ + ServiceAssignmentRule.ServiceAssignmentruleId + “)”
5. List of ModelSeries.ModelSeriesID separated by commas.
6. List of Year Code Combinations including  
YearCodeCombination.ModelYearCode + (if  
YearCodeCombination.ChangeYearCode is not empty: ) “+” +  
YearCodeCombination.ChangeYearCode, separated by commas.
7. Textual representation of Boolean expression of Equipment Codes as defined in DLG\_ServiceAssignmentRuleOverview
8. List of MBcCountry.Name separated by commas
9. SalesTypeCondition.BaumusterWildcard + List of SalesTypeCondition.NST separated by commas

The list of entities for the placeholders 2-4 consist of the respective entries which exist in both rules – the given one and the one found on the database. Table 151 shows an example of overlapping rules. In the example the Model Series, YearCodeCombinations and Equipment to display in the error message are underlined.

Rule ID	Model series	Baumuster wildcard	Year Code Combinations	Equipment	Consumer Country
1	100 200 <u>300</u>	<u>204.0*</u>	805 <u>806</u>	100 <u>200</u> <u>300</u>	„DE, FR, GB, ... (ECE15)“
2	<u>300</u> 400 500	<u>204.0*</u>	<u>806</u> 807 + 057	100 <u>200</u> <u>300</u>	„DE, FR, GB, ... (ECE15)“

Table 151: Example for overlapping/duplicate rules

#### Step 2: Create or update rule

Create or update the rule bringing it in the change session of the user (see **ApplyChangeSessionOnElement**). Also update its relations to the referenced entities.

#### **4.5.22.3 Input**

Name	Type / Length / BOM	Description
Rule	ServiceAssignmentRule	The service assignment rule to be created or updated (including the attached entities, the SalesTypeCondition and the ConsumerCountryCondition which make up the rule).

Table 152: Application function input

---

#### 4.5.22.4 Output

None.

#### 4.5.22.5 Exceptions

Error message SERMAN\_003 in case a duplicate was detected.

If the user doesn't have an opened change session, then the error SESSION\_001 is thrown.

### 4.5.23 AF\_SaveServiceCategory

#### 4.5.23.1 General Description

This application function saves the service category information provided by dialog DLG\_ServiceCategoryDetail.

#### 4.5.23.2 Sequence Description

##### **Step 1: Check uniqueness of sort order**

Check that the given sort order is not already assigned to another service category. If it is, throw error SERMAN\_022.

##### **Step 2: Save service category information**

Update all service category related data provided by the given instance of <ServiceCategory>. If no entry for this instance exists, create it.

#### 4.5.23.3 Input

Name	Type / Length / BOM	Description
ServiceCategory	ServiceCategory	The service category that has to be saved.

Table 153: AF\_SaveServiceCategory input

#### 4.5.23.4 Output

None.

#### 4.5.23.5 Exceptions

If the given sort order is already assigned to another service category, throw error SERMAN\_022.

### 4.5.24 AF\_SaveServiceDetail

This application function saves the service information provided by the dialog DLG\_ServiceDetail.

#### 4.5.24.1 Sequence Description

##### **Step 1: Save service information**

Update all service related data provided by the given instance of <Service>. If no entry for this instance exists, create it.

---

**Step 2: Save MBconnect countries the service is available in**  
Assign the given Service entity to each given MBcCountry entity.

#### 4.5.24.2 Input

Name	Type / Length / BOM	Description
Service	Service	The service that has to be saved.
<b>List of MBconnect countries the service is available in (1..*)</b>		
Country	MbcCountry	The MBconnect countries the service is available in.

Table 154: AF\_SaveServiceDetail input

#### 4.5.24.3 Output

None.

#### 4.5.24.4 Exceptions

None.

### 4.5.25 AF\_SaveServiceMaster

#### 4.5.25.1 General Description

This application function saves the service master information provided by the dialog DLG\_ServiceMasterDetail.

#### 4.5.25.2 Sequence Description

##### **Step 1: Save service information**

Update all service related data provided by the given instance of <ServiceMaster>. If no entry for this instance exists, create it.

##### **Step 2: Save authorization information**

Save all authorization related data by calling IIF\_SetRightsForResource with <ResourceTypeEnum> = SERVICEMASTER, <resourceName> = <ServiceMaster>.<ServiceMasterID> and a list of all roles with their assigned right.

#### 4.5.25.3 Input

Name	Type / Length / BOM	Description
ServiceMaster	ServiceMaster	The service master that has to be saved.
<b>List of roles (each of the following entries exist per role)</b>		
role	EntitlementRole	-
right	AccessRight	-

Table 155: AF\_SaveServiceMaster input

#### 4.5.25.4 Output

None.

#### 4.5.25.5 Exceptions

None.

---

## 4.5.26 AF\_RetrieveCountryCodeForVehicleProductMasterData

This AF will retrieve the list of countries that are affected by the changes brought to a ServiceAssignmentRule/Service.

### 4.5.26.1 Sequence Description

#### 1. Find the set of relevant services

- If the input parameter is a ServiceAssignmentRule, retrieve the associated services from the service assignment rules (ServiceAssignmentRule.Service) that are enabled (Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today). Remember the relevant services and put them to an aggregated unique list of relevant services.
- Else, in case the input parameter is a service, put the given services into the list of relevant services.

#### 2. Find the relevant CountryCode(s) from Service

- Loop for each element contained in the set of services from Step 1:
  - o Retrieve the codes of the countries for which the Service is valid and remember the distinct countryCodes (Service.MBconnectCountries.countryCode).
- Aggregate the received country codes to a unique list of country codes and return the distinct countryCodes.

### 4.5.26.2 Input

BO-Name	Type / Length / BOM	Description
changedEntity	Either ServiceAssignmentRule or Service	The changed form of a ServiceAssignmentRule/Service.

Table 156: AF\_RetrieveCountryCodeForVehicleProductMasterData Input

### 4.5.26.3 Output

BO-Name	Type / Length / BOM	Description
countryCodes	List of String	List of countryCodes affected by the change.

Table 157: AF\_RetrieveCountryCodeForVehicleProductMasterData Output

### 4.5.26.4 Exceptions

None.

---

## 4.5.27 AF\_RetrieveMarketIDForVehicleProductMasterData

This AF will retrieve the list of markets that are affected by the changes brought to a ModelSeries/Equipment.

### 4.5.27.1 Sequence Description

1. Determine set of relevant ServiceAssignmentRules
  - For the input parameter is either a ModelSeries or Equipment check if it is referenced by a service assignment rule (ServiceAssignmentRule) that covers at least one enabled service (Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today):
    - o If no reference exists then return an empty list.
    - o If reference(s) to ServiceAssignmentRules have been found that have both the released form and the change form then check each found occurrence:
      - If the changed form of the serviceAssignmentRule is in the same change session as the changed entity (input parameter) then add the changed form to the set of relevant ServiceAssignmentRules.
      - Else add the released form of the ServiceAssignmentRule to the set of relevant ServiceAssignmentRules
    - o If reference(s) to ServiceAssignmentRules have been found that only have the released form, add the ServiceAssignmentRule(s) to the set of relevant ServiceAssignmentRules
  - 2. Find set of relevant ModelSeries from the ServiceAssignmentRules
    - Loop for each ServiceAssignmentRule found relevant in Step 1:
      - o If the input parameter is not a ModelSeries then retrieve from the serviceAssignmentRule the associated ModelSeries:
        - If no reference exists then return an empty list.
        - If reference(s) to ModelSeries have been found that have both the released form and the changed form then check each found occurrence:
          - if the changed form is in the same change session as the changed entity (input parameter) then add the changed form to the set of relevant ModelSeries
          - Else add the released form of the ModelSeries to the set of relevant ModelSeries
    - 3. Find relevant market Ids from ModelSeries
      - Loop for each element contained in the set of ModelSeries from Step 2
        - retrieve the IDs of the markets for which the ModelSeries is valid and remember the distinct marketIDs
      - Return the distinct marketID(s)

### 4.5.27.2 Input

BO-Name	Type / Length / BOM	Description
changedEntity	Either ModelSeries or Equipment	The changed form of a ModelSeries/Equipment

Table 158: Application function input

---

#### 4.5.27.3 Output

BO-Name	Type / Length / BOM	Description
marketIDs	List of String	List of marketIDs affected by the change.

Table 159: Application function output

#### 4.5.27.4 Exceptions

None.

---

### 4.5.28 AF\_RetrieveOldestServiceAssignmentRuleChangeLog

This application function returns the oldest ServiceAssignmentRuleChangeLog entity. If there does not exist any ServiceAssignmentRuleChangeLog entity, the application function returns nothing.

#### 4.5.28.1 Sequence Description

If there does not exist any ServiceAssignmentRuleChangeLog entity, return nothing. Otherwise compare ServiceAssignmentRuleChangeLog.createdAt of each existing entity.

Return the entity with the smallest time stamp.

#### 4.5.28.2 Input

None.

#### 4.5.28.3 Output

Name	Type / Length / BOM	Description
ServiceAssignmentRuleChangeLog	ServiceAssignmentRuleChangeLog	The oldest ServiceAssignmentRuleChangeLog.

Table 160: AF\_RetrieveOldestServiceAssignmentRuleChangeLog Input

#### 4.5.28.4 Exceptions

None.

---

### 4.5.29 AF\_RetrieveServicesAndMatchedRules

#### 4.5.29.1 General Description

This AF returns all Services, all matched equipment codes and all RuleIDs which are configured to be available for the given configuration in the requested vehicle consumer country and customers address country on a requested date from a requested data. If no matching rule can be found or none of the services from the matching rules is available an empty list is returned (=no error).

#### 4.5.29.2 Sequence Description

The master data retrieves inside this function considers both forms of the data (productive and not yet released data from the personal change session). This is controlled by the productiveStatus parameter; see loading algorithm described in Chapter “Loading of master data elements”.

##### Step 1: Find service assignment rules

Find all service assignment rules (entity ServiceAssignmentRule) which match ALL of the following conditions:

1. The Model Series defined by the rule matches the model series (ModelSeries.ModelSeriesID must match) (Call IIF\_ResolveModelSeriesForSalesType with SalesType.baumuster as input parameter to get the model series for the sales type given in the configuration).
  2. The rule contains no SalesTypeCondition for which useConditionBaumusterWildcards is set,  
OR the rule contains a SalesTypeCondition for which useConditionBaumusterWildcards is set, and the Baumuster wildcard is matched by the baumuster of the given configuration, and, if useConditionNst is set, the list of NST codes contains the NST of the configuration.
- Note: For rules associated with transporter model series useConditionNst is never true.
3. The set of equipment codes contains the ModelYearCode of the Year Code Combination defined by the rule AND the ChangeYearCode of the Year Code Combination defined by the rule is empty or it matches the change of the configuration given as input.
  4. **The Boolean Expression assigned to the SAR is fulfilled by the vehicle configuration.** How the fulfillment can be achieved is shown on the following example:
    - Suppose the configuration is given by model series 205, model year 805 and set of equipment codes {B54, 06U, 05U, 350, 228}. In the previous steps outlined above by comparing model series and model year five service assignment rules with the IDs 6028, 6029, 6030, 6031 and 6032 are matched (see Table 161 below).

Rule ID	Model Series	Year Code Combinations (Model Year + (opt.) Change Year)	Equipment Code Condition	Consumer Countries (optional)	Services
6028	205	805, 806, 807, 808, 809	B54 and 06U	„DE, FR, GB, ... (ECE15)“	1
6029	205	805, 806, 807, 808, 809	B54 and 06U and not 404	„DE, FR, GB, ... (ECE15)“	2
6030	205	805, 806, 807, 808, 809	B54 and (06U or 07U) and 350	„DE, FR, GB, ... (ECE15)“	30, 31, 32, 33
6031	205	805, 806, 807, 808, 809	B54 and (07U or 505)	„DE, FR, GB, ... (ECE15)“	10, 21, 34

6033	205	805, 806, 807, 808, 809	B54 and (06U or 05U) and 350 and not 228	„DE, FR, GB, ... (ECE15)“	23
------	-----	----------------------------	--	------------------------------	----

Table 161: Exemplary selection of service assignment rules

Now for equipment codes from service assignment rules must be checked if they are contained in the vehicle configuration {B54, 06U, 05U, 350, 228}. This is illustrated in Table 162 below.

Rule ID	Equipment Condition	Code	Fulfilled?
6028	B54 and 06U		Yes, because the B54 and 06U are contained in the vehicle configuration
6029	B54 and 06U and not 404		Yes, because the B54 and 06U are contained and 404 is not contained in the vehicle configuration
6030	B54 and (06U or 07U) and 350		Yes, because the B54, 06U and 350 are contained in the vehicle configuration. Although the vehicle configuration doesn't have the code 07U the checking at this part of Boolean expression is successfully performed, because the 07U as well as 06U are parts of OR term, which is satisfied, even if one of its element is matched (here: 06U is matched).  Note: only matched codes from OR term are relevant for the output.
6031	B54 and (07U or 505)		No, because the 07U as well as 505 are not contained in the vehicle configuration (except B54).
6033	B54 and (06U or 05U) and 350 and not 228		No. This rule requires that the vehicle must not contain the equipment 228 (not 228).

Table 162: Matched and not matched service assignment rules

So, the checking above finds out that at this moment the three rules with the IDs 6028, 6029 and 6030 are matched.

5. If for the given vehicle configuration the <ConsumerCountry> input parameter is given then
  - (i) select the rules that contain in the consumer country list the consumer country. (for SAR with useConsumerCountry = true, the <consumer country> is found in the list of the MBcCountry associated to the ServiceAssignmentRule.ConsumerCountryCondition.) and (ii) all rules that do not have a <consumer country condition>

---

If for the given vehicle configuration the <ConsumerCountry> input parameter is NOT given then select only those rules which do not have a <consumer country condition>.

### **Step 2: Filter out services that are from another business area, unavailable in the customer's address country or disabled**

Let <tempServices> be the list of services contained in at least one of the service assignment rules determined in step 1. Remove the disabled services from <tempServices>, i.e. each service that does not satisfy each of the conditions given below.

- <Service>.<enabledFrom> is not empty and is smaller or equal than evaluationDate
- <Service>.<enabledTo> is empty or bigger than evaluationDate

If the BusinessArea from the input is not set to "ALL", retain the services of the given business area only, i.e. remove each service from <tempServices> where Service.BusinessArea does not match the BusinessArea from the input.

If the AddressCountry is given as input parameter, remove all B2C services that are not available in the address country of the user, i.e. where the ServiceMaster.BusinessArea of the corresponding service master is set to "B2C" and AddressCountry from the input is not contained in the list of MbcCountries the Service is available in.

For each service, extract all equipment codes and the service assignment rule id (they are referenced inside the associated service assignment rule). Then from the set of these equipment codes select only codes contained in vehicle configuration and which are not negated. From the example above first the following services {1, 2, 30, 31, 32, 33}, equipment codes {B54, 06U, not 404, 07U, 350} and rules {6028, 6029, 6030} are extracted. Then from these extracted codes only in vehicle configuration contained and not negated codes {B54, 06U, 350} are selected.

Note: The returned list of services, listed RuleIDs and listed Equipment Codes can be empty.

### **Step 3: Determine the vehicle type of the configuration**

Note: A vehicle is a Mercedes connect me-vehicle iff it supports at least one Mercedes connect me-service. A vehicle is an adapter-vehicle iff it supports at least one adapter-service but no Mercedes connect me-services.

- If <tempServices> is empty:
  - Assign NONE to the output parameter <vehicleType>
- If there is at least one remaining service of service class CONNECT\_SERVICE:
  - Assign CONNECT\_VEHICLE to the output parameter <vehicleType>
- If at least one service remains and each is of service class ADAPTER\_SERVICE:
  - Assign ADAPTER\_VEHICLE to the output parameter <vehicleType>

---

#### **Step 4: Filter out services not supported by the configuration's vehicle type**

- If <vehicleType> = CONNECT\_VEHICLE:
  - Remove each service of service class ADAPTER\_SERVICE from <tempServices>

#### **Step 5: Return the result**

Return <tempServices>. Link the list of IDs of matched rules and the corresponding required equipment codes to the services they cover.

Note: The returned list of services, listed RuleIDs and listed Equipment Codes can be empty.

#### **4.5.29.3 Input**

Name	Type / Length / BOM	Description
VehicleConfiguration	VehicleConfiguration	The configuration to find available services for.
BusinessArea	Enum	The business area, one of "B2C", "B2B" and "ALL". Only services from the given business area will be returned.
AddressCountry	MbcCountry.countryCode	<p>The address country of the customer. If the parameter is given, only B2C services that are available in this country will be returned. If the parameter is not given, the B2C services will not be filtered, i.e. all B2C services will be returned.</p> <p>B2B services will never be filtered, i.e. all B2B services will be returned.</p>
consumerCountry	MbcCountry.countryCode	(Optional) Filters for the services available inside the vehicle build for the given consumer country.
evaluationDate	DateDT	The date to find available services on.
productiveStatus	boolean	The data to find available services from.

Table 163: AF\_RetrieveServicesAndMatchedRules Input

#### **4.5.29.4 Output**

Name	Type / Length / BOM	Description
vehicleType	VehicleTypeEnum	Indicates the type of service the given vehicle or vehicle configuration supports. Mercedes connect me-vehicles have a communication module. Adapter-vehicles do not have a communication module but an OBD II-adapter interface. Vehicles that neither have a communication module nor an OBD II-adapter interface do not support any services integrated in Mercedes connect me.
List<Service, List<RuleID, List<EquipmentCodes>>>; for each service the Service Assignment Rule-IDs and for each rule the corresponding equipment codes are returned		
Service	Service	The available services.
List<RuleID>	List of ServiceAssignmentRules	The list of service assignment rule ids by the service.
List <EquipmentCodes>	List of EquipmentCodes	The list of equipment codes required by the service assignment rule.

Table 164: AF\_RetrieveServicesAndMatchedRules Output

#### **4.5.29.5 Exceptions**

None.

## 4.5.30 AF\_RetrieveServiceCountryAssignmentChangeLog

Based on the input parameter, this application function returns a list of ServiceCountryAssignmentChangeLog entities that either log new or deleted assignments between Services and MBcCountries.

If there does not exist any ServiceCountryAssignmentChangeLog entity with the given change operation, the internal interface returns an empty list.

### 4.5.30.1 Sequence Description

Determine each ServiceCountryAssignmentChangeLog entity with ServiceCountryAssignmentChangeLog.changeOperationType corresponds to the given input parameter and return the data.

If there is no corresponding ServiceCountryAssignmentChangeLog entity found, return an empty list.

### 4.5.30.2 Input

Name	Type / Length / BOM	Description
changeOperationType	changeOperationEnum	The requested change operation type. Possible values: "NEW", "DELETE"

Table 165: AF\_RetrieveServiceCountryAssignmentChangeLog Input

### 4.5.30.3 Output

Name	Type / Length / BOM	Description
ServiceCountryAssignmentChangeLogList	(Can be empty) List of <ServiceCountryAssignmentChangeLog>	The list of ServiceCountryAssignmentChangeLog entities with the given changeOperationType.

Table 166: AF\_RetrieveServiceCountryAssignmentChangeLog Output

### 4.5.30.4 Exceptions

None.

## 4.5.31 AF\_TestServiceAssignmentRules

### 4.5.31.1 General Description

This application function returns the MBconnect services, the service assignment rules and equipment codes that are available for the given vehicle, for customers living in the given country and for business partners, on a specific given date, from a specific given data. The vehicle is specified either by a configuration or by VIN/FIN.

Note: For vehicle identified by VIN/FIN, this application function determines a subset of the configuration.

### 4.5.31.2 Sequence Description

**Determine services supported by the given vehicle in the requested customer's address country, for the requested consumer country, on the requested date and vehicle configuration from the requested data**

- If the given vehicle is identified by VIN/FIN:

---

Call IIF\_FetchVehicleData with the given locale in order to get the vehicle configuration including the localized vehicle- and equipment descriptions.

Use IIF\_ResolveModelSeriesForSalesType to resolve the model series of the vehicle:

- a. If a model series could be retrieved:

Get all available equipment entities which are relevant for SARs with Equipment.RelevantForServiceAssignmentRules="true" that have the same product group as the model series. Compare the equipment codes of these entities with the codes returned from IIF\_FetchVehicleData and remember the intersection of both in a list of output-relevant equipment codes.

Call IIF\_GetModelYearsByProductGroup with the product group of the model series determined above as input. Intersect the set of model years with the set of equipment codes to identify the model year code for the output.

Call the **Error! Reference source not found.** to retrieve the list of supported services, the list of matching service assignment rules and the list of matching equipment codes. Fill the required parameters from the retrieved vehicle configuration, the requested address country, the requested evaluationDate, the requested productiveStatus and "ALL" as business area.

Remember the list of supported services, matching service assignment rules and matching equipment codes for the output.

- b. If no model series could be retrieved:

Return an empty list of supported services and abort with the Error Message SERMAN\_018.

- Otherwise, if the given vehicle is identified by configuration:

Call **Error! Reference source not found.** to retrieve the list of supported services, the list of matching service assignment rules and the list of matching equipment codes. Fill the required parameters from the given vehicle configuration, the requested address country, consumer country, evaluationDate, productiveStatus and "ALL" as business area.

Remember the list of supported services, matching service assignment rules and matching equipment codes for the output.

If the given vehicle is identified by VIN/FIN:

Fill the list of relevant equipment codes with the list of output-relevant equipment codes from step a). The corresponding code descriptions are filled with the output of the IIF\_FetchVehicleData.

Otherwise, if the given vehicle is identified by configuration:

Leave the list of relevant equipment codes empty.

Fill the list of supported services, the list of matching service assignment rules and the list of matching equipment codes according to the output of the **Error! Reference source not found..**

#### 4.5.31.3 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
addressCountry	Mand	String	2	MbcCountry.countryCode	Examples: "DE", "CH", "UK", ...	The address country of the user. Based on this country, the list of available services is determined.
consumerCountry	Optional	String	2	MbcCountry.countryCode	Examples: "DE", "CH", "UK", ...	Filters for the services available inside a vehicle that was build for the given consumer country.
productiveStatus	Mand.	Boolean	-	-	True, false	This attribute determines whether the tested data should be the productive data or data of the change session plus productive data.
evaluationDate	Mand.	DateDT	-	-	Example: "20.06.2014"	This attribute determines the date on which the services be tested.
locale	Mand.	String	5	-	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the retuned vehicle- or equipment- descriptions.
Vehicle for which the serviceList needs to be determined: VIN available provided						
finOrVin	Mand. If "VIN available" selected	String	17	-	Example: WDD1690341 J764507	The given VIN/FIN.
Vehicle for which the serviceList needs to be determined: VIN not available provided						
modelSeries	Mand. If "VIN not available" selected	String	3	ModelSeries.ModelSeriesId	Example: "205"	The model series.
baumuster	Mand. If "VIN not available" selected	String	7	SalesType.baumuster	Example: "2050041"	The baumuster as part of the sales type.
NST	Opt.	String	3	SalesType.NST	Example: "DE1"	The NST as optional part of the sales type.
ModelYearCode	Mand. If "VIN					The model year of the configuration.  Mapping: Add the

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
	not available" selected					YearCodeCombination.ModelYearCode to the set of equipments.
ChangeYear Code	Mand. If "VIN not available" selected	String	15	YearCodeCombination	Example: "804", "804+054"	The year code combination including the model year and an optional change year.
List of MBconnect relevant equipment codes (only given, if finOrVin is not given) (0 to n equipment codes)						
Code	Mand. If "VIN not available" selected	String	5	Equipment.Code	Example: "527"	The equipment code of the given vehicle configuration.

Table 167: AF\_TestServiceAssignmentRules Input

#### 4.5.31.4 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Sales Type</b>						
modelSeries	Mand.	String	3	ModelSeries.ModelSeriesId	Example: "205"	The model series.
baumuster	Mand.	String	7	SalesType.baumuster	Example: "2050041"	The baumuster as part of the sales type.
NST	Opt.	String	3	SalesType.NST	Example: "DE1"	The NST as optional part of the sales type.
ModelYearCode	Mand.	String	5	YearCodeCombination.ModelYearCode	Example: "804"	The code of the model year.
ChangeYearCode	Opt.	String	5	YearCodeCombination.ChangeYearCode	Example: "056"	The code of the change year.
ConsumerCountryCode	Opt.	String	2	MbcCountry.countryCode	Examples: "DE", "CH", "UK", ...	Country the vehicle was produced for.
<b>List of MBconnect relevant equipment codes for the vehicle (there can be 0 to n equipment codes)</b>						
Code	Mand.	String	5	Equipment.Code	Example: "527"	The equipment code.
CodeDescription	Mand.	String	256	Equipment.Description (localized)	Example: "COMAND Online system with Media Interface"	The description of the equipment in the requested locale. If no translation is available, the value from the field Code is returned.

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of services supported by the given vehicle, available in the requested addressCountry, on the requested date from the requested data (0...*)</b>						
ServiceID	Mand.	Int	-	Service.ServiceID	Examples: 1, 12356, 1234567 890	The service ID identifies the specific service.
Version	Mand.	Int	-	Service.versionNumber	Example: 1, 2	The version number of the specific service.
ServiceName	Mand.	String	-	ServiceMaster.Name	Examples: "Vehicle Tracker"	The name of the service.
<b>Innerlist of matching service assignment rules</b>						
RuleID	Mand.	Int		ServiceAssignmentRule	Examples: 1, 12, 123, 1234	The Rule for a specific service as a integer list.
<b>Inner-Inner list of matching equipment codes</b>						
MergedMatchCode	Mand.	String		-	Examples: "05U", "228"	The merged equipment codes for a specific service assignment rule returned as a string list.

Table 168: AF\_TestServiceAssignmentRules Output

#### 4.5.31.5 Exceptions

Error message SERMAN\_018 in case the provided modelSeries is missing in the database.

### 4.5.32 AF\_UpdateServices

#### 4.5.32.1 General Description

This application function updates the Service entities.

#### 4.5.32.2 Sequence Description

##### Step 1: Perform the updateMode

Compare the <Service> given as input parameter with the existing <Service> in SOE retrieved by calling AF\_GetServices.

##### If the updateMode = ADD

Add all instances of <Service> which are given as input parameter and not parts of the existing <Service>.

##### If the updateMode = UPDATE

Update all instances of <Service> that are available in SOE which are not completely equal to the <Service> given as input parameter.

##### If the updateMode = DELETE

Delete all instances of <Service> that are available in SOE which are not parts of the <Service> given as input parameter.

## Step 2: Check if Service modifications are relevant for updating Contracts

- If the modified element is an enabled Service (Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today), proof if the assignment to at least one MBcCountry has been changed (therefore the modification is relevant for updating contracts):
- For each MBcCountry the assignment to the Service has been added, call IIF\_UpdateServiceCountryAssignmentChangeLog with the Service.ServiceID, the change operation type “NEW” and the countryCode of the MBcCountry
- For each MBcCountry the assignment to the Service has been deleted, call IIF\_UpdateServiceCountryAssignmentChangeLog with the Service.ServiceID, the change operation type “DELETE” and the countryCode of the MBcCountry

### 4.5.32.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the service entities.
<b>List&lt;Service&gt;: List of all Service entities</b>		
serviceID	Service.serviceID	The ID of a Service.
partNumber	Service.partNumber	The part number of a Service (“Sachnummer”).
versionNumber	Service.versionNumber	The version of a Service.
licenseRequired	Service.licenseRequired	Tells whether a license is required or not in order to use this service.
contractDuration	Service.contractDuration	Optional: Describes the contract duration in months, in case a contract for this service is established. This attribute is optional.
contractStartTrigger	Service.contractStartTrigger	Indicates what event triggers the begin of a contract related to this service.
enabledFrom	Service.enabledFrom	The start date of a service (including this day). From this date the service is available of the sales' point of view.
enabledTo	Service.enabledTo	The last date of a service (including this day). From this date the service is not available of the sales' point of view.
technicalActivationPath	Service.technicalActivationPath	Determines the technical activation path, i.e. the system, that has to be contacted in order to activate or deactivate the service in the vehicle.
serviceClass	Service.serviceClass	Mercedes connect me-services are available for Mercedes connect me-vehicles exclusively. Mercedes connect me-vehicles are vehicles that have a communication module. Adapter-services are available for adapter vehicles exclusively. Adapter-vehicles are vehicles that have an OBD II adapter-interface but no communication module.
ServiceMasterID	ServiceMaster.serviceMasterID	The ID of the ServiceMaster.
- Inner List of <String(4096)>		
Description	Service.description	Translation of a service's description (i18n)
- Inner List of <MbCountry>		
countryCode	MbCountry.countryCode	The CountryCode of a Country.

Table 169: AF\_UpdateServices input

### 4.5.32.4 Output

None.

---

#### 4.5.32.5 Exceptions

None.

### 4.5.33 AF\_UpdateServiceAssignmentRules

#### 4.5.33.1 General Description

This application function updates the ServiceAssignmentRule entities.

#### 4.5.33.2 Sequence Description

##### Step 1: Perform the updateMode

Compare the <ServiceAssignmentRule> given as input parameter with the existing <ServiceAssignmentRule> in SOE retrieved by calling AF\_GetServiceAssignmentRules.

##### If the updateMode = ADD

Add all instances of <ServiceAssignmentRule> which are given as input parameter and not parts of the existing <ServiceAssignmentRule>.

##### If the updateMode = UPDATE

Update all instances of <ServiceAssignmentRule> that are available in SOE which are not completely equal to the <ServiceAssignmentRule> given as input parameter.

##### If the updateMode = DELETE

Delete all instances of <ServiceAssignmentRule> that are available in SOE which are not parts of the <ServiceAssignmentRule> given as input parameter.

#### Step 2: Check if Service assignment rule modifications are relevant for updating Contracts

Verify if the modification of the ServiceAssignmentRule is relevant for updating contracts:

If

- The productive or the changed service assignment rule references one or more enabled service(s) (that means Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today)
- AND there are modifications besides changes to the description, the long description or changes in the assignments to disabled services (added or deleted assignments; disabled services means Service.enabledFrom is not given or Service.EnabledFrom is greater than today)

then

Update the ServiceAssignmentRuleChangeLog by calling IIF\_UpdateServiceAssignmentRuleChangeLog with the ServiceAssignmentRule ID, the ModelSeriesID, the Services of the ServiceAssignmentRule and the change operation (“NEW”, “DELETE” or “UPDATE”). The ModelSeriesID and the related Services are determined as follows:

If the ServiceAssignmentRule is new or has been marked for deletion:

- pass in the ModelSeriesID it is assigned to
- pass in the list of related services

else (the ServiceAssignmentRule has been changed)

- if the assignment to the model series has been changed
  - use the old ModelSeriesID and the new ModelSeriesID
- else (the model series has not been changed)
  - use the ModelSeriesID
- if the assignment to Services has been changed:
  - use the services to which the assignment has been added or deleted (to reduce the amount of services that have to be checked in order to find the affected contracts)
- else
  - use the related services.

#### 4.5.33.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the service assignment rule entities.
<b>List&lt;ServiceAssignmentRule&gt;: List of all ServiceAssignmentRule entities</b>		
serviceAssignmentRuleId	ServiceAssignmentRule.serviceAssignmentRuleId	The ID of a ServiceAssignment Rule.
Description	ServiceAssignmentRule.description	A name for a rule. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
longDescription	ServiceAssignmentRule.longDescription	A comment on the rule that explains or summarizes the rule in greater detail. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
modelSeriesID	ModelSeries.modelSeriesID	The ID of a ModelSeries.
- Inner List of <YearCodeCombination>		
ModelYearCode	ModelYear.code	The Code of a ModelYear.
ChangeYearCode	String	The Code of a ChangeYear
- Inner List of <AndTerm>		
AndTermOrder	AndTerm.displayOrder	The sequence number of OR terms within Boolean expression. Used to restore the OR terms in the sequence originally created.
• Inner List of <OrTerm>		
Code	OrTerm.EquipmentCode Equipment.code	The Code of an Equipment
isNegated	OrTerm.negated	Boolean flag that indicates whether the equipment code is negated within Boolean expression
OrTermOrder	OrTerm.displayOrder	The sequence number of equipment code within OR term of Boolean expression. Used to restore equipment codes in the OR term in the sequence originally created.
- SalesTypeCondition		
baumusterWildcard	SalesTypeCondition.baumusterWildcard	Search condition for a matching Baumuster. This pattern may contain wildcards.
- Inner List of <NST>		
NST	SalesTypeCondition.NST	Search condition for a National Sales Type.
useConditionBaumusterWildcards	SalesTypeCondition.useConditionBaumusterWildcards	Boolean flag that indicates if the search condition under BaumusterWildcard shall be used during a search (TRUE) or ignored (FALSE).
useConditionNst	SalesTypeCondition.useConditionNst	Boolean flag that indicates if the search condition under Nst shall be used during a search (TRUE) or ignored (FALSE).
- ConsumerCountryCondition		
- Inner List of <MbcCountry>		
countryCode	MbcCountry.countryCode	The CountryCode of a Country.
useCondition	ConsumerCountryCondition	Tells whether the country condition shall be taken in

Parameter Name	Type / Length / BOM	Description
	n.useCondition	account (true) or not (false) when the rule is evaluated.
- Inner List of <Service>		
ServiceID	Service.serviceID	The ID of a Service.

Table 170: AF\_UpdateServiceAssignmentRules input

#### 4.5.33.4 Output

None.

#### 4.5.33.5 Exceptions

None.

### 4.5.34 AF\_UpdateServiceAssignmentRuleChangeLog

Each time a service assignment rule has been newly created/deleted/changed and the appropriate change session has been released, this internal interface creates a new ServiceAssignmentRuleChangeLog entity.

#### 4.5.34.1 Sequence Description

Create a new instance of ServiceAssignmentRuleChangeLog and set its attributes corresponding to the given input data. Set ServiceAssignmentRuleChangeLog.createdAt to the current timestamp.

#### 4.5.34.2 Input

Name	Type / Length / BOM	Description
ServiceAssignmentRuleID	ServiceAssignmentRule.ServiceAssignmentRuleID	The ID of the modified service assignment rule.
changeOperationType	ChangeOperationEnum	The change operation type of the modified service assignment rule, Possible values: "NEW", "DELETE", "UPDATE"
ModelSeriesList	List of <ModelSeries.ModelSeriesID>	<u>In case only the assignment to model series has been changed:</u> The list of model series for which the assignment to the service assignment rule has been added or deleted. <u>Otherwise:</u> Each model series that is assigned to the service assignment rule before or after it has been modified.
ServicesList	List of <Service.serviceID>	<u>In case only the assignment to services has been changed:</u> The list of Services for which the assignment to the service assignment rule has been added or deleted. <u>Otherwise:</u> Each Service that is assigned to the service assignment

	rule before or after it has been modified.
--	--

Table 171: AF\_UpdateServiceAssignmentRuleChangeLog Input

#### 4.5.34.3 Output

None.

#### 4.5.34.4 Exceptions

None.

### 4.5.35 AF\_UpdateServiceCategories

#### 4.5.35.1 General Description

This application function updates the ServiceCategory entities.

#### 4.5.35.2 Sequence Description

Compare the <ServiceCategory> given as input parameter with the existing <ServiceCategory> in SOE retrieved by calling AF.GetServiceCategories.

##### If the updateMode = ADD

Add all instances of <ServiceCategory> which are given as input parameter and not parts of the existing <ServiceCategory>.

##### If the updateMode = UPDATE

Update all instances of <ServiceCategory> that are available in SOE which are not completely equal to the <ServiceCategory> given as input parameter.

##### If the updateMode = DELETE

Delete all instances of <ServiceCategory> that are available in SOE which are not parts of the <ServiceCategory> given as input parameter.

#### 4.5.35.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the modus how to update the service category entities.
<b>List of Service Categories (each of the following parameters exist per service category)</b>		
serviceCategoryID	ServiceCategory.ServiceCategoryID	The ServiceCategoryID identifies the service category the service is assigned to.
sortOrder	ServiceCategory.SortOrder	The sort order of the category, i.e. the position this category appears in the sequence of categories, e.g. when displayed by frontend systems.
<b>• Inner List of service category names (each of the following entries exist per locale)</b>		
Locale		
serviceCategoryName	ServiceCategory.name	The localized service category name.

Table 172: AF\_UpdateServiceCategories input

#### 4.5.35.4 Output

None.

---

#### 4.5.35.5 Exceptions

None.

### 4.5.36 AF\_UpdateServiceCountryAssignmentChangeLog

Each time the assignment between an enabled Service and an MBcCountry entity is added or deleted and the appropriate change session has been released, this application function creates or updates the corresponding ServiceCountryAssignmentChangeLog entity.

#### 4.5.36.1 Sequence Description

Proof if there is a ServiceCountryAssignmentChangeLog entity where ServiceCountryAssignmentChangeLog.serviceID matches the given ServiceID and ServiceCountryAssignmentChangeLog.countryCode matches the given countryCode.

- If not, create a new instance of ServiceCountryAssignmentChangeLog with the data given as input parameters. Set ServiceCountryAssignmentChangeLog.createdAt to the current time stamp.
- Otherwise, compare the ServiceCountryAssignmentChangeLog.changeOperationType with the changeOperationType given as input parameter.  
If the operation types differ, delete the existing ServiceCountryAssignmentChangeLog entity. Otherwise do nothing.

#### 4.5.36.2 Input

Name	Type / Length / BOM	Description
serviceID	Service.ServiceID	The ID of the service for which the assignment to a MBcCountry has been modified.
Country	MBcCountry.countryCode	The countryCode of the MBcCountry for which the assignment has been modified.
changeOperationType	ChangeOperationEnum	The change operation type of the modified assignment between Service and MBcCountry. Possible values: "NEW", "DELETED".

Table 173: AF\_UpdateServiceCountryAssignmentChangeLog Input

#### 4.5.36.3 Output

None.

#### 4.5.36.4 Exceptions

None.

---

## 4.5.37 AF\_UpdateServiceMasters

### 4.5.37.1 General Description

This application function updates the ServiceMaster entities.

### 4.5.37.2 Sequence Description

Compare the <ServiceMaster> given as input parameter with the existing <ServiceMaster> in SOE retrieved by calling AF.GetServiceMasters.

#### If the updateMode = ADD

Add all instances of <ServiceMaster> which are given as input parameter and not parts of the existing <ServiceMaster>.

#### If the updateMode = UPDATE

Update all instances of <ServiceMaster> that are available in SOE which are not completely equal to the <ServiceMaster> given as input parameter.

#### If the updateMode = DELETE

Delete all instances of <ServiceMaster> that are available in SOE which are not parts of the <ServiceMaster> given as input parameter.

### 4.5.37.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the modus how to update the service entities.
<b>List&lt;ServiceMaster&gt;: List of all ServiceMaster entities</b>		
serviceMasterId	ServiceMaster.serviceMasterId	The ID of the ServiceMaster.
serviceCategoryID	ServiceCategory.ServiceCategoryID	The ServiceCategoryID identifies the service category the service is assigned to.
activateAutomatically	ServiceMaster.activateAutomatically	States if service(s) assigned to the service master is/are activated automatically when the vehicle is registered to a customer.
confirmationTokenNeeded	ServiceMaster.confirmationTokenNeeded	If "TRUE" service(s) assigned to the service master need(s) a confirmed connection between the user and a vehicle through a verification token in. Such service(s) must be activated only after a successful verification.
personalVerificationNeeded	ServiceMaster.personalVerificationNeeded	If "TRUE" a retailer has to verify the identity of the customer in order to use service(s) assigned to the service master.
trustLevel	ServiceMaster.trustLevel	Determines the minimal trust level of a vehicle registration in order to use service(s) assigned to the service master.
- Inner List of <String(256)>		
name	ServiceMaster.name	Translation of a service master name (i18n)
- Inner List of custom properties		
customProperty	ServiceMaster.customProperties	List of the custom properties required for the activation of service(s) assigned to the service master.
- Inner List of <ProfileDataFieldItem>		
requiredCustomerInformation	ServiceMaster.requiredCustomerInformation.fieldID	List with the mandatory fields for service(s) assigned to the service master that relate to attributes of the customer profile.
requiredVehicleInformation	ServiceMaster.requiredVehicleInformation.fieldID	List with the mandatory fields for service(s) assigned to the service master that relate to attributes of a vehicle.
- Inner List of <ProfileDataFieldGroup>		
requiredCustomerInformation	ServiceMaster.requiredCustomerInformation.groupID	List with the mandatory fields for service(s) assigned to the service master that relate to attributes of the customer profile.
	ServiceMas-	List with the mandatory fields for service(s) assigned to the service mas-

Parameter Name	Type / Length / BOM	Description
requiredVehicleInformation	ter.requiredVehicleInformation.groupID	ter that relate to attributes of a vehicle.

Table 174: AF\_UpdateServiceMasterss input

#### 4.5.37.4 Output

None.

#### 4.5.37.5 Exceptions

None.

### 4.6 Batches

None.

### 4.7 Error Messages

Message Id	Fault Title	Fault Message
SERMAN_003	Duplicate rule detected	The service assignment rule you are trying to save is partly or fully identical to the already existing rule "<1>". The rules are overlapping in the following aspects. Model Series: <2> Sales Type Condition <6> Year Code Combinations: <3> Equipment: <4> Consumer Countries: <5>
SERMAN_007	Mandatory field missing	Please select at least one model series, one year code combination and one service.
SERMAN_008	Invalid service ID	The requested service <1> is not known to SOE.
SERMAN_010	Invalid marketID	The requested marketID <1> is not known to SOE.
SERMAN_011	Mandatory field missing	Please select at least one MBconnect country, the service is available in.
SERMAN_016	Invalid evaluationDate	The evaluationDate has to be today or in the future, not in the past.
SERMAN_017	Mandatory field missing	Please select a Source, enter a service availability Date, select a Country and select a vehicle. If you selected "VIN available" enter a VIN/FIN or if you selected "VIN not available" select a Sales Type and select a Year Code Combination.
SERMAN_018	Invalid modelSeries	The selected modelSeries is not known to SOE.
SERMAN_019	Error connecting to system CCM	An unexpected error occurred while connecting to the system CCM.
SERMAN_020	Mandatory field missing	The mandatory field "<1>" is not provided. Please fill this field. Hint: Mandatory fields are marked with an asterisk ("*").
SERMAN_021	Cannot delete service category, because it is referenced by service masters	The service category <1> could not be deleted, because it is referenced by one or more service masters: <2>.
SERMAN_022	Duplicate sort order	The chosen sort order is already assigned to another service category. Please choose a different order.
SRVSESSION_001	Service <Service> is locked by another user	The service <service> you are trying to modify is already being edited by user <user>. Changes are not possible and will be discarded automatically.
SRVSESSION_002	ServiceAssignmentRule <ServiceAssignmentRule> is locked by another user	The service assignment rule <ServiceAssignmentRule> you are trying to modify is already being edited by user <user>. Changes are not possible and will be discarded automatically.
SRVSESSION_003	ServiceCategory <ServiceCategory> is locked by another user	The service category <ServiceCategory> you are trying to modify is already being edited by user <user>. Changes are not possible and will be discarded automatically.

---

		ly.
--	--	-----

Table 175: Messages of the component Services

# 5 Component Service Management

This component is responsible for managing the relationship between the service master data and user agreements master data.

## 5.1 Dialogs

### 5.1.1 DLG\_ServicesOverview

#### 5.1.1.1 General Description

This dialog is the entry point for maintaining the available MBconnect services. It displays a list of all currently maintained services and allows the creation, deletion or editing of the available MBconnect services.

The screenshot shows a software interface titled 'DLG\_ServicesOverview'. At the top, there is a button labeled 'Add new Service'. Below this is a toolbar with a 'Delete' button and dropdown menus for 'Items per page' (set to 10) and 'Page' (set to 1 of 3). The main area is a table listing seven services:

	Master ID	Service ID	Name	Version	Enabled	Opened Session	Action
<input type="checkbox"/>	1	1	Accident Management	1	Yes		<button>Edit</button>
<input type="checkbox"/>	1	11	Accident Management	2	Yes		<button>View</button>
<input type="checkbox"/>	2	2	Breakdown Management	1	Yes		<button>Edit</button>
<input type="checkbox"/>	2	12	Breakdown Management	2	Yes		<button>Edit</button>
<input type="checkbox"/>	3	3	Maintenance & Predictive Diagnosis	1	Yes		<button>Edit</button>
<input type="checkbox"/>	3	13	Maintenance & Predictive Diagnosis	2	Yes		<button>Edit</button>
<input type="checkbox"/>	4	4	Life Traffic	1	Yes		<button>Edit</button>

At the bottom, there is another 'Delete' button, 'Items per page' dropdown (set to 10), 'Page' dropdown (set to 1 of 3), and an 'Add new Service' button.

Figure 46: DLG\_ServicesOverview

#### 5.1.1.2 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all available MBconnect services and sort them by Service.ServiceId together with the permitted actions for each loaded service by calling the loading algorithm described in Chapter “Loading of master data elements”.
“Add new Service”	Button	Switches to the dialog DLG_ServiceDetail (→ see chapter 5.1.2) to create a new MBconnect service. The dialog is opened in new mode.
“Delete”	Button	After displaying a confirmation popup delete the selected MBconnect services by calling the AF_DeleteServices (→ see chapter 5.5.4). Display potential error messages in the according message popup.

Linked label / button labeling	Type	Action description
"Edit"	Button	Switches to the dialog DLG_ServiceDetail (→ see chapter 5.1.2) to edit the MBconnect service. The dialog is opened in new mode.
"View"	Button	Switches to the dialog DLG_ServiceDetail (→ see chapter 5.1.2) to edit the MBconnect service. The dialog is opened in read-only mode.

Table 176: Buttons and functions (DLG\_ServicesOverview)

### 5.1.1.3 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Table column "ServiceMasterId"	Label	The id of the service master.	ServiceMaster.ServiceMasterId
Table column "ServiceId"	Label	The id of the service.	Service.ServiceId
Table column "Name"	Label	The description of the service (localized to the user's language).	ServiceMaster.Name
Table column "Version"	Label	The version number of the service.	Service.versionNumber
Table Column "Enabled"	Label	States if the service is enabled ("Yes") or not.	Displayed value: - If Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today: display "Yes". -Otherwise: "No",
Table Column "Opened Sessions"	Label/Icon	Indicates whether the entity Document is edited inside a change session.	-

Table 177: Form fields and front-end data objects

### 5.1.1.4 Dialog field validation

None.

### 5.1.1.5 Configurability (incl setting for roles)

None.

### 5.1.1.6 Dialog Elements States

See states of the common elements displayed by the "Overview" dialogs.

## 5.1.2 DLG\_ServiceDetail

### 5.1.2.1 General Description

This dialog allows the user to view or edit a Mercedes Connect me service. It allows to maintain the contract activation trigger and contract period. The dialog also includes maintaining the translations of the service description.

Additionally, the user can select countries, the service is available in.

Note:

- The section "Service enablement" allows maintaining the enablement of the Ser-

vice. Each service that is enabled is provided to third party systems by the external SOE interfaces.

- The required data fields per service are not country specific:  
This means, it is only possible to maintain the Mercedes Connect me countries, the service is available in. It is not possible to configure different requirements for services in different countries. In case of the account related mandatory fields, it will be ensured that the administrator can only reference to non market specific data fields. To do so, all data fields that are shown in the list have to be optional for all countries.

Service ID*	4																	
Version*	2																	
Service Master*	Accident Management (1)																	
Part number	A345511342																	
Service enablement																		
Is enabled: Yes																		
Enabled from:	01.01.2014	Enabled to: 01.01.2015																
Contract period (months)	36																	
License required	<input checked="" type="checkbox"/>																	
Service Class*	Mercedes connect me-service																	
Contract activation trigger*	Vehicle registration																	
Technical activation path*	Activate via DaiVB																	
Service Description																		
<p>This service is available in the following MBconnect countries:*</p> <table border="1"> <thead> <tr> <th>Language</th> <th>Translation</th> </tr> </thead> <tbody> <tr> <td>German (Germany)</td> <td>Beschreibung der Dienstversion</td> </tr> <tr> <td>German (Austria)</td> <td></td> </tr> <tr> <td>German (Switzerland)</td> <td></td> </tr> <tr> <td>English (UK)</td> <td>Description of the service versio</td> </tr> <tr> <td>English (Switzerland)</td> <td></td> </tr> <tr> <td>French (France)</td> <td>Description de la version de serv</td> </tr> <tr> <td>French (Belgium)</td> <td></td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <input type="checkbox"/> Select/Deselect all  <input checked="" type="checkbox"/> Austria (At)  <input checked="" type="checkbox"/> Belgium (Be)  <input type="checkbox"/> Denmark (Dk)  <input checked="" type="checkbox"/> France (Fr)  <input checked="" type="checkbox"/> Germany (De)  <input type="checkbox"/> Hungary (Hu)  <input checked="" type="checkbox"/> Italy (It)  <input checked="" type="checkbox"/> Luxembourg (Lu)  <input checked="" type="checkbox"/> Netherlands (Nl)  <input type="checkbox"/> Poland (Pl)  <input checked="" type="checkbox"/> Slovakia (Sk)  <input type="checkbox"/> Spain (Es)  <input type="checkbox"/> Switzerland (Ch)  <input checked="" type="checkbox"/> Czech Republic (Cz)         </div>			Language	Translation	German (Germany)	Beschreibung der Dienstversion	German (Austria)		German (Switzerland)		English (UK)	Description of the service versio	English (Switzerland)		French (France)	Description de la version de serv	French (Belgium)	
Language	Translation																	
German (Germany)	Beschreibung der Dienstversion																	
German (Austria)																		
German (Switzerland)																		
English (UK)	Description of the service versio																	
English (Switzerland)																		
French (France)	Description de la version de serv																	
French (Belgium)																		
<input type="button" value="Save"/> <input type="button" value="Cancel"/>																		

---

Figure 47: DLG\_ServiceDetail

### 5.1.2.2 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		<p>Edit, Read only mode: Load the service and all available translations. Match the available translations against all maintainable translations, which are available in the application configuration.</p> <p>Load all MBconnect countries, the service is available in. Also load all available countries and list them in the order of their names. For a list of MBconnect countries, IIF_GetMBconnectCountries is called.</p> <p>New mode: Load the maintainable languages from the application configuration and generate the table rows. The service id will be empty until the service is saved.</p> <p>Load all available countries and sort them in the order of their names. For a list of countries, IIF_GetMBconnectCountries is called.</p> <p>Consider the loading algorithm described in Chapter Loading of master data elements</p>
“Save”	Button	Save the currently maintained MBconnect service and navigate back to <b>DLG_ServicesOverview</b> . Call AF_SaveService with the list of MbcCountry entities - that are selected - as input parameter in order to save the service details. Bring the currently maintained Service in the change session of the user. (see general algorithm ApplyChangeSessionOnElement)
“Cancel”	Button	Discard all changes and navigate back to <b>DLG_ServicesOverview</b>
“Select/Deselect all”	Checkbox	Selects or deselects all check boxes in the section.

Table 178: Buttons and functions (DLG\_ServiceDetail)

### 5.1.2.3 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Service Id	Textbox	The id of the service. The id has to be unique and set manually.	Service.serviceld  Available values: All positive integers that are not referenced yet.
Version	Textbox	The version number of the service.	Service.versionNumber
Service Master	Dropdown	The ServiceMaster the Service is assigned to.  Default: empty	<u>Displayed value:</u> ServiceMaster.name + "(" + ServiceMaster.ServiceMasterId + ")"  <u>Checked value:</u> The Service Master entity, the service entity is assigned to.  <u>Available values:</u> All available ServiceMaster entities.  <u>Sorted by:</u> ServiceMaster.name
Part number	Textbox	The part number of the service for the license purchasing process ("Sachnummer"). The part number has to be unique and set manually.	Service.partNumber  Available values: All Strings that are not referenced yet.
<b>Section "Service enablement"</b>			
Is enabled	Label	Shows whether the service is enabled or not. Services that are enabled are provided to third party systems.	<u>Displayed value:</u> - "Yes", if for the <u>productive</u> form of the Service applies: Service.EnabledFrom is given and smaller or equal than today and Service.enabledTo is empty or bigger than today.  "No", otherwise.
EnabledFrom	Textbox	Specifies the start date for the enablement of this service	Service.EnabledFrom
EnabledTo	Textbox	Specifies the end date for the enablement of this service.	Service.EnabledTo
License Required	Checkbox	The information whether there must exist a valid license in order to use a service.	Service.licenseRequired
Contract activation trigger	Dropdown	Describes what triggers the activation of a service and so starts a connected contract period.  Default: VEHICLE_REGISTRATION	<u>Displayed value:</u> ContractStartTriggerEnum.Description  <u>Selected value:</u> Service.contractStartTrigger  <u>Available values:</u> All from enumeration ContractStartTriggerEnum
Contract period (months)	Textbox	Optionally describes how long the contract duration lasts if a contract for this service is started.	Service.contractDuration
Technical activation path	Dropdown	This attribute determines whether a technical activation is needed or not.  Default: ACTIVATE_VIA_DAIVB	<u>Displayed value:</u> TechnicalActivationPathEnum.Description  <u>Selected value:</u> Service.technicalActivationPath

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
			Available values: All from enumeration TechnicalActivationPathEnum
Service Class	Dropdown	<p>A service's class indicates the type of vehicle it is supported by. The class of Mercedes connect me-services is supported by Mercedes connect me-vehicles. Mercedes connect me-vehicles are vehicles that have a communication module. The class of adapter-services is supported by adapter-vehicles. Adapter-vehicles are vehicles that have an OBD II-adapter interface but no communication module.</p> <p>Default: CONNECT_VEHICLE</p>	<u>Displayed value:</u> <u>ServiceClassEnum.Description</u>  <u>Selected value:</u> Service.serviceClass  Available values: All from enumeration ServiceClassEnum
Table “MBconnect countries the service is available in”	Checkbox list	Defines the MBconnect countries, the MBconnect service is available in.	<u>Displayed value:</u> MbcCountry.name + "(" + MbcCountry.countryCode + ")"  <u>Checked values:</u> All MbcCountry entities, the Service entity is assigned to.  <u>Available values:</u> All MbcCountry entities returned by IIF_GetMBconnectCountries.  <u>Enabled:</u> If ServiceMaster.BusinessArea is set to “B2C”.  <u>Disabled:</u> If ServiceMaster.BusinessArea is set to “B2B”.
Table column “Language”	Label	Locale of the maintained translation.	-
Table column “Translation”	Textbox	The description of the service (localized).	Service.description
ReadOnlyInfo	Label	Presents this information: “The dialog is opened in read-only mode.”	-

Table 179: Form fields and front-end data objects (DLG\_ServiceDetail)

#### 5.1.2.4 Dialog field validation

“Service ID”, “Service Master” and “Contract Activation Trigger” cannot be left blank. Also at least one translation should be filled.

Linked Field	Validation	Error Message
Version	All positive integers that are not referenced by the selected service master yet.	SERMGM_013
Service Master	The combination of Version and Service Master has to be unique.	SERMGM_014
MBconnect countries the service is available in	In the given list at least one item must be selected.	SERMGM_011
EnabledFrom	The service enablement date needs to be provided in format dd.mm.yyyy	SERMGN_009
	Date must be greater than today if modified.	SERMGN_010

EnabledTo	The service enablement date needs to be provided in format dd.mm.yyyy	SERMGN_009
	Date must be greater than today and greater than EnabledFrom if modified.	SERMGN_012

Table 180: Dialog field validation (DLG\_ServiceDetail)

### 5.1.2.5 Configurability (incl setting for roles)

None.

### 5.1.2.6 Dialog Elements States

Linked Label	Type	State Description
Service Id	Textbox	Will always be visible.
Version	Textbox	Will always be visible.
Service Master	Dropdown	Will always be visible.
Part number	Textbox	Will always be visible.
Section "Service enablement"	-	Will always be visible.
EnabledFrom	Textbox	Disabled if for the <u>productive</u> form of the Service the following applies: Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today.
EnabledTo	Textbox	Disabled if for the <u>productive</u> form of the Service the following applies: Service.enabledTo is given and (Service.enabledTo +1) equals today.
License Required	Checkbox	<u>Enabled</u> : If ServiceMaster.BusinessArea is set to "B2C". <u>Disabled</u> : If ServiceMaster.BusinessArea is set to "B2B", this checkbox will be disabled and the value will be set to "False".
Contract activation trigger	Dropdown	Will always be visible.
Technical activation path	Dropdown	Will always be visible.
Service Class	Dropdown	Will always be visible.
Contract period (months)	Textbox	<u>Enabled</u> : If License Required is set to "True". <u>Disabled</u> : If License Required is set to "False", this textbox will be disabled and the value will be left blank (i.e. unlimited contract duration).
Table column "Language"	Label	Will always be visible.
Table column "Translation"	Textbox	Will always be visible.
Table "MBconnect countries the service is available in"	Checkbox list	Will always be visible.
"Save"	Button	Will always be visible.
"Cancel"	Button	Will always be visible.
"Master Data"	Tab	Will always be visible.
"Authorization Data"	Tab	Will always be visible.
ReadOnlyInfo	Label	<u>Visible</u> : if dialog is in read-only mode <u>Invisible</u> : if dialog is in edit mode
<all buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Enabled</u> : if dialog is in edit mode <u>Disabled</u> : if dialog is in read-only mode

Table 181: Dialog Elements States (DLG\_ServiceDetail)

### 5.1.3 DLG\_ServiceMasterOverview

#### 5.1.3.1 General Description

This dialog is the entry point for maintaining the available service masters. It displays a list of all currently maintained service masters and allows the creation, deletion or editing the available service masters.

Add new Service Master						
Delete					Items per page	Page <a href="#">1</a> <a href="#">2</a> <a href="#">3</a>
	Master Id	Business Area	Category	Name	Opened Session	Action
<input type="checkbox"/>	1	B2C	Standard Services	Accident Recovery		<a href="#">Edit</a>
<input type="checkbox"/>	2	B2C	Standard Services	Breakdown Management		<a href="#">View</a>
<input type="checkbox"/>	3	B2C	Standard Services	Maintenance Management		<a href="#">Edit</a>
<input type="checkbox"/>	4	B2C	Vehicle Set-Up	Remote Retrieval of Vehicle Status		<a href="#">Edit</a>
<input type="checkbox"/>	5	B2C	Vehicle Set-Up	Programming of Auxiliary Heating		<a href="#">Edit</a>
<input type="checkbox"/>	6	B2B	Vehicle Monitoring	Parked Vehicle Locator		<a href="#">Edit</a>

Add new Service Master						
Delete					Items per page	Page <a href="#">1</a> <a href="#">2</a> <a href="#">3</a>
	Master Id	Business Area	Category	Name	Opened Session	Action

Figure 48: DLG\_ServiceMasterOverview

### 5.1.3.2 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		Load all available service masters and sort them by ServiceCategory.sortOrder and then by ServiceMaster.ServiceMasterId together with the permitted actions for each loaded service by calling the loading algorithm described in Chapter "Loading of master data elements".
"Add new Service Master"	Button	Switches to the dialog DLG_ServiceMasterDetail (→ see chapter 5.1.4) to create a new service master. The dialog is opened in new mode.
"Delete"	Button	After displaying a confirmation popup delete the selected service masters by calling the IIF_DeleteServiceMasters (→ see chapter 4.4.3). Display potential error messages in the according message popup.
"Edit"	Button	Switches to the dialog DLG_ServiceMasterDetail (→ see chapter 5.1.4) to edit the service master. The dialog is opened in new mode.
"View"	Button	Switches to the dialog DLG_ServiceMasterDetail (→ see chapter 5.1.4) to view the service master. The dialog is opened in read-only mode.

Table 182: Buttons and functions (DLG\_ServiceMasterOverview)

### 5.1.3.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Table column "MasterId"	Label	The id of the service master.	ServiceMaster.ServiceMasterId
Table column "BusinessArea"	Label	The business area of the service master.	ServiceMaster.BusinessArea
Table column "Category"	Label	The name of the service category the service master is assigned to.	ServiceCategory.name
Table column "Name"	Label	The name of the service master (localized to the user's language).	ServiceMaster.Name
Table Column "Opened Sessions"	Label/Icon	Indicates whether the entity Document is edited inside a change session.	-

Table 183: Form fields and front-end data objects (DLG\_ServiceMasterOverview)

### 5.1.3.4 Dialog field validation

None.

### 5.1.3.5 Configurability (incl setting for roles)

None.

### 5.1.3.6 Dialog Elements State

See states of the common elements displayed by the “Overview” dialogs.

## 5.1.4 DLG\_ServiceMasterDetail

### 5.1.4.1 General Description

This dialog allows the user to view or edit a service master. The dialog includes maintaining the translations of the service master name and the customer or vehicle related mandatory fields as well as the role rights assignment. Additionally, the user can add custom properties for the activation of service(s) assigned to the service master. Furthermore, the service master is assigned to a service category using this dialog.

Note:

- The section “Group fields” describes a dependency between a service master and a set of profile data fields. Contrary to required data fields, a group fields claims that at least one of the included data fields is required in order to activate the linked service(s). Group fields shall not be visible in any user interface, and function more as a virtual as a real profile data field.
- The section “Data fields” describes a dependency between a service master and one single data field or a set of data field. In contrast to groups, data fields are always stored in the user account and are fully visible on the user interface.
- Groups and dependent data field combinations will only be shown for the first and second level.

For a further description of the roles and rights see MBconnect role concept (→ see chapter 2.3.15).

**Master Data | Authorization Data**

Service Master ID*	30
Business Area *	B2C
Service Category*	Standard Services
Activate automatically	<input type="checkbox"/>
Confirmation token needed	<input type="checkbox"/>
Personal verification needed	<input type="checkbox"/>
Trust Level*	3

Service Master Name

Language	Translation
<b>German (Germany)</b>	Wartungsmanagement
German (Austria)	
German (Switzerland)	
<b>English (United Kingdom)</b>	Maintenance Management
English (Switzerland)	
<b>French (France)</b>	Gestion de la maintenance
French (Belgium)	

This service requires the following custom properties (CP) for activation:

cp.FBSHardware  
 cp.Hotelpreference

**Delete selected CP**

Add a new custom property (CP) for activation

**Add CP**

This service requires the following customer information for activation:

Data fields:

E-Mail  
 Accepted Privacy Policy  
 Mobile Phone  
 Landline Phone  
 Newsletter  
 By Email  
 By Phone

Group fields:

Communication Channel  
 - By Email  
 - By Phone  
 - By Letter

This service requires the following vehicle information for activation:

Data fields:

Service Dealer

**Save**   **Cancel**

Figure 49: DLG\_ServiceMasterDetail (Master Data)

Master Data   Authorization Data					
<p>Note: If a service needs a signed user agreement in order to be activated, the service master has to be visible to the master user ("WRITE" right).</p>					
Right \ Role	MBC.MBC_RETAIL	MBC.MBC_SUPPORT	MBC.VEHICLE_MASTERUSER	MBC.VEHICLE_SUBUSER	MBC.DFS
NONE	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
READ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
ACTIVATE	<input type="radio"/>				
DEACTIVATE	<input type="radio"/>				
WRITE	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 50: DLG\_ServiceMasterDetail (Authorization Data)

#### 5.1.4.2 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		<p>Edit, Read only mode:  Load the service master, the custom property assigned to the service master, the category the service is assigned to, and all available translations. Match the available translations against all maintainable translations, which are available in the application configuration. Also check the required customer and vehicle information fields that are listed in ServiceMaster.RequiredCustomerInformation and ServiceMaster.RequiredVehicleInformation. Sort the entries in the list by their lexicographic order. Further, load all rights role assignments by calling IIF_GetRightsForResource with &lt;ResourceTypeEnum&gt; = SERVICEMASTER and &lt;resourceName&gt; = &lt;ServiceMaster&gt;.&lt;ServiceMasterID&gt;.</p> <p>New mode:  Load the maintainable languages from the application configuration and generate the table rows. The service master id will be empty until the service master is saved. Sort the entries in the "Required customer information" and "Required vehicle information" list by their lexicographic order.</p> <p>Further, the roles will be assigned to the following rights:</p> <ul style="list-style-type: none"> <li>A. MBC.MBC_RETAIL: "WRITE"</li> <li>B. MBC.MBC_SUPPORT: "WRITE"</li> <li>C. MBC.MASTERUSER: "WRITE"</li> <li>D. MBC.SUBUSER: "READ"</li> <li>E. MBC.DFS: "NONE"</li> </ul>

		Consider the loading algorithm described in Chapter Loading of master data elements
“Save”	Button	Save the currently maintained service master and navigate back to <b>DLG_ServiceMasterOverview</b> . Call <b>IIF_SaveServiceMaster</b> (→ see chapter 4.4.19) in order to save the service master details. Bring the currently maintained service master in the change session of the user. (see general algorithm <b>ApplyChangeSessionOnElement</b> )
“Cancel”	Button	Discard all changes and navigate back to <b>DLG_ServiceMasterOverview</b>
“Select/Deselect all”	Checkbox	Selects or deselects all check boxes in the section.
“Master Data”	Tab	Shows the elements of the tab “Master Data”.
“Authorization Data”	Tab	Shows the elements of the tab “Authorization Data”.
“Add CP”	Button	<u>Only enabled</u> if the textbox “New custom property” is not empty.  Save the custom property defined in the textbox “New custom property” in the list of custom properties assigned to the service. Reload the table “Required Custom Properties”.
“Delete selected CP”	Button	<u>Only enabled</u> if the table “Required Custom Properties” is not empty and one item in this table is selected.  After displaying a confirmation popup delete the selected Custom Property.

Table 184: Buttons and functions (DLG\_ServiceMasterDetail)

#### 5.1.4.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Service Master Id	Textbox	The id of the service master. The id has to be unique and set manually.	ServiceMaster.serviceMasterId  Available values: All positive integers that are not referenced yet.
Business Area	Dropdown	The business area of the service master. If the business area is set to “B2C” then the services assigned to this service master are available for regular customers. If the business area is set to “B2B” then the services assigned to this service master are available for business partners.	ServiceMaster.BusinessArea
Service Category	Dropdown	The localized name (language of the user) of the service category this service (service master) is assigned to,	ServiceCategory.name

Activate Automatically	Checkbox	States if the service is activated automatically when the vehicle is registered for a customer.  Default: TRUE	ServiceMaster.activateAutomatically
Confirmation token needed	Checkbox	The information whether the customer has to enter a secret code in order to use service(s) assigned to the servicemaster.	ServiceMaster.confirmationTokenNeeded
Personal verification needed	Checkbox	The information whether the customer has to be verified in order to use service(s) assigned to the servicemaster.	ServiceMaster.personalVerificationNeeded
Trust Level	Dropdown	The minimal trust level of a vehicle registration in order to use service(s) assigned to the servicemaster.  Default: "3"	<u>Selected value:</u> ServiceMaster.trustLevel  <u>Available values:</u> "0", "1", "2", "3"
Table column "Language"	Label	Locale of the maintained translation.	-
Table column "Translation"	Textbox	The name of the service master (localized).	ServiceMaster.Name
Table "Required Custom Properties"	Listbox	Optionally defines the custom properties that are needed to activate the service(s) linked to this service master.	ServiceMaster.customProperty
New custom property	Textbox	Optionally define a new custom property that is needed to activate the service(s) linked to this service master.	-
Table "Required customer information (data fields)"	Checkbox list	Optionally defines customer information that is needed to activate the service(s) linked to this service master.	<u>Displayed value:</u> datafieldID from IIF_GetProfileDataFieldsAndGroups  Show childrenIDs associated with that datafieldID as indented listing without checkboxes.  <u>Checked values:</u> All matching entries contained in ServiceMaster.RequiredCustomerInformation.  <u>Available values:</u> datafieldID from IIF_GetProfileDataFieldsAndGroups where fieldOwnerType is ACCOUNT and marketSpecific is false.
Table "Required customer information (group fields)"	Checkbox list	Optionally defines customer information that is needed to activate the service(s) linked to this service master.	<u>Displayed value:</u> groupID from IIF_GetProfileDataFieldsAndGroups.  Show childrenIDs associated with that groupID as indented listing without checkboxes.  <u>Checked values:</u> All matching entries contained in ServiceMaster.RequiredCustomerInformation.  <u>Available values:</u> groupID from

			IIF_GetProfileDataFieldsAndGroups
Table "Required vehicle information (data fields)"	Checkbox list	<p>Optionally defines vehicle information that is needed to activate the service(s) linked to this service master.</p>	<p><u>Displayed value:</u> datafieldID from IIF_GetProfileDataFieldsAndGroups</p> <p>Show childrenIDs associated with that datafieldID as indented listing without checkboxes.</p> <p><u>Checked values:</u> All matching entries contained in Service-Master.RequiredCustomerInformation.</p> <p><u>Available values:</u> datafieldID from IIF_GetProfileDataFieldsAndGroups where fieldOwnerType is VEHICLE and marketSpecific is false.</p>
Table "Right \Role"	Radiobutton matrix	<p>Defines the right of each role determining what the representative of that role is allowed to do with the service(s) linked to this service master.</p> <p>Default: By default, all roles are assigned to the highest right.</p>	<p>Set all rights as determined by IIF_GetRightsForResource with &lt;ResourceTypeEnum&gt; = SERVICEMASTER and resourceName = ServiceMaster.ServiceMasterID.</p>
ReadOnlyInfo	Label	Presents this information: "The dialog is opened in read-only mode."	-

Table 185: Form fields and front-end data objects (DLG\_ServiceMasterDetail)

#### 5.1.4.4 Dialog field validation

"Service Master ID", "Service Category", and "Trust Level" cannot be left blank.

#### 5.1.4.5 Configurability (incl setting for roles)

None.

### 5.1.4.6 Dialog Elements State

Linked Label	Type	State Description
Service Master Id	Textbox	Will be visible when tab "Master Data" is selected.
Service Category	Dropdown	Will be visible when tab "Master Data" is selected.
Activate Automatically	Checkbox	Will be visible when tab "Master Data" is selected
Confirmation token needed	Checkbox	Will be visible when tab "Master Data" is selected.
Personal verification needed	Checkbox	Will be visible when tab "Master Data" is selected.
Trust Level	Dropdown	Will be visible when tab "Master Data" is selected.
Table column "Language"	Label	Will be visible when tab "Master Data" is selected.
Table column "Translation"	Textbox	Will be visible when tab "Master Data" is selected.
Table "Required custom properties"	Listbox	Will be visible when tab "Master Data" is selected.
"Delete selected CP"	Button	Will be visible when tab "Master Data" is selected.
New custom property	Textbox	Will be visible when tab "Master Data" is selected.
"Add CP"	Button	Will be visible when tab "Master Data" is selected.
Table "Required customer information (data fields)"	Checkbox list	Will be visible when tab "Master Data" is selected.
Table "Required customer information (group fields)"	Checkbox list	Will be visible when tab "Master Data" is selected.
Table "Required vehicle information (data fields)"	Checkbox list	Will be visible when tab "Master Data" is selected.
Table "Right \Role"	Radiobutton matrix	Will be visible when tab "Authorization Data" is selected.
"Save"	Button	Will always be visible.
"Cancel"	Button	Will always be visible.
"Master Data"	Tab	Will always be visible.
"Authorization Data"	Tab	Will always be visible.
ReadOnlyInfo	Label	<u>Visible:</u> if dialog is in read-only mode <u>Invisible:</u> if dialog is in edit mode
<all buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Enabled:</u> if dialog is in edit mode <u>Disabled:</u> if dialog is in read-only mode

Table 186: Dialog Elements State (DLG\_ServiceMasterDetail)

### 5.1.5 DLG\_DocumentOverview

This dialog provides an overview over the maintained documents.

From the column "Preview" the user can visualize the content of a document.

Add new Document				
Items per page <input type="button" value="10"/> Page <a href="#">1</a> <a href="#">2</a> <a href="#">3</a>				
Document	Opened Sessions	Action	Preview	
User Agreement - Basic Services		<input type="button" value="Edit"/>	<input type="button" value="Preview"/>	
User Agreement - LiveTraffic		<input type="button" value="View"/>	<input type="button" value="Preview"/>	
User Agreement - Remote		<input type="button" value="Edit"/>	<input type="button" value="Preview"/>	
Terms and Conditions		<input type="button" value="Edit"/>	<input type="button" value="Preview"/>	

Items per page  Page [1](#) [2](#) [3](#)

Add new Document

Figure 51: DLG\_DocumentOverview

### 5.1.5.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load available documents and list them in their lexicographic order together with the permitted actions for each loaded document by calling the loading algorithm described in Chapter <b>Loading of master data elements</b> (see details in subchapter “Dialog Elements States”)
“Add new Document”	Button	Navigates to the dialog DLG_DocumentDefinition to add a new document and maintain their versions. The dialog is opened in new mode.
“Edit” (1..n)	Button	Navigates to the dialog DLG_DocumentDefinition to edit the document definition. The dialog is opened in edit mode.
“View” (1..n)	Button	Navigates to the dialog DLG_DocumentDefinition to visualize the document definition. The dialog is opened in read-only mode.
“Preview” (1..n)	Button	Navigates to dialog DLG_DocumentPreview

### 5.1.5.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Table column “Document”	Label	The name of the document.	All maintained Documents, with Document.documentID being displayed as the name
Table Column “Opened Sessions”	Label/Icon	Indicates whether the entity Document is edited inside a change session.	-

Table 187: Form fields and front-end data objects

### 5.1.5.3 Dialog field validation

None.

---

#### **5.1.5.4 Configurability**

None.

#### **5.1.5.5 Dialog Elements States**

See states of the common elements displayed by the “Overview”-dialoges.

### **5.1.6 DLG\_DocumentDefinition**

This dialog allows the user to view or edit document definitions, as well as create a new document definition.

Depending on whether the “is User Agreement” checkbox is selected, the bottom panel is either the user agreement specific one or the one for general documents. The panel differences are illustrated in figures below.

#### ***Enabling User Agreements:***

Before a new user agreement version can be used by the system and thus be available to the customers, it has to be enabled first via the button “Enable”. This will freeze the dates and prevent them from being altered. As long as the user agreement is not enabled, it will not be activated and will have no effect on the customers. Enabled user agreements do have to be valid, which means that service will only be covered by one enabled user agreement at a time. Additionally, the countries for which the user agreement is available have to be selected.

A validity check will be performed when an enabled user agreement will be saved.

#### ***Document titles for User Agreements:***

Document titles can be provided if the document is a user agreement. All default languages are marked bold. The available languages (and defaults) are configured in the application configuration (→ see PROP\_LOCALES)

#### ***Summary of User Agreement changes:***

When a user agreement gets updated to a newer version, a summary of the changes done must be provided. This is done by referencing a document block together with the appropriate version (see section called “Summary of User Agreement change”).

#### ***Validity of versions:***

All documents (that are not a user agreement) only have one valid version.

User agreements are special, in that they can have one version that is still valid for existing customers, while a newer version of the user agreement is already valid for new customers. While this is true for the terms and conditions as well, they are referenced as a document template from within the user agreement. It is therefore necessary to create a new version of user agreements for a new version of terms and conditions.

Document	User Agreement - Basic Services												
Version	2												
DocType	A PDF												
<input checked="" type="checkbox"/> is User Agreement													
<b>User Agreement Detail</b>													
Is Enabled	yes												
<input type="button" value="Enable"/>													
Valid from (new customer)	01.01.2014												
Valid from (existing customer)	01.03.2014												
Info Email	15.01.2014												
The document is available in the following MBconnect countries:*													
<input type="checkbox"/> Select/Deselect all													
<input checked="" type="checkbox"/> Austria (AT)													
<input checked="" type="checkbox"/> Belgium (BE)													
<input type="checkbox"/> Denmark (DK)													
<input checked="" type="checkbox"/> France (FR)													
<input checked="" type="checkbox"/> Germany (DE)													
<input type="checkbox"/> Hungary (HU)													
<input checked="" type="checkbox"/> Italy (IT)													
<input checked="" type="checkbox"/> Luxembourg (LU)													
<input checked="" type="checkbox"/> Netherlands (NL)													
<input type="checkbox"/> Poland (PL)													
<input checked="" type="checkbox"/> Slovakia (SK)													
<input type="checkbox"/> Spain (ES)													
<input type="checkbox"/> Switzerland (CH)													
<input checked="" type="checkbox"/> Czech Republic (CZ)													
<input type="checkbox"/> United Kingdom (UK)													
<b>Summary of User Agreement Change</b>													
Legal Changes Block Name	UA_BasicServices_Changed_Block												
Legal Changes Block Version	2												
<table border="1"> <thead> <tr> <th>Language</th> <th>Translation</th> </tr> </thead> <tbody> <tr> <td>German (Germany)</td> <td>Basisdienste</td> </tr> <tr> <td>German (Austria)</td> <td>Standarddienste</td> </tr> <tr> <td>German (Switzerland)</td> <td></td> </tr> <tr> <td>English (United Kingdom)</td> <td>Basic Services</td> </tr> <tr> <td>English (United States)</td> <td></td> </tr> </tbody> </table>		Language	Translation	German (Germany)	Basisdienste	German (Austria)	Standarddienste	German (Switzerland)		English (United Kingdom)	Basic Services	English (United States)	
Language	Translation												
German (Germany)	Basisdienste												
German (Austria)	Standarddienste												
German (Switzerland)													
English (United Kingdom)	Basic Services												
English (United States)													
<b>XML</b>													
<...>													
<input type="button" value="New Version"/> <span style="float: right;"><input type="button" value="Save"/></span>													

Figure 52: DLG\_DocumentDefinition-UserAgreement

Document: Vehicle Separation Information

Version: 2

DocType: PDF

Country: ALL

is User Agreement

General Detail

Valid from: 01.01.2014

XML

```
<...>
```

New Version      Save

Figure 53: DLG\_DocumentDefinition for non user agreements

### 5.1.6.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Edit Mode, Read-Only Mode: Load the available instances of VersionedDocument and fill in the content by calling AF_RetrieveVersionedDocumentElementsForUser. New Mode: Load available DocTypes and set the corresponding layout.
"Enable"	Button	Sets the user agreement to the enabled state. This freezes the dates and prevents them from being altered. The button itself as well as each date widget is then disabled. Call AF_EnableUserAgreementCheck (see 5.5.6) for further validations.
"New Version"	Button	If the document is a User Agreement and the validFromExistingCustomer of the actual latest version is greater than today abort with SERMGM_013 else:  Create a new version of the document definition by incrementing the version number.  The following attributes are extracted from the previous latest version of a document and used to initialize the new created version: <ul style="list-style-type: none"> <li>- Document.docType</li> <li>- Document.isUserAgreement</li> <li>- VersionedDocument.&lt;&lt;i18N&gt;&gt;title</li> <li>- VersionedDocument.documentXMLDefinition</li> <li>- If the versioned document is a user agreement, then also consider attribute: <ul style="list-style-type: none"> <li>o UserAgreementLegalChanges.legalChangesBlockName</li> <li>o List of countries where the user agreement is already available in</li> </ul> </li> </ul> Bring the new versioned document together with its parent document (in the change session of the user. (see general algorithm <b>ApplyChangeSessionOnElement</b> )

---

<b>Linked label / button labeling</b>	<b>Type</b>	<b>Action description</b>
“Save”	Button	<p>Save the currently maintained document and its versioned documents and navigate back to DLG_DocumentOverview. The XML is validated against the DocType specific XSD to insure the structural validity of the XML. Therefore AF_SaveDocument is called.</p> <p>Bring the currently maintained document (for which an available version is being edited) in the change session of the user. ( see general algorithm ApplyChangeSessionOnElement)</p>

## 5.1.6.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Document	Textbox	The unique name of the document.	Document.documentID
DocType	Drop-Down	The document type.  Default: “Plaintext”	<u>Selected value:</u> Document.docType  <u>Available values:</u> 1. PDF: Document type for PDF files 2. TXT: Document type for text files
Is User Agreement	Checkbox	Classification if the document is a user agreement.	Document.isUserAgreement
Country	Label	“ALL” is provided in order to indicate that the document (not user agreement!) is valid for all MBconnect countries.	-
The user agreement is available in the following MBconnect countries.*	List of Checkboxes	Specifies the MBconnect countries for which the document is valid	<u>Available values:</u> all countries enabled for MBconnect: call IIF_GetMBconnectCountries
Table column “Version”	Combobox	The version.	<u>Available:</u> Edit Mode: All instances of VersionedDocument that belong to the same Document.  <u>Read-Only mode:</u> Load only the available instances of the VersionedDocument that are not being edited inside the change session of other users  <u>Selected:</u> VersionedDocument.versionID
Is Enabled	Label	Shows whether the user agreement is enabled. Only visible for user agreements.	UserAgreement.enabled
Valid from	Textbox	Specifies the start date for the validity of this document.  This field is used on documents which are not marked as user agreements.	VersionedDocument.validFrom
Valid from (new customer)	Textbox	Specifies when this document is valid for new customers. Only visible for user agreements.  This field is used on user agreements.	VersionedDocument.validFrom
Valid from (existing customer)	Textbox	Specifies when this document is valid for old customers. Only visible for user agreements.  This field is used on user agreements.	UserAgreement.validFromForExistingCustomer
Info Email	Textbox	The date for sending an info email to inform about new legal documents. Only visible for user agreements.	UserAgreement.informByEMailDate
Table Column “Translation”	Textbox	The translations of the versioned document. Only visible for user agreements.	VersionedDocument.title
XML	Textbox	The document definition in XML that specifies how the document is generated.	DocumentXmlDefinition
ReadOnlyInfo	Label	Presents this information: “The dialog is opened in read-only mode.”	-
Legal Changes	Combobox	the name of the document block/Default = null	<u>Selected value:</u> DocumentBlock.blockName

Block Name			<p><u>Available values:</u> all document blocks that are of type: "UA_CHANGE_SUMMARY". Note: The combobox is only displayed if "Is User Agreement" is checked and if the version of the user agreement is &gt; 1 (VersionedDocument.version &gt; 1).</p> <p><u>Edit Mode:</u> Load all instances of DocumentBlock that are not being edited inside the change session of other users</p>
Legal Changes Block Version	Combobox	The version of the document block/ Max(VersionedDocumentBlock.versionID)	<p><u>Selected value:</u> VersionedDocumentBlock.versionID</p> <p><u>Available values:</u> Loads all versions of the selected document block (see "Legal Changes Block Name" combobox). Note: The combobox is only displayed if "Is User Agreement" is checked and if the version of the user agreement is &gt; 1 (VersionedDocument.version &gt; 1).</p> <p><u>Edit Mode:</u> Load all instances of VersionedDocumentBlock that are associated to the selected DocumentBlock and are not being edited inside the change session of other users</p>

Table 188: Form fields and front-end data objects

### 5.1.6.3 Dialog field validation

Linked Field	Validation	Error Message
Valid from (existing customer)	Date must be greater than today. Date must be greater than or equal to Valid from (new customer). Date must be greater than Info Email.	SERMGM_016 SERMGM_018 SERMGM_022
Valid from (new customer)	Date must be greater than today. Date must be greater than Info Email. Date must be smaller than or equal to Valid from (existing customer).	SERMGM_016 SERMGM_017 SERMGM_021
Valid from	Date must be greater than today.	SERMGM_016
Info Letter	Date must be greater than today. Date must be smaller than Info Email.	DOCMAS_001 DOCMAS_017
Info Email	Date must be greater than today. Date must be smaller than Valid from (existing customer). Date must be greater than or equal to Valid from (new customer).	SERMGM_016 SERMGM_019 SERMGM_020

### 5.1.6.4 Configurability

None.

### 5.1.6.5 Dialog Elements States

Linked Label	Type	State Description
Is Enabled	Label	Will only be shown if "Is User Agreement" is checked.
"Enabled"	Button	Will only be shown if "Is User Agreement" is checked.

---

		<u>Disabled:</u> The button is disabled if the user agreement is already enabled.
Valid from (new customer)	Textbox	Will only be shown if "Is User Agreement" is checked.  <u>Disabled:</u> The Textbox is disabled if the dates are already fixed.
Valid from (existing customer)	Textbox	Will only be shown if "Is User Agreement" is checked.  <u>Disabled:</u> The Textbox is disabled if the dates are already fixed.
Info Email	Textbox	Will only be shown if "Is User Agreement" is checked.  <u>Disabled:</u> The Textbox is disabled if the dates are already fixed.
Table Column "Translation"	Textbox	Will only be shown if "Is User Agreement" is checked.
Valid from	Textbox	Will only be shown if "Is User Agreement" is NOT checked.
ReadOnlyInfo	Label	<u>Visible:</u> if dialog is in read-only mode <u>Invisible:</u> if dialog is in edit mode
New Version	Button	<u>Enabled:</u> if the selected <VersionedDocument> has the maximum version. <u>Disabled:</u> otherwise
<all buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Enabled:</u> if dialog is in edit mode <u>Disabled:</u> if dialog is in read-only mode
Document Block	Combobox	Will only be shown if "Is User Agreement" is checked and if the version of the user agreement > 1.
Document Block Version	Combobox	Will only be shown if "Is User Agreement" is checked and if the version of the user agreement > 1.
Country	Label	Will only be shown if "Is User Agreement" is not checked
List of Countries	List of Checkboxes	Will only be shown if "Is User Agreement" is checked.  <u>Disabled:</u> - The Checkboxes are disabled if dialog is in read-only mode. - The Checkboxes are disabled for all previously enabled versions (all versions < max(version))

### 5.1.7 DLG\_UserAgreementServiceAssignmentOverview

This dialogue provides an overview of all user agreements and allows the user to view or edit the assignment between user agreements and services.

			Items per page	10	▼	Page	<a href="#">1</a>	<a href="#">2</a>	<a href="#">3</a>
Document		Open sessions	Actions						
User Agreement - Basic Services			<a href="#">Edit</a>						
User Agreement - LiveTraffic			<a href="#">View</a>						
User Agreement - Remote			<a href="#">Edit</a>						

Items per page [10](#) ▼ Page [1](#) [2](#) [3](#)

Figure 54: DLG\_UserAgreementServiceAssignmentOverview

#### 5.1.7.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all available user agreements together with the permitted actions for each loaded document by calling the loading algorithm described in Chapter “Loading of master data elements”. List them in their alphabetically order.
“Edit” (1..n)	Button	Switches to the dialog DLG_UserAgreementServiceAssignment to view or edit the selected assignment. The dialog is opened in edit mode.
“View” (1..n)	Button	Switches to the dialog DLG_UserAgreementServiceAssignment (view mode) to view the selected assignment. The dialog is opened in read-only mode.

#### 5.1.7.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Table column “User Agreement”	Label	The user agreement document (only show documents where Document.isUserAgreement is true).	Document.documentID
Table Column “Opened Sessions”	Label/Icon	Indicates whether the user agreement service assignment is edited inside a change session.	-

#### 5.1.7.3 Dialog field validation

No changes.

#### 5.1.7.4 Configurability (incl. setting for roles)

None.

### 5.1.7.5 Dialog Elements States

See the states of the common elements displayed by the “overview”-dialogues.

### 5.1.8 DLG\_UserAgreementServiceAssignment

This dialog allows the association between specific, versioned user agreements and services that they cover.

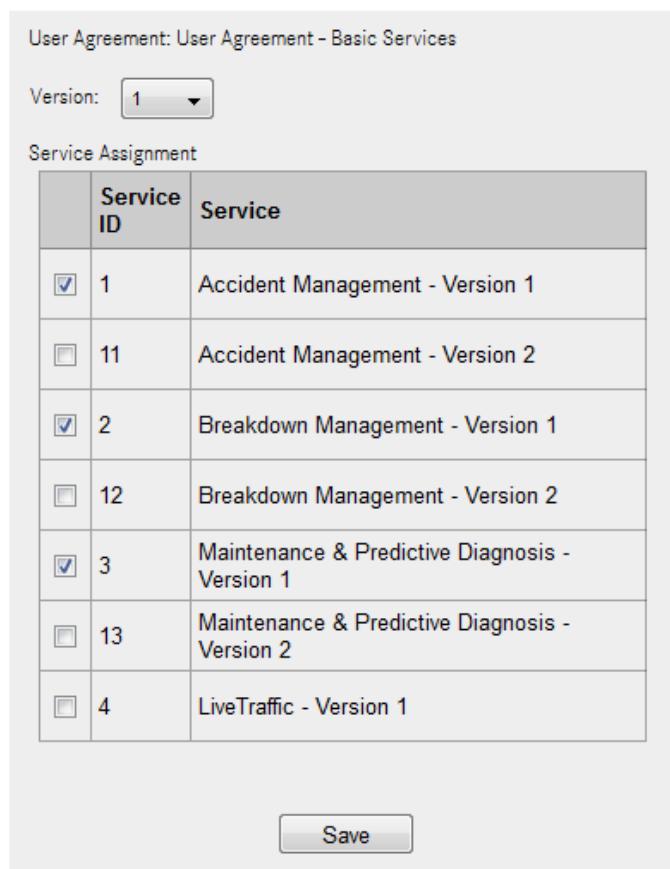


Figure 55: DLG\_UserAgreementServiceAssignment

#### 5.1.8.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		For the requested user agreement load all available versions and user agreement service assignments and order all available services alphabetically. Only show services that a Mbconnect master user (MBC.VEHICLE_MASTERUSER) can see ("WRITE" right). To determine the rights of the roles, call IIF.GetServiceRightsByRoles with <roles> = MBC.VEHICLE_MASTERUSER and <serviceIDs> = <Service>.<serviceID> for each of the given services.
"Save"	Button	Saves the UserAgreementServiceAssignment entity that describes the association between the selected services and the user agreement in the selected version. Calls AF_SaveUserAgreementServiceAssignment (→ see chapter 5.5.8). In case of an error, show the error description and cancel the action. Bring the currently maintained assignment in the change session of the user (see general algorithm ApplyChangeSessionOnElement).

### 5.1.8.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
User Agreement	Label	The user agreement document (only show documents where Document.isUserAgreement is true).	Document.documentID
Version	Drop-Down	The version of the user agreement.	UserAgreement.versionID
Table column "Service"	Label	The services that the user agreement can be associated with.	<u>Displayed value:</u> ServiceMaster.name + " – Version " + Service.versionNumber  <u>Available values:</u> All available Services
Table column "Service Id"	Label	The ID of the service.	Service.ServiceId
ReadOnlyInfo	Label	Presents this information: "The dialog is opened in read-only mode."	-

### 5.1.8.3 Dialog field validation

None.

### 5.1.8.4 Configurability

None.

### 5.1.8.5 Dialog Elements States

Linked Label	Type	State Description
ReadOnlyInfo	Label	<u>Visible</u> : if dialog is in read-only mode <u>Invisible</u> : if dialog is in edit mode
<all checkboxes>	Checkboxes	<u>Enabled</u> : if dialog is in edit mode <u>Disabled</u> : if dialog is in read-only mode

## 5.2 External View - Offered Interfaces

### 5.2.1 IF\_SOE\_GetUserAgreementServiceAssignment

This interface provides “user agreement service assignment” master data. It returns for each given user agreement id with given user agreement version the serviceID of each covered MBconnect service.

Internally, the AF\_GetUserAgreementServiceAssignment is called to determine the services.

If there is no service found for a certain user agreement, an empty list is returned for this user agreement.

#### 5.2.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of user agreements (1..*)</b>						
uaID	Mand.	String	-	Document.documentID	Examples: 1, 12356, 1234567890	The id of the user agreement
uaVersion	Mand.	Int	10	Document.VersionID	Examples: 1, 12356,	The version of the user agreement.

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
					1234567890	

Table 189: Input parameters

### 5.2.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of user agreements and assigned services (each of the following parameters exist per ualID) (1..*)</b>						
ualID	Mand.	String	-	Document.DocumentID	Examples: 1, 12356, 1234567890	The id of the user agreement
uaVersion	Mand.	Int	10	Document.VersionID	Examples: 1, 12356, 1234567890	The version of the user agreement.
serviceIDList	Mand.	List of Strings	-	List of Service.ServiceID	-	The id that identifies the assigned telematics service. Can be empty.

Table 190: Output parameters

### 5.2.1.3 Exceptions

None.

## 5.3 External View - Consumed Interfaces

None.

## 5.4 Internal View - Offered Interfaces

### 5.4.1 IIF\_GetUserAgreementServiceAssignments

Internally calls AF.GetUserAgreementServiceAssignments (→ see chapter 5.5.1) to retrieve all existing UserAgreementServiceAssignment entities.

### 5.4.2 IIF\_UpdateUserAgreementServiceAssignments

Internally calls AF\_UpdateUserAgreementServiceAssignments (→ see chapter 5.5.2) to update all UserAgreementServiceAssignment entities.

### 5.4.3 IIF\_GetServicesByUserAgreements

For each user agreement in the given list of user agreements this internal interface retrieves the MBconnect services which are covered by the user agreement. If there is no service found for a certain user agreement ID, an empty list is returned for this user agreement.

Internally calls AF\_GetServicesByUserAgreements (→ section 5.5.9) to retrieve the available services.

---

## 5.5 Implementation

### 5.5.1 AF\_GetUserServiceAssignments

#### 5.5.1.1 General Description

This application function provides all existing User Agreement Service Assignment entities.

#### 5.5.1.2 Sequence Description

Determine all instances of entity <UserAgreementServiceAssignment> and information from related entities that are available in SOE and return the data.

#### 5.5.1.3 Input

None.

#### 5.5.1.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;UserAgreementServiceAssignment&gt;: List of all UserAgreementServiceAssignment entities</b>		
DocumentID	Document.documentID	The DocumentID of a Document.
VersionID	VersionedDocument.versionID	The VersionID of a VersionedDocument.
- Inner List of <Service>		
ServiceID	Service.ServiceID	The ServiceID of a Service.

Table 191: AF\_GetUserServiceAssignments output

#### 5.5.1.5 Exceptions

None.

### 5.5.2 AF\_UpdateUserServiceAssignments

#### 5.5.2.1 General Description

This application function updates the UserAgreementServiceAssignment entities.

#### 5.5.2.2 Sequence Description

Compare the <UserAgreementServiceAssignment> given as input parameter with the existing <UserAgreementServiceAssignment> in SOE retrieved by calling AF\_GetUserServiceAssignments.

##### If the updateMode = ADD

Add all instances of <UserAgreementServiceAssignment> which are given as input parameter and not parts of the existing <UserAgreementServiceAssignment>.

##### If the updateMode = UPDATE

Update all instances of <UserAgreementServiceAssignment> that are available in SOE which are not completely equal to the <UserAgreementServiceAssignment> given as input parameter.

##### If the updateMode = DELETE

---

Delete all instances of <UserAgreementServiceAssignment> that are available in SOE which are not parts of the <UserAgreementServiceAssignment> given as input parameter.

### 5.5.2.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the user agreement service assignment entities.
<b>List&lt;UserAgreementServiceAssignment&gt;: List of all UserAgreementServiceAssignment entities</b>		
DocumentID	Document.documentID	The DocumentID of a Document.
VersionID	VersionedDocument.versionID	The VersionID of a VersionedDocument.
- Inner List of <Service>		
ServiceID	Service.ServiceID	The ServiceID of a Service.

Table 192: AF\_UpdateUserAgreementServiceAssignments input

### 5.5.2.4 Output

None.

### 5.5.2.5 Exceptions

None.

## 5.5.3 AF\_SaveService

This application function ensures that the service is available in all countries where the user agreements referencing this service are also available. If the validation is successful, the service information is saved.

### 5.5.3.1 Sequence Description

#### **Step 1: Validate Service Availability in Countries**

Ensure that the service is available in all countries where the user agreements referencing this service are also available: call AF\_CheckCountryAvailabilityForService.  
If the call returns with an exception, abort execution.

#### **Step 2: Save service information**

Update all service related data provided by the given instance of <Service>. If no entry for this instance exists, create it: call AF\_SaveServiceDetail.

### 5.5.3.2 Input

Name	Type / Length / BOM	Description	Changes
Service	Service	The service that has to be saved.	Unchanged
<b>List of MBconnect countries the service is available in (1..*)</b>			
Country	MbcCountry	The MBconnect countries the service is available in.	Unchanged

Table 193: AF\_SaveService Input

### 5.5.3.3 Output

None.

### 5.5.3.4 Exceptions

- If the service is not available in countries in which the user agreements are available the error SERMGM\_001 will be returned.

### 5.5.4 AF\_DeleteServices

This AF deletes the given services if they are disabled and not used anymore by any ServiceAssignmentRule, UserAgreement or Contract.

#### 5.5.4.1 Sequence Description

Loop for all given services:

##### **Verify if service is already inside a change session**

- retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown executions stops here.
- verify if service is *already inside a change session of another user*. if yes, throw error SRVSESSION\_001. Execution stops here.

If at least one service is enabled (Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today), abort with the error message SERMGM\_008, with <Service.Name> as placeholder 1.

##### **Duplicate check**

If at least one service is still being referenced by one or more ServiceAssignmentRules or UserAgreements abort with the error message SERMGM\_006.

- Use Service.Name as placeholder 1.
- Optional: As placeholder 2 use a list of all ServiceAssignmentRule.ServiceAssignmentRuleId which reference to the Service.
- Optional: As placeholder 3 use a list of all UserAgreement.Title which reference to the Service.

If at least one service is still being referenced by one or more Contracts, abort with the error message SERMGM\_007. Use <Service.Name> as placeholder 1.

##### **Mark object as deleted**

- Bring the service in the change session of the user (see general algorithm ApplyChangesSessionOnElement).
- Mark the service as deleted (ChangeOperation.DELETE).

#### 5.5.4.2 Input

Name	Type / Length / BOM	Description
Services	List of <Service>	The services to be deleted.

Table 194: Application function input

---

#### **5.5.4.3 Output**

None.

#### **5.5.4.4 Exceptions**

Error message SERMGM\_006/ SERMGM\_007 in case one of the services to delete is still referenced by other entities.

Error message SERMGM\_008 in case one of the services to delete is still enabled.

### **5.5.5 AF\_SaveDocument**

This AF saves the document. In case of an enabled user agreement, the assignment between the user agreement and its services will be validated in order to inhibit the following scenarios:

- Assignment of a service to more than one user agreement in the same validity period
- Availability of a user agreement in countries in which the services covered by the user agreement are not available.

#### **5.5.5.1 Sequence Description**

##### **Step 1: Check XML and save document definition**

Call Documents.AF\_SaveDocumentDefinition (see chapter **6.5.27**)

**Step 2 Check assignment validity** If the document represents a user agreement which is enabled call AF\_EnableUserAgreementCheck (5.5.6).

#### **5.5.5.2 Input**

Name	Type / Length / BOM	Description
Document	DocumentDefinition	The document and its version.
<b>List of MBconnect countries where the user agreement is available in (0..*)</b>		
Country	MBcCountry	Country

#### **5.5.5.3 Output**

None.

#### **5.5.5.4 Exceptions**

- If the assignment is not valid because services of the user agreement are assigned to other user agreements in the same period of time, the error SERMGM\_004 will be returned.
- If the user agreement is available in countries in which the services covered by the user agreement are not available the error SERMGM\_003 will be returned
- If the user agreement has been withdrawn from an existing country the error DOCMAS\_012 will be returned.

### **5.5.6 AF\_EnableUserAgreementCheck**

This AF performs several validations that need to be fulfilled by an enabled user agreement. The following scenarios are verified:

- 
- Assignment of a service to more than one user agreement in the same validity period
  - Availability of a user agreement in countries in which the services covered by the user agreement are not available.

### 5.5.6.1 Sequence Description

#### **Step 1: Retrieve the services assigned to the user agreement**

Retrieve the services assigned to the user agreement.

#### **Step 2: Check assignment validity**

*Step 2.1 Check if services of the user agreement are assigned to other user agreements in the same period of time:*

Call AF\_CheckUserServiceAssignment with the user agreement and the services retrieved in step 1. If the AF returns with valid == false, throw exception SERMGM\_004 with the conflicting user agreements and the conflicting service / services. Otherwise, proceed.

*Step 2.2 Check if the user agreement is available in countries in which the services covered by the user agreement are not available:*

For all services retrieved in step 1, get the unique list of countries in which the services assigned to the user agreement are available, into set A.

If the list of countries in which the user agreement is available is not included set A, throw exception SERMGM\_003.

### 5.5.6.2 Input

Name	Type / Length / BOM	Description
userAgreement	UserAgreement	The user agreement that is/gets enabled.
<b>List of MBconnect countries where the user agreement is available in (0..*)</b>		
Country	MBcCountry	Country

### 5.5.6.3 Output

None.

### 5.5.6.4 Exceptions

- If the assignment is not valid because services of the user agreement are assigned to other user agreements in the same period of time, the error SERMGM\_004 will be returned.
- If the user agreement is available in countries in which the services covered by the user agreement are not available the error SERMGM\_003 will be returned

## 5.5.7 AF\_CheckUserServiceAssignment

This AF checks an assignment between a user agreement and certain services, which either has a productive form or is in the change session of the requesting user.

---

Please note:

- The check if a master user has sufficient rights for a service (“WRITE” right), will be performed at the initialization of the dialog. The administrator can only assign services to a user agreement which pass this validation.

### 5.5.7.1 Sequence Description

#### **Step 1: Determination of Validity Period**

Retrieve all enabled instances of <UserAgreements> in their productive form or in the change session of the requesting user:

Call IIF\_GetUserAgreementsByDate to get all available user agreements, with the input:

- Smallest Java Date value as the Valid From Date
- ChangeSession of the user

For all user agreements for which the condition <UserAgreement.documentID> does match the documentID of the user agreement (input parameter) is fulfilled, determine the next enabled version after the version of the user agreement and remember its validFromNewCustomer date as validTo date. If there is not such a user agreement, remember “infinity” as validTo.

#### **Step 2: Determination of User Agreements inside Validity Period**

Select all user agreements for which the following condition matches:

- 1) <UserAgreement.documentID> does not match the documentID of the user agreement
- 2) reference at least one of the serviceId that are passed as input parameter <Services> AND the corresponding UserAgreementServiceAssignment entity has productive form or is in the change session of the requesting user.

From all these instances selected above, discard the user agreements for which the validFromNewCustomer date is bigger than the validTo date determined in step 1. For all other user agreements, determine its validTo date as described in step 1. Also discard all user agreements which validTo date is smaller than the validFromNewCustomer date from the committed user agreement. The remaining user agreements represent the relevant user agreements inside the validity period of the committed user agreement.

#### **Step 3: Check Validations of Assignment**

For each user agreement determined in step 2, check if one of the assigned services is also selected for the user agreement. If a match can be found, remember the name of the service as well as the user agreement and continue.

#### **Step 4: Return Validity**

If the check in step 3 is performed with one or more matches, return the first conflicting user agreement with a list of the conflicting services and valid = false.

If the check in step 3 is performed without any matches, return with valid = true.

### 5.5.7.2 Input

Name	Type / Length / BOM	Description	Changed
------	---------------------	-------------	---------

---

UserAgreement	User Agree- ment	The user agreement and its version.	Unchanged
Services	List of Ser- vices	The assigned services	New

### 5.5.7.3 Output

Name	Type / Length / BOM	Description
Valid	Boolean	Determines if the assignment is valid or not.
<b>Conflicts (optional)</b>		
UserAgreement	User Agreement	The user agreement and its version.
Services	List of Services	The conflicted services

### 5.5.7.4 Exceptions

None.

## 5.5.8 AF\_SaveUserServiceAssignment

This AF saves the UserAgreementServiceAssignment between a user agreement and certain services.

The assignment between the user agreement and its services will be validated in order to exclude the following scenarios:

- Assignment of a service to more than one user agreement in the same validity period
- Availability of a user agreement in countries in which the services covered by the user agreement are not available.

### 5.5.8.1 Sequence Description

#### **Verify if the equipment is already inside a change session**

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown executions stops here.
- Verify if the UserAgreementServiceAssignment is *already inside a change session of another user*: if yes, throw error DOCSESSION\_001. Execution stops here.

#### **Step 1: Check validy**

Step 1.1: Check if services of the user agreement are assigned to other user agreements in the same period of time

Call AF\_CheckUserServiceAssignment with the user agreement and the services. If the AF returns with valid == false, throw exception SERMGM\_005 with the conflicting user agreement and the conflicting service / services. Otherwise, proceed.

Step 1.2: Check if the user agreement is available in countries in which the services covered by the user agreement are not available

Retrieve the union of countries in which the services assigned to the user agreement are available into a set. If the list of countries in which the user agreement is available is not included in the set determined above, throw exception SERMGM\_002.

## **Step 2: Apply Changes**

Associate the committed instance of <UserAgreement> with the selected services and bring the UserAgreementServiceAssignment in the change session of the user (see general algorithm ApplyChangeSessionOnElement).

### **5.5.8.2 Input**

Name	Type / Length / BOM	Description	Changes
UserAgreement	User Agreement	The user agreement and its version.	unchanged
SelectedServices	List of Services	The assigned services	unchanged

### **5.5.8.3 Output**

None.

### **5.5.8.4 Exceptions**

- If the assignment is not valid because services of the user agreement are assigned to other user agreements in the same period of time, the error SERMGM\_005 will be returned.
- If the user agreement is available in countries in which the services covered by the user agreement are not available the error SERMGM\_002 will be returned.

## **5.5.9 AF\_GetServicesByUserAgreements**

For each given user agreement this AF retrieves the covered MBconnect services that are enabled in SOE.

### **5.5.9.1 Sequence Description**

For each given user agreement retrieve the covered MBconnect services with Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today. If there is no service found for a certain user agreement ID, an empty list is returned for this user agreement.

### **5.5.9.2 Input**

Name	Type / Length / BOM	Description
UserAgreementList	List of UserAgreement	The list of user agreements to find the services for.

Table 195: AF\_GetServicesByUserAgreements Input

### **5.5.9.3 Output**

Name	Type / Length / BOM	Description
<b>List of user agreement - service assignments (a list of services exists per UserAgreement) (1..*)</b>		
UserAgreement	UserAgreement	The user agreement to find the services for.
Services	List of Services	The services covered by the requested user agreement. Can be empty.

Table 196: AF\_GetServicesByUserAgreements Output

### **5.5.9.4 Exceptions**

None.

---

## **5.5.10 AF\_GetUserAgreementServiceAssignment**

For each given user agreement id with given user agreement version, this AF retrieves the serviceID of each covered MBconnect services that are enabled in SOE.

### **5.5.10.1 Sequence Description**

For each given user agreement retrieve the service IDs of covered MBconnect services with Service.enabledFrom is given and is smaller or equal than today and Service.enabledTo is empty or bigger than today. If there is no service found for a certain user agreement ID and user agreement version, an empty list is returned for this user agreement.

### **5.5.10.2 Input**

See internal interface “IIF\_GetServicesByUserAgreements”.

### **5.5.10.3 Output**

See internal interface “IIF\_GetServicesByUserAgreements”.

### **5.5.10.4 Exceptions**

See internal interface “IIF\_GetServicesByUserAgreements”.

## **5.5.11 AF\_CheckCountryAvailabilityForService**

This AF validates whether the list of available countries of the associated user agreement(s) is included in the list of countries where the service is made available in. In case this validity check fails, an error is thrown.

### **5.5.11.1 Sequence Description**

**Step1:** Retrieve all enabled instances of <UserAgreement> in their productive form or in the change session of the requesting user. Call IIF\_GetUserAgreementsByDate to get all available user agreements, with the input:

- Smallest Java Date value as the Valid From Date
- ChangeSession of the user

Select only the user agreements that:

- 1) references the service passed in as input parameter.
- 2) have the highest (latest) version.

**Step2:** Ensure that the service is available in all countries where the user agreements selected above are available:

Add the country sets, in which the user agreements are available, into a union set A. If the set A is not included in the list of countries where the service (provided as input parameter) is available, throw exception SERMGM\_001.

### **5.5.11.2 Input**

Name	Type / Length / BOM	Description
service	Service	The service for which the validation of country availability needs to be performed.

---

### 5.5.11.3 Output

None.

### 5.5.11.4 Exceptions

- If country assignment is not valid, the SERMGM\_001 is returned.

## 5.6 Batches

None.

## 5.7 Error Messages

Message Id	Fault Title	Fault Message
SERMGM_001	Service needs to be made available in more Countries	The availability of the Service <service> needs to cover further countries: <countryList>. Contact "MBconnect Legal Department" for more details.
SERMGM_002	Services cannot be assigned to the user agreement	The selected services <serviceList> could not be assigned to the user agreement <userAgreement> because the services are not available in all countries where the user agreement is available in.
SERMGM_003	User agreement cannot be enabled for all countries	The user agreement cannot be enabled for the following countries <countryList> because the services <serviceList> are not available in those countries. Contact the department "Product Management" for more information.
SERMGM_004	User agreement cannot be enabled	The user agreement cannot be enabled due to an already assigned service. The following service <service> is already assigned to the user agreement <user agreement> in version <version>.
SERMGM_005	Selected service already assigned	The selected service <service> is already assigned to user agreement <user agreement> in version <version>.
SERMGM_006	Cannot delete service	The service <1> could not be deleted because it is still being referenced by one or more service assignment rules, user agreements or contracts. Please go to the respective maintenance screen and remove the reference first. The service assignment rule(s) with the following ID(s) have been identified as referencing to this item: <2>. The user agreement(s) with the following ID(s) have been identified as referencing to this item: <3>.
SERMGM_007	Cannot delete service	The service <1> could not be deleted because it is still being referenced by contracts.
SERMGM_008	Cannot delete service	The service <1> cannot be deleted because it is enabled.
SERMGM_009	Invalid format for service enablement date provided.	The service enablement date needs to be provided in format dd.mm.yyyy
SERMGM_010	Invalid service enablement date provided.	The service enablement date must be greater than today.
SERMGM_011	Mandatory field missing	Please select at least one MBconnect country, the service is available in.
SERMGM_012	Invalid service enablement date provided.	The service enablement date must be greater than today and greater than the "enabled from" date.
SERMGM_013	Invalid version number	The version number has to be a positive integer. Also this positive integer has to be unique for the selected ServiceMaster <ServiceMaster>.
SERMGM_014	Invalid version and service master combination	There still exists a Version <service.versionNumber> for the selected ServiceMaster <ServiceMaster>. The version and servicemaster combination has to be unique
SERMGM_015	User agreement cannot be created in a higher version	The user agreement cannot be created in a higher version because the actual version of this user agreement is not valid for existing customers.
SERMGM_016	Date not valid	The date <date> must be greater than today.
SERMGM_017	Date not valid	The date <date> must be smaller than or equal to Info Email.
SERMGM_018	Date not valid	The date <date> must be greater than or equal to Valid from (new cus-

---

		tomer).
SERMGM_019	Date not valid	The date <date> must be smaller than Valid from (existing customer).
SERMGM_020	Date not valid	The date <date> must be greater than or equal to Valid from (new customer).
SERMGM_021	Date not valid	The date <date> must be smaller than or equal to Valid from (existing customer).
SERMGM_022	Date not valid	The date <date> must be greater than Info Email.

Table 197 Error Messages of component “Service Management”

# 6 Component „Documents“

## 6.1 Dialogs

The deletion of created master data is not allowed in any of the document dialogs. This is necessary to preserve the state of documents. Since the documents are also versioned, there is no need for a delete option.

For already maintained user agreements, there is no automatic update on the coverage of services for customers that have already signed that user agreement. This means if the user agreement is changed to cover one more service, there is no update for customers that have already signed the customer agreement previous to the change and will not be able to use that additional service. This change will only affect future consents given.

From the dialogs: DLG\_DocumentOverview, DLG\_PDFOverview, DLG\_CustomTagOverview and DLG\_DocumentBlockOverview the user has an overview over the document elements.

The initialization function of these dialogs will only load document elements that have a productive form and documents that are currently being added/modified by the logged on user. Therefore document elements that are being created by other users in their change sessions will not be visible for the logged on user.

From these “overview”-dialogs, the user can trigger to either edit or visualize the selected document element. The information whether the logged on user is able to *(i) edit or (ii) only visualize* a document element is indicated by the columns “Opened sessions” and “Action”.

Inside the column “Opened sessions” the logged on user can see whether a document is:

- edited inside the change session of another user: icon 
- edited by himself: icon 
- not edited currently: no icon

The column “Action” indicates the mode in which the logged on user can open a document. The information inside this column is derived based on the information provided by the column “Opened sessions”:

- if the document is edited inside the change session of another user then the available action “View”
- if the document is edited by himself then the available action is “Edit”
- if the document is not being edited then the available action is “Edit”

From the dialogs: DLG\_DocumentDefinition, DLG\_PDFDetail, DLG\_CustomTagDetail and DLG\_DocumentBlockDetail the user opens a document element either in “edit”-, “read-only”- or “new”-mode.

Whether a dialogs is opened in “read-only” or in “edit” mode is computed by the system. The possible action for each dialog is provided in the corresponding “overview”-dialog.

In Read-Only mode: the dialog is opened in this mode if the user has not opened his change session or if the document element is already being edited by other user. In this mode the user cannot edit or trigger changes to the document element.

In New mode: the dialog is opened in this mode if the user wants to create a new version of a document element. This mode is only possible if the user has opened a change session.

*Note: a new version of the document element will be marked with the decorator “[new]”.*

In Edit mode: the dialog is opened in this mode if the user has opened his change session and the document element is not being edited by other user. In this mode the user can modify the content of an existent version of a document element or create a new version.

### 6.1.1 DLG\_PDFOverview

This dialog provides an overview of the localized Document Templates. The Document Template itself is localized and versioned.

Add new Document Template				
Items per page 10 ▾ Page 1 2 3				
Document Template	Country	Language	Opened Sessions	Action
UA_BS	*	German (Germany)		<button>Edit</button>
UA_BS	*	Italian (Italy)		<button>View</button>
UA_BS	*	Spanish (Spain)		<button>Edit</button>

Items per page 10 ▾ Page 1|2|3

Add new Document Template

Figure 56: DLG\_PDFOverview

#### 6.1.1.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load maintained Document Templates together with the permitted actions for each loaded document template by calling the loading algorithm described in

Linked label / button labeling	Type	Action description
		Chapter "Loading of master data elements" (see details in subchapter "Dialog Elements States") Sort the list by DocumentTemplate.title, DocumentTemplate.country, DocumentTemplate.locale.
"Add new Document Template"	Button	Navigates to the dialog DLG_PDFDetail (see chapter 6.1.2) to add new Document Template.
"Edit" (1..n)	Button	Navigates to the dialog DLG_PDFDetail (see chapter 6.1.2) to edit the selected Document Template. The dialog is opened in edit mode.
"View" (1..n)	Button	Navigates to the dialog DLG_PDFDetail (see chapter 6.1.2) to visualize the selected Document Template. The dialog is opened in read-only mode.

### 6.1.1.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Table column "Document Template"	Label	Name of the Document Template.	DocumentTemplate.title
Table column "Country"	Label	Specifies which country the Document Template is localized for.	DocumentTemplate.country
Table column "Language"	Label	Specifies the language the Document Template is localized in.	DocumentTemplate.locale
Table Column "Opened Sessions"	Label/Icon	Indicates whether the entity DocumentTemplate is edited inside a change session.	-

### 6.1.1.3 Dialog field validation

None.

### 6.1.1.4 Configurability

None.

### 6.1.1.5 Dialog Elements States

See **states of the common elements displayed** by the "overview"-dialogs .

## 6.1.2 DLG\_PDFDetail

This dialog provides a detailed view of the Document Template and its versions. The Document Template itself is localized and versioned. A particular Document Template is referenced via the Document Template name and version within a Document Definition. The PDF of a certain version can be re-uploaded in case of errors that need to be fixed. When a new Document Template is created, it always starts with version 1. Further versions can be added by clicking the "New Version" button. The reason why PDF templates are versioned is because they contain localized and specific content that may change for a newer document version, while there is a necessity to retain the previous information and create the previous version of a document at the same time.

A document template can be uploaded with the same name for different country and language settings. For the document generation, the template with the most specific settings will be retrieved.

Document Template: UA\_BS

Versions:

1
2
3

Country: All

Language: German (Germany)

PDF Detail

File Uploaded: None

File: C:\UA\_Basic\_Services.pdf

Browse...      Upload

New Version      Save

Figure 57: DLG\_PDFDetail

#### 6.1.2.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		<p><u>New mode:</u> Set the initial version in the version table to “1”.</p> <p><u>Edit mode, Read-Only mode:</u> Load all maintained Document Template versions of the same DocumentTemplate.title, DocumentTemplate.country and DocumentTemplate.locale by calling AF_RetrieveVersionedDocumentElementsForUser</p>
“Browse...”	Button	Opens the native file chooser dialog to select the file for upload.
“Upload”	Button	Uploads the selected file to the server.
“New Version”	Button	<p>Increments the version number and leaves the PDF Detail empty.</p> <p>The following attributes are extracted from the previous latest version and are used to initialize the new version:</p> <ul style="list-style-type: none"> <li>- DocumentTemplate.country</li> <li>- DocumentTemplate.locale</li> <li>- DocumentTemplate.title</li> </ul> <p>Bring the currently created DocumentTemplate and the new version of the DocumentTemplate in the change session of the user. ( see general algorithm <b>ApplyChangeSessionOnElement</b>)</p>
“Save”	Button	<p>Saves the maintained PDF. Calls AF_SaveDocumentTemplate (see chapter 6.5.28).</p> <p>Bring the currently maintained DocumentTemplate (for which an available version is being edited) in the change session of the user. (see general algorithm <b>ApplyChangeSessionOnElement</b>)</p>

### 6.1.2.2 Form fields and front-end data

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Table column “Versions”	Label	The version of the PDF.	<u>Selected value:</u> DocumentTemplate.versionID  <u>Available values:</u> All DocumentTemplate.versionID of the same DocumentTemplate.title, DocumentTemplate.country and DocumentTemplate.locale  <u>Edit Mode:</u> All instances of VersionedDocumentTemplate that belong to the same Document.  <u>Read-Only mode:</u> Load only the available instances of the VersionedDocumentTemplate that are not being edited inside the change session of other users
Document Template	Textbox	Name of the Document Template.	DocumentTemplate.title
Version	Label	Version of the Document Template.	DocumentTemplate.versionID
Country	Drop-Down	Specifies which country the Document Template is localized for. Default: “All”	<u>Selected value:</u> DocumentTemplate.country  <u>Available values:</u> All countries that are supported by MBconnect. For a list of countries, <b>IIF_GetMBconnectCountries</b> is called (→ see chapter 13.4.4).  <u>Note:</u> Also a wildcard (“ALL”) is provided in order to cover all MBconnect countries (the wildcard is not part of the response by <b>IIF_GetMBconnectCountries</b> ).
Language	Drop-Down	Specifies the language the Document Template is localized in. Default: “German (Germany)”	<u>Selected value:</u> DocumentTemplate.locale  <u>Available values:</u> All available locales in the application configuration (see PROP_LOCALES, chapter 1.3.6)
File Uploaded	Label	Shows the uploaded file as a link if available and which can be downloaded by the user if clicked on.	DocumentTemplate.pdf
ReadOnlyInfo	Label	Presents this information: “The dialog is opened in read-only mode.”	-

### 6.1.2.3 Dialog field validation

Linked Field	Validation	Error Message
Document Template	The Document Template name must not be empty.	DOCMAS_008
File Uploaded	A file must have been uploaded.	DOCMAS_009

#### 6.1.2.4 Configurability

None.

#### 6.1.2.5 Dialog Elements States

Linked Label	Type	State Description
ReadOnlyInfo	Label	<u>Visible</u> : if dialog is in read-only mode <u>Invisible</u> : if dialog is in edit mode
<all buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Enabled</u> : if dialog is in edit mode <u>Disabled</u> : if dialog is in read-only mode
New Version	Button	<u>Enabled</u> : if the selected <VersionedDocumentTemplate> has the maximum version. <u>Disabled</u> : otherwise
Country	Combobox	Disabled

#### 6.1.3 DLG\_DocumentBlockOverview

This dialog provides an overview over the Document Blocks that are referenced via the block name and version within the Document Definition and represent the actual localized content.

The screenshot shows a dialog titled "Add new Document Block". At the top right are buttons for "Items per page" (set to 10) and "Page" (set to 1 of 3). Below is a table with the following data:

Document Block	Country	Language	DocTyp	Opened Sessions	Action
Content Block 1	*	German (Germany)	PDF		<button>Edit</button>
Content Block 1	*	Italian (Italy)	PDF		<button>View</button>
Content Block 1	*	Spanish (Spain)	PDF		<button>Edit</button>

At the bottom are buttons for "Items per page" (10) and "Page" (1 of 3), and a "Add new Document Block" button.

Figure 58: DLG\_DocumentBlockOverview

#### 6.1.3.1 Buttons and functions

Linked label / button labeling	Type	Action description

Linked label / button labeling	Type	Action description
<init>		Load all maintained document blocks together with the permitted actions for each loaded document block by calling the loading algorithm described in Chapter <b>Loading of master data elements</b> (see details in subchapter “Dialog Elements States”)  Sort the list by DocumentBlock.blockName, DocumentTemplate.country, DocumentTemplate.locale, and DocumentBlock.docType.
“Add New Document Block”	Button	Navigates to the dialog DLG_DocumentBlockDetail (see chapter 6.1.4) to add a new document block.
“Edit” (1..n)	Button	Navigates to the dialog DLG_DocumentBlockDetail (see chapter 6.1.4) to edit the selected document block. The dialog is opened in edit mode.
“View” (1..n)	Button	Navigates to the dialog DLG_DocumentBlockDetail (see chapter 6.1.4) to visualize the selected document block. The dialog is opened in read-only mode.

### 6.1.3.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Table column “Document Block”	Label	The name of the document block.	DocumentBlock.blockName
Table column “DocType”	Label	The document type.	DocumentBlock.docType
Table column “Country”	Label	Specifies which country the document block is localized for.	DocumentBlock.country
Table column “Language”	Label	Specifies the language the document block is localized in.	DocumentBlock.locale
Table Column “Opened Sessions”	Label/Icon	Indicates whether the entity DocumentBlock is edited inside a change session.	-

### 6.1.3.3 Dialog field validation

None.

### 6.1.3.4 Configurability

None.

### 6.1.3.5 Dialog Elements States

See **states of the common elements displayed** by the “overview”-dialogs

## 6.1.4 DLG\_DocumentBlockDetail

This dialog allows the maintenance of DocumentBlocks that are referenced via the block name and version within the document and represent the actual localized content. It is not possible to delete a maintained DocumentBlock. The reason why Document Blocks are versioned is because they contain localized and specific content that may change for a newer document version, while there is a necessity to retain the previous information and create the previous version of a document at the same time.

A categorization of DocumentBlock is introduced in order to enable only the loading of relevant document blocks in dialogs where document blocks are being referenced.

Block Name: Content Block 1

Versions: 1  
2  
3

DocType: PDF

Country: All

Language: German (Germany)

Block Type: GENERAL

DocBlock Detail

XML

<...>

New Version

Save

Figure 59: DLG\_DocumentBlockDetail

#### 6.1.4.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		<p><u>New mode:</u> Set the initial version in the version table to "1".</p> <p><u>Edit mode, Read-Only mode:</u> Load all maintained document block versions of the same DocumentBlock.blockName, DocumentBlock.country and DocumentBlock.locale by calling AF_RetrieveVersionedDocumentElementsForUser</p>
"New Version"	Button	<p>Increments the version number and leaves the document block detail empty.</p> <p>The following attributes are extracted from the previous latest version and are used to initialize the new version:</p> <ul style="list-style-type: none"> <li>- DocumentBlock.docType</li> <li>- DocumentBlock.blockName</li> <li>- DocumentBlock.locale</li> <li>- DocumentBlock.country</li> <li>- DocumentBlock.blockType</li> <li>- VersionedDocumentBlock.XMLDefinition</li> </ul> <p>Bring the parent element DocumentBlock and the new version of the DocumentBlock in the change session of the user (see general algorithm <b>ApplyChangeSessionOnElement</b>)</p>
"Save"	Button	Saves the maintained document block. Calls AF_SaveDocumentBlock (see chapter 6.5.29). The XML is validated against the DocType specific XSD to insure the structural validity of the XML.

---

<b>Linked label / button labeling</b>	<b>Type</b>	<b>Action description</b>
		Bring the currently maintained document block (for which an available version is being created) in the change session of the user. Call ApplyChangeSessionOnElement.

### 6.1.4.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Table column “Versions”	Label	Specifies the version of the document block.	<u>Selected value:</u> DocumentBlock.versionID  <u>Available values:</u> All DocumentBlock.versionID of the same DocumentBlock.title, DocumentBlock.country and DocumentBlock.locale  <u>Edit Mode:</u> All instances of VersionedDocumentTemplate that belong to the same Document.  <u>Read-Only mode:</u> Load only the available instances of the VersionedDocumentTemplate that are not being edited inside the change session of other users
Block Name	Textbox	The name of the document block.	DocumentBlock.blockName
DocType	Drop-Down	The document type.  Default: “plaintext”	<u>Selected value:</u> DocumentBlock.docType  <u>Available values:</u> All available enumerations of DocumentTypeEnum: <ul style="list-style-type: none"> <li>1. <i>PDF</i>: The document block is used for generation of PDF files</li> <li>2. <i>TXT</i>: The document type is used for generation of TXT files</li> <li>3. <i>PDF and TXT</i>: The document block is used for generation of PDF and TXT files</li> </ul>
Country	Drop-Down	Specifies which country the document block is localized for.  Default: “ALL”	<u>Selected value:</u> DocumentBlock.country  <u>Available values:</u> All countries that are supported by MBconnect. For a list of countries, <b>IIF_GetMBconnectCountries</b> is called (→ see chapter <b>13.4.4</b> ).  Note: Also a wildcard (“ALL”) is provided in order to cover all MBconnect countries (the wildcard is not part of the response by <b>IIF_GetMBconnectCountries</b> ).
Language	Drop-Down	Specifies the language the document block is localized in.  Default:	<u>Selected value:</u> DocumentBlock.locale  <u>Available values:</u>

		“German (Germany)”	All locales in the application configuration (see PROP_LOCALES, chapter 2.3.15)
XML	Textbox	The XML that specifies the actual localized content of the document block.	DocumentBlock.xmlDefinition
Block Type	Combobox	<p>The block type. Default: “GENERAL”</p>	<u>Selected value:</u> DocumentBlock.blockType  <u>Available values:</u> All available enumerations of DocumentBlockTypeEnum

#### 6.1.4.3 Dialog field validation

Linked Field	Validation	Error Message
Block name	The block name must not be empty.	DOCMAS_008

#### 6.1.4.4 Configurability

None.

#### 6.1.4.5 Dialog Elements States

	Type	State Description
New Version	Button	<u>Enabled</u> : if the selected <VersionedDocumentBlock> has the maximum version. <u>Disabled</u> : otherwise
Country	Combobox	Disabled
all other <buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Visible</u> : if dialog is in read-only mode <u>Invisible</u> : if dialog is in edit mode
all other <buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Enabled</u> : if dialog is in edit mode <u>Disabled</u> : if dialog is in read-only mode

### 6.1.5 DLG\_DocumentTriggerAssignment

This dialog allows the association of document triggers with certain documents for a specific output type. If an event is fired that contains a certain trigger, e.g. a trigger for which the customer needs to be informed, the associated documents will be sent out. Not every document trigger has all output types available. If an output type is not available, the related widgets are disabled and grayed out in the dialog. If an output type is available, it means that it is also expected, and therefore a document must be set. Refer to Document Trigger (see chapter 2.3.17.6.1) for a more detailed description of which output types are used for a certain trigger.

---

TriggerType

Output Email  
Email   
 Add relevant User Agreements

Output Mail  
Cover Letter   
 Add relevant User Agreements

Output Online PDF

Figure 60: DLG\_DocumentTriggerAssignment

#### 6.1.5.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all maintained document trigger.
“Save”	Button	Saves the association between the selected documents and the document trigger.

### 6.1.5.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
TriggerType	Drop-Down	The document trigger type.	<u>Available values:</u> All enumerations of DocumentTriggerEnum with the exception of "USER AGREEMENT LOOKUP" and "USER AGREEMENT FOR SIGNING" (because there is nothing that could be configured for them)
Email	Drop-Down	The email that is ought to be sent out.	<u>Selected value:</u> The Document, that is assigned as EMailOutputAssignment to the DocumentTriggerEnum <u>Text:</u> Document.documentID <u>Available values:</u> All instances of Document.documentID
Cover Letter	Drop-Down	The cover letter that is ought to be sent out.	<u>Selected value:</u> The Document, that is assigned as LetterOutputAssignment to the DocumentTriggerEnum <u>Text:</u> Document.documentID <u>Available values:</u> All instances of Document.documentID
Output Online PDF	Drop-Down	The main document for online output.	<u>Selected value:</u> The Document, that is assigned as EDocumentOutputAssignment to the DocumentTriggerEnum <u>Text:</u> Document.documentID <u>Available values:</u> All instances of Document.documentID
Add relevant User Agreements	Checkbox	Specifies whether relevant user agreements should be attached to the email/letter.	<u>Checked:</u> LetterOutputAssignment.addRelevantUserAgreements and EMailOutputAssignment.addRelevantUserAgreements

### 6.1.5.3 Dialog field validation

None.

### 6.1.5.4 Configurability

None.

### 6.1.5.5 Dialog Elements States

Linked Label	Type	State Description
Email	Drop-Down	Will only be enabled for the following trigger types: "User Agreement – accepted", "User Agreement – declined" and "User Agreement – legal document changed"
Cover Letter	Drop-Down	Will only be enabled for the following trigger types: "User Agreement – accepted", "User Agreement – declined", "User Agreement – legal document changed", "Vehicle Registration" and "Vehicle Separation Information"
Output Online PDF	Drop-Down	Will only be enabled for the following trigger types: "User Agreement – lookup", "User Agreement – signature", "Vehicle Separation Authorization" and "Vehicle Registration Confirmation"
Add relevant User Agree- ments	Checkbox	Will only be enabled for the following trigger types: "User Agreement – accepted", "User Agreement – declined", "User Agreement – legal document changed", "Vehicle Registration" and "Vehicle Separation Information"

### 6.1.6 DLG\_CustomTagOverview

This dialog provides an overview over the maintained custom tags.

Custom Tag Overview				
<a href="#">Add new Custom Tag</a>	<a href="#">Items per page 10</a>	<a href="#">Page 1   2   3</a>		
Custom Tag Type	Country	Doc Type	Opened Sessions	Action
CUSTOMER_ADDRESS_LONG	Germany	PDF		<a href="#">Edit</a>
CUSTOMER_ADDRESS_LONG	Italy	PDF		<a href="#">View</a>
CUSTOMER_ADDRESS_LONG	Spain	PDF		<a href="#">Edit</a>

Figure 61: DLG\_CustomTagOverview

### 6.1.6.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		Load all maintained custom tags together with the permitted actions for each loaded custom tag by calling the loading algorithm described in Chapter "Loading of master data elements".  Sort the "CustomTagType List" by CustomTag.tagType, CustomTag.country, CustomTag.documentTypeEnum.
"Add New Custom Tag"	Button	Navigates to the dialog DLG_CustomTagDetail to add a new custom tag.
"Edit" (1..n)	Button	Navigates to the dialog DLG_CustomTagDetail to edit the selected custom tag. The dialog is opened in edit mode.
"View" (1..n)	Button	Navigates to the dialog DLG_CustomTagDetail to visualize the selected custom tag. The dialog is opened in read-only mode.

### 6.1.6.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Table column "CustomTagType"	Label	The type of custom tag.	CustomTag.tagType
Table column "Country"	Label	Specifies which country this tag is localized for.	CustomTag.country
Table column "DocType"	Label	The document type of the custom tag.	CustomTag.documentElementEnum
Table Column "Opened Sessions"	Label/Icon	Indicates whether the entity CustomTag is edited inside a change session.	-

### 6.1.6.3 Dialog field validation

None.

### 6.1.6.4 Configurability

None.

### 6.1.6.5 Dialog Elements States

See states of the common elements displayed by the "overview"-dialogs

## 6.1.7 DLG\_CustomTagDetail

This dialog allows the configuration of a country specific CustomTag, like the customer address, which can then be referenced within the documents by the CustomTagType. The custom tag is not versioned because they serve a more structural purpose (than document blocks for instance) and when this structure changes, it should be reflected in the documents/emails that use the particular custom tags.

Note: The CustomTagType for the customer address, with the DocType PDF, is also used to format the address of a customer for external systems (see AF\_GetFormattedAddress, chapter 6.5.25).

CustomTagType

DocType

Country

\* The XML definition for the Custom Tag Type CUSTOMER\_ADDRESS also affects other systems that request the formatted address of a customer

XML

```
<...>
```

Figure 62: DLG\_CustomTagDetail

#### 6.1.7.1 Buttons and functions

Linked label / button labeling	Type	Action description
<init>		<u>New mode:</u> Clear all fields and select the first CustomTagType in the combo box. <u>Edit, Read only mode:</u> Load all custom tag information of the same
"Save"	Button	Saves the maintained CustomTag. The XML is validated against the DocType specific XSD to insure the structural validity of the XML. Bring the currently maintained CustomTag in the change session of the user. ( see general algorithm ApplyChangeSessionOnElement)
"Cancel"	Button	Discard the information for the current custom tag and return to DLG_CustomTagOverview.

### 6.1.7.2 Form fields and front-end data objects (AS08 reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
CustomTagType	Drop-Down	The type of custom tag.  Default: “CUSTOMER_ADDRESS_LONG”	<u>Selected value:</u> CustomTag.tagType  <u>Available values:</u> All enumerations of CustomTagType
DocType	Drop-Down	The document type of the custom tag.	<u>Selected value:</u> CustomTag.tagType  <u>Available values:</u> 1. PDF: Document type for PDF files 2. TXT: Document type for TXT files
Country	Drop-Down	Specifies which country this tag is localized for.  Default: “ALL”	<u>Selected value:</u> CustomTag.country  <u>Available values:</u> All countries that are supported by MBconnect. For a list of countries, <b>IIF_GetMBconnectCountries</b> is called (→ see chapter 13.4.4).  <u>Note:</u> Also a wildcard (“ALL”) shall be provided in order to cover all MBconnect countries (the wildcard is not part of the response by <b>IIF_GetMBconnectCountries</b> ).

### 6.1.7.3 Dialog field validation

None.

### 6.1.7.4 Configurability

None.

### 6.1.7.5 Dialog Elements States

Linked Label	Type	State Description
ReadOnlyInfo	Label	<u>Visible</u> : if dialog is in read-only mode <u>Invisible</u> : if dialog is in edit mode
<all buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Enabled</u> : if dialog is in edit mode <u>Disabled</u> : if dialog is in read-only mode

### 6.1.8 DLG\_DocumentPreview

In order to preview the impact of modified data on the document **Basic Services** the following data is required

Version

Select Country and language

DocumentUsageType

Override Template for preview

Template for preview

Figure 63: DLG\_DocumentPreview

### 6.1.8.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		<p>For dialogue initialization the following AF(s) are called:</p> <ul style="list-style-type: none"> <li>- AF_RetrieveVersionedDocumentElementsForUser: sort the returned result in ascendant order. Fill the "Version" dropdown box.</li> <li>- AF_GetSupportedMBconnectCountryLocales sort the returned result in lexicographic order. Fill the "CountryAndLanguage" dropdown box.</li> <li>- Load maintained Document Templates, extract each version and create a flat list comprising all versioned document templates. Fill the "Document Template" dropdown box.</li> </ul>

---

Override document template for Preview	Checkbox	Activates / Disables the Document Template dropdown box. If checkbox is unchecked, the current selection will be reverted and no Document Template selection is possible.
Download Document for preview	Button	Downloads the document as PDF for preview. Calls AF_GeneratePDFDocumentForPreview.  Please note:  If the checkbox "Override document template for Preview" is enabled: The parameter VersionedDocumentTemplate is filled with the value chosen in the drop down box "documentTemplate".  If the checkbox is disabled: VersionDocumentTemplate=NULL.
OK	Button	Disposes dialog

### 6.1.8.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribute (AS08 reference)
Version	Dropdown		<u>Available values:</u> See return result of AF_RetrieveVersionedDocumentElementsForUser <u>Selected value:</u> First element of the return list.
CountryAndLanguage	Dropdown	Country + "(" + Language + ")"	<u>Available values:</u> See return result of AF_GetSupportedMBconnectCountryLocales  <u>Selected value:</u> First element of the return list.
DocumentUsageEnum	Dropdown	The document type of the custom tag.	<u>Available values:</u> All available enumerations of DocumentUsageEnum <u>Selected value:</u> First element of the return list
documentTemplate	Dropdown	The document template to be used for document preview  DocumentTemplateTitle+(v"+ TemplateVersion +" "+Language	<u>Available values</u> All available Document Templates, including offered versions and languages <u>Selected value:</u> First element of the return list

### 6.1.8.3 Dialogue field validation

None.

### 6.1.8.4 Configurability (incl setting for roles)

None.

### 6.1.8.5 Dialog Elements State

Linked Label	Type	State Description
Template to use	Dropdown	<u>Enabled:</u> Elements in the dropdown box can be only selected if override document template for preview checkbox is checked.

### 6.1.9 DLG\_ImageOverview

This dialog provides an overview of images uploaded and persisted in SOE. The images are versioned.

The screenshot shows a dialog titled 'DLG\_ImageOverview'. At the top right are buttons for 'Add new' and 'Items per page' (set to 10) and 'Page' (set to 1). Below this is a table with three columns: 'Document Image', 'Opened Sessions', and 'Action'. The table contains two rows: one for 'FACSIMILE\_1' with an 'Edit' button and a gear icon, and one for 'FACSIMILE\_2' with a 'View' button. At the bottom are buttons for 'Add new' and 'Items per page' (set to 10) and 'Page' (set to 1).

Document Image	Opened Sessions	Action
FACSIMILE_1		Edit
FACSIMILE_2		View

Figure 64: DLG\_ImageOverview

### 6.1.9.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		Load maintained Document Images together with the permitted actions for each loaded document template by calling the loading algorithm described in Chapter "Loading of master data elements" (see details in subchapter "Dialog Elements States") Sort the list by DocumentImage.name.
"Add new"	Button	Navigates to the dialog DLG_ImageDetail see (chapter 6.1.10) to add a new Document Image.
"Edit" (1..n)	Button	Navigates to the dialog DLG_ImageDetail see (chapter 6.1.10) to edit the selected Document Image. The dialog is opened in edit mode.
"View" (1..n)	Button	Navigates to the dialog DLG_ImageDetail see (chapter 6.1.10) to visualize the selected Document Image. The dialog is opened in read-only mode.

Table 198: Buttons and functions

### 6.1.9.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Table column “Document Image”	Label	Name of the Document Image.	DocumentImage.name
Table Column “Opened Sessions”	Label/Icon	Indicates whether the entity DocumentImage is edited inside a change session.	-

Table 199: Form fields and front-end data objects

### 6.1.9.3 Dialogue field validation

None.

### 6.1.9.4 Configurability (incl setting for roles)

None.

### 6.1.9.5 Dialog Elements State

See states of the common elements displayed by the “Overview”-dialogs (in chapter “Target Business Functionality”).

## 6.1.10 DLG\_ImageDetail

This dialog provides a detailed view of the document images and its versions. The Document Image itself is versioned. A particular image is referenced via the Document Image name and version within a Document Definition. The image behind a certain version can be re-uploaded in case of corrections or updates. When a new Document Image is created, it always starts with version 1. Further versions can be added by clicking the “New Version” button. The reason why Images are versioned is because there is a necessity to retain the previous information and create the previous version of a document at the same time.

Image Name: FACSIMILE\_1

Versions: 1  
2  
3

Image Detail

File Uploaded: C:\Facsimile\_1.jpg

File\*: Browse...

300 x 150

New Version      Save      Cancel

Figure 65: DLG\_ImageDetail

---

### 6.1.10.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		<u>New mode:</u> Set the initial version in the version table to "1". <u>Edit mode, Read-Only mode:</u> Load all maintained Document Image versions of the same DocumentImage.name by calling AF_RetrieveVersionedDocumentElementsForUser
"Browse..."	Button	Opens the native file chooser dialog to select the file for upload. Only allow following extensions: jpg, tif, gif, bmp, png and wmf.
"New Version"	Button	Increments the version number and leaves the Document Image empty. The following attributes are extracted from the previous latest version and are used to initialize the attribute of the new version: DocumentImage.name
"Save"	Button	Saves the maintained image. Calls AF_SaveDocumentImage (see chapter 6.5.30). Bring the currently maintained DocumentImage (for which an available version is being edited) in the change session of the user.
"Cancel"	Button	Cancels any changes.

Table 200: Buttons and functions

### 6.1.10.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Table column “Versions”	Label	The version of the Image.	<p><u>Selected value:</u> VersionedDocumentImage.versionID</p> <p><u>Available values:</u> All VersionedDocumentImage.versionID of the image identified by DocumentImage.name</p> <p><u>If dialog is opened in “Edit” Mode:</u> Load all productive instances of VersionedDocumentImage Together with all instances of the VersionedDocumentImage that are being created/edited inside the change session of user</p> <p><u>If dialog is opened in “Read-Only” mode:</u> All productive instances of VersionedDocumentImage of the image identified by DocumentImage.name</p>
Image Name	Textbox	Name of the Document Image.	DocumentImage.name
Version	Label	Version of the Document Image.	VersionedDocumentImage.versionID
File Uploaded	Label	Shows the uploaded file as a link if available and which can be downloaded by the user if clicked on.	The path from where the VersionedDocumentImage has been uploaded
Image Preview Area	Panel	Shows the image preview of the uploaded file inside an area limited to 300 x 150 pixels.	VersionedDocumentImage.image The image is scaled to fit inside a window having the following dimensions: Width = 300 pixels Height = 150 pixels
ReadOnlyInfo	Label	Presents this information: “The dialog is opened in read-only mode.”	-

Table 201: Form fields and front-end data objects

#### 6.1.10.3 Dialogue field validation

None.

#### 6.1.10.4 Configurability (incl setting for roles)

None.

### 6.1.10.5 Dialog Elements State

Linked Label	Type	State Description
ReadOnlyInfo	Label	<u>Visible</u> : if dialog is in read-only mode <u>Invisible</u> : if dialog is in edit mode
<all buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	<u>Enabled</u> : if dialog is in edit mode <u>Disabled</u> : if dialog is in read-only mode

## 6.2 External View - Offered Interfaces

### 6.2.1 IF\_SOE\_GetDocumentForConfirmationOfVehicleRegistration

**Communication type:** Synchronously

This interface returns a document with a confirmation code. With this confirmation code the customer can confirm the pre-existing vehicle registration to his MBconnect account inside of MyMercedes.

Internally, AF\_GetDocumentForConfirmationOfVehicleRegistration is called to retrieve the documents.

#### 6.2.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
user	Mand.	UserType		CPD::User		The user of a customer.
confirmationCode	Mand.	String		Used as input parameter for AF_GetDocumentForConfirmationOfVehicleRegistration	-	The confirmation code to confirm the vehicle registration. This information is printed onto the document.
locale	Mand.	String	5	Used as input parameter for AF_GetDocumentForConfirmationOfVehicleRegistration	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies which language the document is supposed to be created in.

Table 202: External interface input

#### 6.2.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Document	Mand.	Binary PDF	-	Binary PDF	-	The requested vehicle confirmation document in PDF format.

Table 203: External interface output

### **6.2.1.3 Exceptions**

None.

## **6.2.2 IF\_SOE\_GetDocumentForVehicleSeparation**

**Communication type:** Synchronously

This interface returns a document which, when signed by the customer, authorize the retailer to separate the customer's vehicle from its linked account.

Internally, AF\_GetDocumentForVehicleSeparation is called to retrieve the documents.

### **6.2.2.1 Input**

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
User-Id	Mand.	String		User.userID		The user ID of the customer. The ID is necessary to retrieve customer information (e.g name) that is printed onto the documents.
FIN	Mand.	String	17	Vehicle.fin	Example: WDD169007 1J236589	The vehicle identification number is vehicle specific. This information is printed onto the document.
Locale	Mand.	String	5	Used as input parameter to determine document language (AF_GetDocumentForVehicleSeparation )	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies which language the document is supposed to be created in.

### **6.2.2.2 Output**

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Document	Mand.	Binary PDF	-	Binary PDF	-	The requested vehicle separation document in PDF format.

### **6.2.2.3 Exceptions**

- If the user ID does not exist, the error ACCDAS\_002 is returned
- If the FIN does not exist, the error VEHPRO\_005 is returned

## **6.2.3 IF\_SOE\_GetFormattedAddress**

**Communication type:** Synchronously

This interface returns the formatted address of the given customers based on their country.

Internally, AF\_GetFormattedAddress is called to retrieve the formatted address, with the following parameters:

- <sentFromCountry> = leave empty
- <localeForAddressCountry> = leave empty
- all other parameters are forwarded unchanged

### 6.2.3.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
AddressType Enum	Mand.	String	5	-	Example: "SHORT", "LONG"	Specifies whether the short or long version of the address is supposed to be returned. The short version does not contain any name related attributes.
Locale	Mand.	String	5	Used as input parameter to determine output language	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country).
<b>List of matchingCode + UserType</b>						
matchingCode	Opt.	String		-	Examples: "1", "user_03"	The matchingCode is necessary to identify a User Profile with no UserID.
See <b>Handling of Profile Data within SOE</b> (→ see chapter 2.3.18) for an overview of the attributes.						

Table 204: IF\_SOE\_GetFormattedAddress Input

### 6.2.3.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of matchingCode + formatted addresses</b>						
matchingCode	Opt.	String		-	Examples: "1", "user_03"	The matchingCode is necessary to identify a User Profile with no UserID.
<b>List of Address Lines (1..n)</b>						
Address Line	Mand.	String	-	-	Example: "Max Mustermann", "1 My Street", "70599 Stuttgart"	Formatted address line based on the customer's country.

Table 205: IF\_SOE\_GetFormattedAddress Output

### 6.2.3.3 Exceptions

- If the country specific <CustomTag> cannot be found, return the error GENDOC\_006

## 6.2.4 IF\_SOE.GetUserAgreementMasterData

This interface provides access to the user agreement related master data. If there is no list of user agreements given as input parameter, the answer covers all user agreements existing in SOE.

Internally, AF.GetUserAgreementMasterData is called.

### 6.2.4.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of user agreements (0..*) - Optional</b>						
uaID	Mand.	String	-	Document.documentID	Examples: 1, 12356, 1234567890	The id of the user agreement
uaVersion	Mand.	Int		Document.VersionID	Examples: 1, 12356, 1234567890	The version of the user agreement.

Table 206: IF\_SOE\_GetUserAgreementMasterData Input

### 6.2.4.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of user agreements (each of the following parameters exist per uaID) (1..*)</b>						
uaID	Mand.	String	-	Document.documentID	Examples: 1, 12356, 1234567890	The id of the user agreement
uaVersion	Mand.	Int	10	Document.VersionID	Examples: 1, 12356, 1234567890	The version of the user agreement.
<b>- Inner List of user agreement descriptions for different locales (can be empty)</b>						
locale	Mand.	String	5	-	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (language in combination with a country). Specifies which language the user agreement name is supposed to be created in.
uaName	Mand.	String	-	UserAgreement.Name	Example: "User Agreement – Basic Services"	The localized name of the user agreement.

Table 207: IF\_SOE\_GetUserAgreementMasterData Output

### 6.2.4.3 Exceptions

None.

## 6.2.5 IF\_SOE\_GetTermsOfUseMasterDataForCountry

**Communication type:** Synchronously

This interface provides access to the available user agreement related master data of a given country.

Internally, AF\_GetTermsOfUseMasterDataForCountry is called.

### 6.2.5.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
addressCoun	Mand	String	2	MbcCountry.country	Examples:	The address country of the

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
try				Code	"DE", "CH", "UK", ...	retailer. Based on this country, the list of available user agreements is determined.
locale	Opt.	String	5	-	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (language in combination with a country). Specifies which language the user agreement name is supposed to be created in.

Table 208: IF\_SOE\_GetTermsOfUseMasterDataForCountry Input

### 6.2.5.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of user agreements (each of the following parameters exist per user agreement ID) (0..*)</b>						
User Agreement ID	Mand.	String	-	Document.DocumentID	Examples: 1, 12356, 1234567890	The id of the user agreement
User Agreement Version	Mand.	Int	10	Document.VersionID	Examples: 1, 12356, 1234567890	The version of the user agreement.
<b>- Inner List of user agreement descriptions for different locales (can be empty)</b>						
locale	Mand.	String	5	-	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (language in combination with a country). Specifies which language the user agreement name is created in.
uaName	Mand.	String	-	UserAgreement.Name	Example: "User Agreement – Basic Services"	The localized name of the user agreement.
<b>List of available languages (0..*)</b>						
language	Mand.	String	2	-	Examples: "DE", "FR"	Language code consisting of two characters, e.g. "de". This field specifies in which language the returned documents are to be delivered.

Table 209: IF\_SOE\_GetTermsOfUseMasterDataForCountry Output

### 6.2.5.3 Exceptions

No exceptions.

## 6.3 External View - Consumed Interfaces

None.

---

## **6.4 Internal View - Offered Interfaces**

This chapter contains interfaces which are offered for other components within the SOE system (so called internal interfaces).

### **6.4.1 IIF\_GetDocumentTemplates**

Internally calls AF\_GetDocumentTemplates (→ see chapter 6.5.1) to retrieve all existing DocumentTemplate entities.

### **6.4.2 IIF\_GetDocuments**

Internally calls AF\_GetDocuments (→ see chapter 6.5.2) to retrieve all existing Document entities.

### **6.4.3 IIF\_GetDocumentBlocks**

Internally calls AF\_GetDocumentBlocks (→ see chapter 6.5.3) to retrieve all existing DocumentBlock entities.

### **6.4.4 IIF\_GetTrigger2DocAssignments**

Internally calls AF\_GetTrigger2DocAssignments (→ see chapter 6.5.4) to retrieve all existing Trigger2DocAssignments entities.

### **6.4.5 IIF\_GetCustomTags**

Internally calls AF\_GetCustomTags (→ see chapter 6.5.5) to retrieve all existing CustomTag entities.

### **6.4.6 IIF\_GetDocumentImages**

Internally calls AF\_GetDocumentImages (→ see chapter 6.5.6) to retrieve all existing DocumentImage entities.

### **6.4.7 IIF\_UpdateDocumentTemplates**

Internally calls AF\_UpdateDocumentTemplates (→ see chapter 6.5.7) to update all DocumentTemplate entities.

### **6.4.8 IIF\_UpdateDocuments**

Internally calls AF\_UpdateDocuments (→ see chapter 6.5.8) to update all Document entities.

### **6.4.9 IIF\_UpdateDocumentBlocks**

Internally calls AF\_UpdateDocumentBlocks (→ see chapter 6.5.9) to update all DocumentBlock entities.

### **6.4.10 IIF\_UpdateTrigger2DocAssignments**

Internally calls AF\_UpdateTrigger2DocAssignments (→ see chapter 6.5.10) to update all Trigger2DocAssignment entities.

---

#### **6.4.11 IIF\_UpdateCustomTags**

Internally calls AF\_UpdateCustomTags (→ see chapter 6.5.11) to update all CustomTag entities.

#### **6.4.12 IIF\_UpdateDocumentImages**

Internally calls AF\_UpdateDocumentImages (→ see chapter 6.5.12) to update all DocumentImage entities.

#### **6.4.13 IIF\_GetServicesByUserAgreements**

For each user agreement in the given list of user agreements this internal interface retrieves the MBconnect services which are covered by the user agreement. If there is no service found for a certain user agreement ID, an empty list is returned for this user agreement.

Internally calls AF\_GetServicesByUserAgreements (→ section 5.5.9) to retrieve the available services.

#### **6.4.14 IIF\_InformCustomer**

This interface determines the documents/emails and sends them to the customer to inform him about certain events that happened, e.g. when he signed a user agreement or registered a new vehicle.

Internally AF\_InformCustomer (→ section 6.5.13) is called to send documents/emails to the customer.

#### **6.4.15 IIF.GetUserAgreementsByDate**

This interface determines the enabled form of the user agreements that either have an inform date or a valid from date that lies between the given date and today. Additionally, if a previous productive version of the user agreement is available, it is also returned along with the determined user agreement.

Internally call AF.GetUserAgreementsByDate (→ section 6.5.32) to get correct user agreements.

#### **6.4.16 IIF.GetUserAgreementsByIdAndVersion**

This interface returns the user agreement that has the provided Id and matching version id.

Internally call AF.GetUserAgreementsByIdAndVersion (→ section 6.5.36) to get correct user agreement.

#### **6.4.17 IIF\_BuildLegalDocuments**

**Communication type:** Synchronously

This interface returns a list of legal documents (user agreements + terms and conditions). If the request leads to the generation of several documents, the documents can be requested as document bundle i.e. all documents are merged into one resulting document. If the input parameter “RequestDocumentBundle” is set to true, then only distinct

---

versions of the terms and conditions will be appended at the end of the generated document.

Internally AF\_BuildLegalDocuments (→ section 6.5.21) is called to retrieve the documents.

#### **6.4.18 IIF\_LogChangedServiceAvailability**

Each time, when the service availability changes for a customer and his vehicle caused by

- enablement of services or adding the service availability for the customers address country or
  - disablement of services or removing the service availability for the customers address country or
  - changing service assignment rules
- the internal interface logs this information.

*Hint: Currently this interface does NOT cover the sending of documents/emails to inform the customer about changed service availability. SOE only logs the changed service availability for the customer. In future this internal interface additionally will cover the sending of documents/emails to inform the customer.*

Internally AF\_InformCustomerOfChangedServiceAvailability (→ section 6.5.14) will be called.

### **6.5 Implementation**

#### **6.5.1 AF\_GetDocumentTemplates**

##### **6.5.1.1 General Description**

This application function provides all existing DocumentTemplate entities.

##### **6.5.1.2 Sequence Description**

Determine all instances of entity <DocumentTemplate> and information from related entities that are available in SOE and return the data.

##### **6.5.1.3 Input**

None.

##### **6.5.1.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;DocumentTemplate&gt;: List of all DocumentTemplate entities</b>		
Country	DocumentTemplate.country.countryCode	The Country of a DocumentTemplate
Locale	DocumentTemplate.locale	The Locale of a DocumentTemplate.
Title	DocumentTemplate.title	The Title of a DocumentTemplate.
- Inner List of <VersionedDocumentTemplate>		
versionId	VersionedDocumentTemplate.versionID	Version Id of the document template. This version is independent of the VersionedDocument's version id.
Pdf	VersionedDocumentTemplate.pdf	Binary attribute storing the PDF document that is used as template for the document. This is the actual template data.

---

Table 210: AF\_GetDocumentTemplates output

### 6.5.1.5 Exceptions

None.

## 6.5.2 AF\_GetDocuments

### 6.5.2.1 General Description

This application function provides all existing Document entities.

### 6.5.2.2 Sequence Description

Determine all instances of entity <Document> and information from related entities that are available in SOE and return the data.

### 6.5.2.3 Input

None.

### 6.5.2.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;Document&gt;: List of all Document entities</b>		
DocumentID	Document.documentID	The ID of a Document.
DocType	Document.docType	The Type of a Document.
isUserAgreement	Document.isUserAgreement	Indicates whether a document represents a user agreement or not. (This is necessary for special functional treatment of user agreements.)
- Inner List of <VersionedDocument>		
Versioned	VersionedDocument.versionId	The VersionID of a Versioned Document.
documentXmlDefinition	VersionedDocument.document	Root entity of the document definition. It represents the brace for all contained elements.
validFrom	VersionedDocument.validFrom	The validFrom Date of a VersionedDocument.
1. Inner List of <String(50)>		
title	VersionedDocument.title	Translation of a versioned document's title (i18n)
2. Inner List of <MbcCountry>		
country	MbcCountry.countryCode	The Country of a Versioned Document
- UserAgreement		
Enabled	UserAgreement.enabled	<u>Optional:</u> Boolean flag that indicates if the user agreement is enabled and thus be available to be processed, e.g. for signing. It also indicates whether the dates for a user agreement can still be changed or not. Once the flag is TRUE, it is not possible to change the dates any more.
informByEMailDate	UserAgreement.informByEMailDate	<u>Optional:</u> Date when the notifications of user agreement changes are sent out via email (for customers with mail address).
validFromForExisting-Customer	UserAgreement.validFromForExistingCustomer	<u>Optional:</u> Start of validity of a user agreement for existing customer. Note, "validFrom" is used for all new clients. This field here is only used for existing customers instead.

Table 211: AF\_GetDocuments output

### 6.5.2.5 Exceptions

None.

---

### **6.5.3 AF\_GetDocumentBlocks**

#### **6.5.3.1 General Description**

This application function provides all existing Document Block entities.

#### **6.5.3.2 Sequence Description**

Determine all instances of entity <DocumentBlock> and information from related entities that are available in SOE and return the data.

#### **6.5.3.3 Input**

None.

#### **6.5.3.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;DocumentBlock&gt;: List of all DocumentBlock entities</b>		
BlockName	DocumentBlock.BlockName	The Name of a Document Block.
Country	Document-Block.country.countryCode	The Country of a Document Block.
DocType	DocumentBlock.docType	The DocType of a Document Block.
Locale	DocumentBlock.locale	The Locale of a Document Block.
BlockType	DocumentBlock.blockType	The BlockType of a Document Block.
- Inner List of <VersionedDocumentBlock>		
versionId	VersionedDocumentBlock.versionID	The VersionID of a Versioned Document Block.
xmlDefinition	VersionedDocumentBlock.xmlDefinition	The XMLDefinition of a Versioned Document Block.

Table 212: AF\_GetDocumentBlocks output

#### **6.5.3.5 Exceptions**

None.

### **6.5.4 AF\_GetTrigger2DocAssignments**

#### **6.5.4.1 General Description**

This application function provides all existing Trigger2DocAssignment entities.

#### **6.5.4.2 Sequence Description**

Determine all instances of entity <Trigger2DocAssignment> and information from related entities that are available in SOE and return the data.

#### **6.5.4.3 Input**

None.

#### **6.5.4.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;Trigger2DocAssignment&gt;: List of all Trigger2DocAssignment entities</b>		
triggerType	Trigger2DocAssignment.triggerType	The triggerType of a Trigger2DocAssignment.
- List of <OutputTypeAssignment>		
outputType	OutputTypeAssignment	The outputType of a OutputTypeAssignment.

Parameter Name	Type / Length / BOM	Description
	ment.outputType	
documentID	Document.documentID	The DocumentID of a Document
addRelevantUser-Agreements	LetterOutputAssignment.addRelevantUserAgreements / EMailOutputAssignment.addRelevantUserAgreements	<u>Optional:</u> Flag indicating if relevant user agreements shall be added.

Table 213: AF\_GetTrigger2DocAssignments output

#### 6.5.4.5 Exceptions

None.

### 6.5.5 AF\_GetCustomTags

#### 6.5.5.1 General Description

This application function provides all existing Custom Tag entities.

#### 6.5.5.2 Sequence Description

Determine all instances of entity <CustomTag> that are available in SOE and return the data.

#### 6.5.5.3 Input

None.

#### 6.5.5.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;CustomTag&gt;: List of all CustomTag entities</b>		
Country	Custom-Tag.country.countryCode	The country of a Custom Tag.
DocumentType	CustomTag.documentElementType	The DocumentType of a Custom Tag.
TagType	CustomTag.tagType	The TagType of a Custom Tag.
xmlDefinition	CustomTag.xmlDefinition	The XMLDefinition of a Custom Tag.

Table 214: AF\_GetCustomTags output

#### 6.5.5.5 Exceptions

None.

### 6.5.6 AF\_GetDocumentImages

#### 6.5.6.1 General Description

This application function provides all existing DocumentImage entities.

#### 6.5.6.2 Sequence Description

Determine all instances of entity <DocumentImage> and information from related entities that are available in SOE and return the data.

#### 6.5.6.3 Input

None.

#### 6.5.6.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;DocumentImage&gt;: List of all Image entities</b>		
Name	DocumentImage.name	The Name of a DocumentImage.
- Inner List of <VersionedDocumentImage>		
versionId	VersionedDocumentImage.versionID	The binary representation of the image that is represented by the VersionedDocumentImage.
Image	VersionedDocumentImage.image	Version Id of the image. This version is independent of the VersionedDocument's version id.

Table 215: AF\_GetDocumentImage output

#### 6.5.6.5 Exceptions

None.

### 6.5.7 AF\_UpdateDocumentTemplates

#### 6.5.7.1 General Description

This application function updates the DocumentTemplate entities.

#### 6.5.7.2 Sequence Description

Compare the <DocumentTemplate> given as input parameter with the existing <DocumentTemplate> in SOE retrieved by calling AF\_GetDocumentTemplates.

##### If the updateMode = ADD

Add all instances of <DocumentTemplate> which are given as input parameter and not parts of the existing <DocumentTemplate>.

##### If the updateMode = UPDATE

Update all instances of <DocumentTemplate> that are available in SOE which are not completely equal to the <DocumentTemplate> given as input parameter.

##### If the updateMode = DELETE

Delete all instances of <DocumentTemplate> that are available in SOE which are not parts of the <DocumentTemplate> given as input parameter.

#### 6.5.7.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the document template entities.
<b>List&lt;DocumentTemplate&gt;: List of either all DocumentTemplate entities or specific DocumentTemplate entities – if given</b>		
Country	DocumentTemplate.country.countryCode	The Country of a DocumentTemplate
Locale	DocumentTemplate.locale	The Locale of a DocumentTemplate.
Title	DocumentTemplate.title	The Title of a DocumentTemplate.
- Inner List of <VersionedDocumentTemplate>		
VersionId	VersionedDocumentTemplate.versionID	Version Id of the document template. This version is independent of the VersionedDocument's version id.
Pdf	VersionedDocumentTemplate.pdf	Binary attribute storing the PDF document that is used as template for the document. This is the actual template data.

Table 216: AF\_UpdateDocumentTemplates input

---

## 6.5.7.4 Output

None.

## 6.5.7.5 Exceptions

None.

## 6.5.8 AF\_UpdateDocuments

### 6.5.8.1 General Description

This application function updates the Document entities.

### 6.5.8.2 Sequence Description

Compare the <Document> given as input parameter with the existing <Document> in SOE retrieved by calling AF\_GetDocuments.

#### If the updateMode = ADD

Add all instances of <Document> which are given as input parameter and not parts of the existing <Document>.

#### If the updateMode = UPDATE

Update all instances of <Document> that are available in SOE which are not completely equal to the <Document> given as input parameter.

#### If the updateMode = DELETE

Delete all instances of <Document> that are available in SOE which are not parts of the <Document> given as input parameter.

## 6.5.8.3 Input

Parameter Name	Type / Length / BOM	Description
<b>List&lt;Document&gt;: List of all Document entities</b>		
DocumentID	Document.documentID	The ID of a Document.
DocType	Document.docType	The Type of a Document.
isUserAgreement	Document.isUserAgreement	Indicates whether a document represents a user agreement or not. (This is necessary for special functional treatment of user agreements.)
- Inner List of <VersionedDocument>		
Versioned	VersionedDocument.versionId	The VersionID of a Versioned Document.
documentXmlDefinition	VersionedDocument.document	Root entity of the document definition. It represents the brace for all contained elements.
validFrom	VersionedDocument.validFrom	The validFrom Date of a VersionedDocument.
1. Inner List of <String(50)>		
title	VersionedDocument.title	Translation of a versioned document's title (i18n)
2. Inner List of <MbcCountry>		
country	MbcCountry.countryCode	The Country of a Versioned Document
- UserAgreement		
Enabled	UserAgreement.enabled	<u>Optional:</u> Boolean flag that indicates if the user agreement is enabled and thus be available to be processed, e.g. for signing. It also indicates whether the dates for a user agreement can still be changed or not. Once the flag is TRUE, it is not possible to change the dates any more.
informByEMailDate	UserAgreement.informByEMailDate	<u>Optional:</u> Date when the notifications of user agreement changes are sent out via email (for customers with mail address).

Parameter Name	Type / Length / BOM	Description
validFromForExisting-Customer	UserAgreement.validFromForExistingCustomer	Optional: Start of validity of a user agreement for existing customer. Note, "validFrom" is used for all new clients. This field here is only used for existing customers instead.

Table 217: AF\_UpdateDocuments input

#### 6.5.8.4 Output

None.

#### 6.5.8.5 Exceptions

None.

### 6.5.9 AF\_UpdateDocumentBlocks

#### 6.5.9.1 General Description

This application function updates the DocumentBlock entities.

#### 6.5.9.2 Sequence Description

Compare the <DocumentBlock> given as input parameter with the existing <DocumentBlock> in SOE retrieved by calling AF\_GetDocumentBlocks.

##### If the updateMode = ADD

Add all instances of <DocumentBlock> which are given as input parameter and not parts of the existing <DocumentBlock>.

##### If the updateMode = UPDATE

Update all instances of <DocumentBlock> that are available in SOE which are not completely equal to the <DocumentBlock> given as input parameter.

##### If the updateMode = DELETE

Delete all instances of <DocumentBlock> that are available in SOE which are not parts of the <DocumentBlock> given as input parameter.

#### 6.5.9.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the document block entities.
<b>List&lt;DocumentBlock&gt;: List of either all DocumentBlock entities or specific DocumentBlock entities – if given</b>		
BlockName	DocumentBlock.BlockName	The Name of a Document Block.
Country	Document-Block.countryCode	The Country of a Document Block.
DocType	DocumentBlock.docType	The DocType of a Document Block.
Locale	DocumentBlock.locale	The Locale of a Document Block.
BlockType	DocumentBlock.blockType	The BlockType of a Document Block.
- Inner List of <VersionedDocumentBlock>		
versionId	VersionedDocumentBlock.versionID	The VersionID of a Versioned Document Block.
xmlDefinition	VersionedDocumentBlock.xmlIDefinition	The XMLDefinition of a Versioned Document Block.

Table 218: AF\_UpdateDocumentBlocks input

---

#### 6.5.9.4 Output

None.

#### 6.5.9.5 Exceptions

None.

### 6.5.10 AF\_UpdateTrigger2DocAssignments

#### 6.5.10.1 General Description

This application function updates the Trigger2DocAssignment entities.

#### 6.5.10.2 Sequence Description

Compare the <Trigger2DocAssignment> given as input parameter with the existing <Trigger2DocAssignment> in SOE retrieved by calling AF\_GetTrigger2DocAssignments.

##### If the updateMode = ADD

Add all instances of <Trigger2DocAssignment> which are given as input parameter and not parts of the existing <Trigger2DocAssignment>.

##### If the updateMode = UPDATE

Update all instances of <Trigger2DocAssignment> that are available in SOE which are not completely equal to the <Trigger2DocAssignment> given as input parameter.

##### If the updateMode = DELETE

Delete all instances of <Trigger2DocAssignment> that are available in SOE which are not parts of the <Trigger2DocAssignment> given as input parameter.

#### 6.5.10.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the model trigger 2 doc assignment entities.
<b>List&lt;Trigger2DocAssignment&gt;: List of either all Trigger2DocAssignment entities or specific Trigger2DocAssignment entities – if given</b>		
triggerType	Trigger2DocAssignment.triggerType	The triggerType of a Trigger2DocAssignment.
- List of <OutputTypeAssignment>		
outputType	OutputTypeAssignment.outputType	The outputType of a OutputTypeAssignment.
documentID	Document.documentID	The DocumentID of a Document
addRelevantUserAgreements	LetterOutputAssignment.addRelevantUserAgreements / EMailOutputAssignment.addRelevantUserAgreements	<u>Optional:</u> Flag indicating if relevant user agreements shall be added.

Table 219: AF\_UpdateTrigger2DocAssignments input

#### 6.5.10.4 Output

None.

---

### **6.5.10.5 Exceptions**

None.

## **6.5.11 AF\_UpdateCustomTags**

### **6.5.11.1 General Description**

This application function updates the CustomTag entities.

### **6.5.11.2 Sequence Description**

Compare the <CustomTag> given as input parameter with the existing <CustomTag> in SOE retrieved by calling AF\_GetCustomTags.

#### **If the updateMode = ADD**

Add all instances of <CustomTag> which are given as input parameter and not parts of the existing <CustomTag>.

#### **If the updateMode = UPDATE**

Update all instances of <CustomTag> that are available in SOE which are not completely equal to the <CustomTag> given as input parameter.

#### **If the updateMode = DELETE**

Delete all instances of <CustomTag> that are available in SOE which are not parts of the <CustomTag> given as input parameter.

### **6.5.11.3 Input**

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the custom tag entities.
<b>List&lt;CustomTag&gt;: List of either all CustomTag entities or specific CustomTag entities – if given</b>		
Country	Custom-Tag.country.countryCode	The country of a Custom Tag.
DocumentType	CustomTag.documentType	The DocumentType of a Custom Tag.
TagType	CustomTag.tagType	The TagType of a Custom Tag.
xmlDefinition	CustomTag.xmlDefinition	The XMLDefinition of a Custom Tag.

Table 220: AF\_UpdateCustomTags input

### **6.5.11.4 Output**

None.

### **6.5.11.5 Exceptions**

None.

## **6.5.12 AF\_UpdateDocumentImages**

### **6.5.12.1 General Description**

This application function updates the DocumentImage entities.

### **6.5.12.2 Sequence Description**

Compare the <DocumentImage> given as input parameter with the existing <DocumentImage> in SOE retrieved by calling AF\_GetDocumentImages.

#### **If the updateMode = ADD**

Add all instances of <DocumentImage> which are given as input parameter and not parts of the existing <DocumentImage>.

#### **If the updateMode = UPDATE**

Update all instances of <DocumentImage> that are available in SOE which are not completely equal to the <DocumentImage> given as input parameter.

#### **If the updateMode = DELETE**

Delete all instances of <DocumentImage> that are available in SOE which are not parts of the <DocumentImage> given as input parameter.

### **6.5.12.3 Input**

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the document image entities.
<b>List&lt;DocumentImage&gt;: List of either all Image entities or specific DocumentImage entities – if given</b>		
name	DocumentImage.name	The Name of a DocumentImage.
- Inner List of <VersionedDocumentImage>		
versionID	VersionedDocumentImage.versionID	The binary representation of the image that is represented by the VersionedDocumentImage.
image	VersionedDocumentImage.image	Version Id of the image. This version is independent of the VersionedDocument's version id.

Table 221: AF\_UpdateDocumentImages input

### **6.5.12.4 Output**

None.

### **6.5.12.5 Exceptions**

None.

## **6.5.13 AF\_InformCustomer**

### **6.5.13.1 General Description**

This AF determines documents/emails that the customer needs to be informed with and sends them to the customer to inform him about certain events that happened, e.g. when he signed a user agreement or registered a new vehicle. This AF does not collect customers that need to be informed or documents that need to be sent. This must be done by the caller:

For instance, the information about changed user agreements is triggered by the batch **AF\_SendNewLegalDocumentsBatch**, which finds out which customers need to be informed and which user agreements have changed and calls this AF for each information that needs to be sent out. Another example is **AF\_SetUserAgreementStateOfConsent** that sets or revokes a consent given by the

---

customer for a list of user agreements and needs to inform the customer thereof. This list of affected user agreements is determined by the caller, not this AF.

### 6.5.13.2 Sequence Description

#### Step 1: Determine Output Type

If `<UserType.isOfflineUser>` is FALSE, the output type email. Otherwise the output type is letter.

#### Step 2: Determine Main and Attached Documents

Determine the `<DocumentTriggerEnum>`, by mapping the given `<PerformedActionEnum>` against the corresponding `<DocumentTriggerEnum>`.

DocumentTriggerEnum / PerformedAction	VEHICLE_REGISTERED	VEHICLE_SEPARATED	USER AGREEMENT ACCEPTED	USER AGREEMENT CHANGED	USER AGREEMENT REVOKED
USER AGREEMENT ACCEPTED			X		
USER AGREEMENT REVOKED					X
USER AGREEMENT CHANGED				X	
VEHICLE_REGISTER	X				
VEHICLE_SEPARATE		X			

Table 222: DocumentTriggerEnum and PerformedActionEnum mapping

Call `AF_GetDocumentsForDocTrigger` (see 6.5.24 `AF_GetDocumentsForDocTrigger`), with the given input parameters to retrieve the main document plus attachments:

- The determined `<DocumentTriggerEnum>`
- The determined communication channel from step 2 as `<OutputTypeEnum>`
- List of given `<UserAgreement>`, omit if no list was given

Retrieve the highest version `<VersionedDocument>` of the returned main document, for which the `<VersionedDocument.validFrom>` is equal or smaller than today.

#### Step 3: Generate Attachments

If the output type is email:

For all attached user agreements (if available), call `AF_GenerateDocuments` with the parameter `requestDocumentBundle = FALSE` to generate the documents, with the parameters:

- The list of `<DocumentXmlDefinition, locale>` of the given attachments (input parameter Map<user Agreement, locale>)
- The given `<UserType>`
- Omit vehicle information
- Omit org city information
- `DocumentTriggerEnum` as the condition
- `requestDocumentBundle = FALSE`

- If the country of the customer has the ISO CODE equal to “DE” then:
  - sentFromCountry = “DE”
  - localeForAddressCountry = “de\_DE”
- else
  - sentFromCountry = <leave empty>
  - localeForAddressCountry = <leave empty>

Compose the file name for the attachments as

- <VersionedDocument.title> + “\_v” + <VersionedDocument.versionId> + “.pdf”

The document title is determined using the locale. If no document title exists for the given locale, then the <Document.DocumentId> is used instead of the title.

#### If the output type is letter:

Call AF\_GenerateDocuments to generate all documents (=legal documents) that are appended to the cover letter with the parameters:

- The <DocumentXmlDefinition> of the given attachments
- The preferred language of <UserType> as the locale
- The given <UserType>
- Omit vehicle information
- Omit org city information
- DocumentTriggerType as the condition
- requestDocumentBundle = TRUE
- If the country of the customer has the ISO CODE equal to “DE” then:
  - sentFromCountry = “DE”
  - localeForAddressCountry = “de\_DE”
- else
  - sentFromCountry = <leave empty>
  - localeForAddressCountry = <leave empty>

#### **Step 4: Fill the necessary contentFields**

If DocumentTriggerType = USER AGREEMENT CHANGED then:

For all user agreements collected above, the associated summary changes need to be extracted to fill the contentField <legalChangesSummary>. This is done by calling AF\_CollectSummaryOfLegalChanges (see 6.5.34) with the following parameters:

- For user agreements and locale(s):
  - list of user agreements (forward input parameter of this AF) and locale

- 
- (preferred language of <UserType> for output type email)
  - list of user agreements and the locales in which they were signed (forward input parameter Map<user Agreement, locale>) for output type letter
  - country (forward input parameter of this AF)
  - docType = {"PDF" if outputType is "LETTER" or "TXT" if outputType is "EMAIL"}

If DocumentTriggerType = USER AGREEMENT CHANGED, USER AGREEMENT ACCEPTED or USER AGREEMENT REVOKED then:

For all user agreements collected above, the associated document titles are summarized by calling AF\_CollectLegalTitles (see chapter 6.5.35) with the following parameters:

- list of user agreements (forward input parameter of this AF)
- locale (preferred language of <UserType>)

and by saving the returned parameter in the contentField <titleOfLegalDocuments>.

### **Step 5: Generate main document and Send all generated Documents to the end customer**

The generation and sending of documents slightly differs depending on whether the customer is reachable via email (channel email) or only via mail (channel letter).

Independently on the channel, generate the reference number <referenceNumber> for the documents to be send.

#### **If the output type is email:**

Call AF\_GeneratePDF (see chapter 6.5.15) to generate the email that needs to be sent out, with the parameters:

- The <DocumentXmlDefinition> of the given <VersionedDocument> of the main document
- The preferred language of <UserType> as the locale
- The given <UserType>
- The given vehicle configuration
- DocumentTriggerEnum as the condition

Call the interface IIF\_SendEmail to send an email to the customer, with the input:

- <UserType.email> as the recipient together with the recipient's full name:  
<UserType.firstName> + " " + <UserType.lastName1>
- The subject and body from the output of AF\_GenerateEmail
- The list of generated user agreements as the email attachments (see list of attachments generated at step 4, associated to output type = email).
- Timestamp when the file was generated
- <UserType.userID>
- referenceNumber

- 
- The list of values that identify the templates based on which the documents (main + attachments) have been generated: List of <document Template IDs, document versions>
  - Forward the input parameter “performedAction” mapping it like this:
    - CommunicationLogBookMetaData.key = “TriggerAction”
    - CommunicationLogBookMetaData.value = performedAction
  - Forward the input parameter “Vehicle Configuration” (if available) mapping it like this:
    - CommunicationLogBookMetaData.key = “Fin”
    - CommunicationLogBookMetaData.value = VehicleConfiguration.fin

**If the output type is letter:**

Call IIF\_IsMBconnectCountry with < ADDRESS\_COUNTRY> as country to check if the user’s address country is supported by Mercedes Connect me or not. If the result is true, continue. If the result is false return.

Call AF\_GenerateDocuments to generate the main document, with the parameters:

- The <DocumentXmlDefinition>
- The preferred language of <UserType> as the locale
- The given <UserType>
- Omit vehicle information
- Omit org city information
- DocumentTriggerType as the condition
- requestDocumentBundle = false
- sentFromCountry = “DE”
- localeForAddressCountry = “de\_DE”
- List of XML-Definition of attachments
- requestDocumentBundle = TRUE.

Call the AF\_MergePDFs with the list of generated documents (main document + attachments).

Name the merged document after the above generated reference number.

Call the interface IIF\_SendLetter, with the input:

- List of generated PDF document
- Timestamp when the file was generated
- Timestamp when the customer (address) has been fetched from CPD
- <UserType.userID>
- The list of values that identify the templates based on which the documents (main document+ attachments) have been generated: List of <document Template IDs,

- document versions>
- referenceNumber
  - Forward the input parameter “performedAction” mapping it like this:
    - CommunicationLogBookMetaData.key = “TriggerAction”
    - CommunicationLogBookMetaData.value = performedAction
  - Forward the input parameter (if available) “Vehicle Configuration” mapping it like this:
    - CommunicationLogBookMetaData.key = “fin”
    - CommunicationLogBookMetaData.value = VehicleConfiguration.fin

### 6.5.13.3 Input

Name	Type / Length / BOM	Description
Performed Action	PerformedActionEnum	The performed action that requires a certain set of documents. Necessary to determine the particular set of documents.
Map<user Agreement, locale>	Map<UserAgreement, LocaleDT>	Specifies the possible list of user agreements (together with the locales in which they were signed), which depending on the configuration, may be attached to the main document/email that is sent out. This is optional, because not every document/email requires user agreements attached.
User	UserType	The user's profile.
Vehicle	VehicleConfiguration	The vehicle information that appears in the document/email. This is optional, because not every email/document requires vehicle information.

### 6.5.13.4 Output

None.

### 6.5.13.5 Exceptions

- If the given <DocumentTriggerEnum> is unknown, return the error GENDOC\_007

## 6.5.14 AF\_InformCustomerOfChangedServiceAvailability

This application function logs messages regarding changes of service availabilities or changes of licenses for a user and his or her vehicle.

### 6.5.14.1 Sequence Description

Based on the given triggerType log the following info messages:

- TriggerType = SERVICE\_DISABLED: Removed availability of service *serviceID* for vehicle *F/N* and user *userID*.
- TriggerType = SERVICE\_ENABLED: Removed availability of service *serviceID* for vehicle *F/N* and user *userID*.
- TriggerType = LICENSE\_REMOVED: Removed license of service *serviceID* for vehicle *F/N* and user *userID*.
- TriggerType = LICENSE\_ADDED: Added license of service *serviceID* for vehicle *F/N* and user *userID*.

### 6.5.14.2 Input

Name	Type / Length /	Description
------	-----------------	-------------

	<b>BOM</b>	
userID	UserProxy.userID	The id of the customer to be informed.
vehicle	VehicleProxy.fin	The vin/fin of the vehicle for which the service availability has changed.
serviceID	Service.ServiceID	The id of the service that either is additional available or not available anymore for the customers vehicle.
triggerType	String	The trigger event for calling the internal interface. (Possible values: SERVICE_ENABLED, SERVICE_DISABLED, LICENSE_REMOVED, LICENSE_ADDED)

Table 223: AF\_LogChangedServiceAvailability Input

#### 6.5.14.3 Output

None.

#### 6.5.14.4 Exceptions

None.

### 6.5.15 AF\_GeneratePDF

This AF generates a document (e.g. user agreement) as PDF in the language of the given locale.

Note: This AF supports the generation of blank documents.

#### 6.5.15.1 Sequence Description

##### *Step 1: Retrieve Templates*

Retrieve the localized `<DocumentTemplate>` that is specified within `<DocumentXmlDefinition>`. Do this by using the given input locale and if a user is given the address country of the input `<UserType>` to retrieve the correctly localized `<DocumentTemplate>`. If several templates are using the same document name, the one with the most matching language and country settings will be retrieved.

##### *Step 2: Retrieve Font Styles*

Retrieve the font style sheet and look for each style used in the `<DocumentXmlDefinition>` based on the input “locale”. If the font style cannot be found, use the default font style specified in the font style sheet.

If font style sheet cannot be found, the error GENDOC\_002 is returned.

##### *Step 3: Retrieve Dynamic Content*

Retrieve and expand each referenced localized `<CustomTag>` and `<DocumentBlock>`. The expansion is done by replacing the `<CustomTag>` and `<DocumentBlock>` with the content that was defined in the retrieved `<CustomTag>` and `<DocumentBlock>`.

Exception to this rule applies if a user is given for the `<CustomTag>` where the `<CustomTag.tagType>` is of `<CustomTagType.CUSTOMER_ADDRESS_SHORT>` or `<CustomTagType.CUSTOMER_ADDRESS_LONG>`. In this case:

- If the input parameters <sentFromCountry> and < localeForAddressCountry> are given, then call AF\_GetFormattedAddress (forward the parameters <sentFromCountry> and < localeForAddressCountry>).

**Else**

- Expand and replace the custom tag with the content that was defined for it.

Retrieve the dynamic content to substitute each <DataField>. Dynamic content can be vehicle data, customer data or contract data. If there is no dynamic content for a certain <DataField>, the field will be omitted.

If while resolving the content of a <block/> element or of a referenced <DocumentBlock> a nested DocumentImage is referenced that cannot be find, than throw the error GENDOC\_012 and return. In the case of vehicle data, if a vehicle reference is needed and the VIN is available, then the VIN will be used otherwise the VIN will be used.

If one of the XML tags has a condition attribute specified that does not match the given condition input, then omit this XML tag.

#### **Step 4: Generate Document**

The <DocumentTemplate> is set as a background. All dynamic content, which was defined in the <DocumentXmlDefinition> is rendered upon the <DocumentTemplate> in accordance to the font style sheet.

The static content will be resolved and returned as an output parameter:  
each referenced XML Element:

```
<staticAttachment templateId= "" templateVersion="##" />
```

will be added to the output list in form of the following triple: <Binary PDF, templateID, templateVersion>

#### **6.5.15.2 Input**

Name	Type / Length / BOM	Description
documentReferenceNumber	(optional) String	Content for the documentReferenceNumber data field.
<b>List of attributes</b>		
Document XML Definition	DocumentXmlDefinition	The document xml definition contains structural and content information for how to generate the document.
Locale	Locale	Specifies the language the document is supposed to be created in.
User	UserType	The user information is printed onto the document. This is optional, because not every document requires user information.
Vehicle	Vehicle	The vehicle information that is printed onto the document. This is optional, because not every document requires vehicle information.
Org City	String	The org. city is used to print the outlet location onto the document. This is optional, because not every document requires the outlet location.
Condition	String	Some XML tags are only resolved when a certain condition is fulfilled.
sentFromCountry	-	The country for which the address is formatted such that the address can be handled by the post in that country
localeForAddressCountry	-	The locale in which the users address country is

		translated, such that the address can be handled by the post in that country
--	--	--

Table 224: AF\_GeneratePDF - Input

### 6.5.15.3 Output

Name	Type / Length / BOM	Description
GeneratedPDF	Binary PDF	The generated document as a binary PDF.
<b>List of Attachments</b>		
generatedAttachment	Binary PDF	The static attachment document in form of a binary PDF
templateID	String	The ID of the template
templateVersion	String	The version of the template

Table 225: AF\_GeneratePDF - Output

### 6.5.15.4 Exceptions

- If font style sheet cannot be found, the error GENDOC\_002 is returned.
- If referenced pictures cannot be found, the error GENDOC\_12 is returned.

## 6.5.16 AF\_MergePDFs

This AF merges several documents (e.g. user agreements) into one combined PDF.

### 6.5.16.1 Sequence Description

#### Step 1: Merge Documents

If the input parameter <startOnFrontPage> is TRUE then make sure that each merged part document starts on an even page:

- loop through all documents received as input parameter:
- if the number of pages of the part document is uneven, then add an empty page at the end of this document

Finally, merge all sub documents into one combined PDF in their order of appearance.

### 6.5.16.2 Input

Name	Type / Length / BOM	Description
Documents	List of Binary PDFs	List of documents (e.g. user agreements) as several binary PDFs.
startOnFrontSide	boolean	TRUE if at merging each document needs to start on even pages. FALSE otherwise

### 6.5.16.3 Output

Name	Type / Length / BOM	Description
GeneratedPDF	Binary PDF	The merged document as a binary PDF.

### 6.5.16.4 Exceptions

None.

---

## **6.5.17 AF\_RetrieveVersionedDocumentElementsForUser**

This AF retrieves for the given element the available versions for the user.

### **6.5.17.1 Sequence Description**

Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter 17.4.1). If the error SESSION\_001 is thrown then load and return from the database all associated versioned child elements in their productive form. Execution stops here.

Checks if there is an association between the document element given as input parameter and a change session:

If the element is inside the change session:

1. If the element is inside the change session of the user then load from the database all associated versioned child elements: if among the child elements some are modified inside the change session then load these ones over their productive form and return the result.
2. If the element is inside the change session of another user then:
  - (i) If the element has a productive form then load all associated versions of the element in their productive form and return the result.
  - (ii) If the element hasn't a productive form then throw exception: DOCSESSION\_002. Execution stops here and returns the result.

Else load and return from the database all associated versioned child elements in their productive form.

### **6.5.17.2 Input**

Name	Type / Length / BOM	Description
documentElement	One of the following instance: Document, DocumentTemplate, DocumentBlock, CustomTag	Parent document element for which the child elements need to be loaded.
loggedOnUser	String	Current logged on user

### **6.5.17.3 Output**

Name	Type / Length / BOM	Description
<b>List&lt;document elements&gt;</b>		
versionedDocumentElement	One of the following instance: VersionedDocument, VersionedDocumentTemplate, VersionedDocumentBlock, CustomTag	Child elements to be displayed to the user.

### **6.5.17.4 Exceptions**

Deny the request to load the versions of a document element that is currently being created by other users comes, by throwing DOCSESSION\_002.

---

## **6.5.18 AF\_GetSupportedMBconnectCountryLocales**

This AF retrieves the locales for all supported MBconnect countries.

### **6.5.18.1 Sequence Description**

- Step1: Fetch maintained locales from the application configuration (see PROP\_LOCALES)
- Step2: From each maintained locale extract the ISO country code and the ISO language code.
- Step3: Get the list of MBconnect Countries by calling IIF\_GetMBconnectCountries. For each MBconnect country extract the ISO language codes from step 2 (Based on the ISO country codes of the MBconnect country).

### **6.5.18.2 Input**

None.

### **6.5.18.3 Output**

Name	Type / Length / BOM	Description
countryCode	String	The country code according to ISO 639-1
languageCode	String	The language code according to ISO 3166 Alpha 2

### **6.5.18.4 Exceptions**

None.

---

## **6.5.19 AF\_RetrieveCountryAndLocaleCombination**

From the maintained locales this AF retrieves all combinations between the country and the language.

### **6.5.19.1 Sequence Description**

- Step 1: Retrieve maintained locales for all supported MBconnect countries by calling AF\_GetSupportedMBconnectCountryLocales
- Step 2: For each locale retrieved in Step1 resolve the country and language names and generate the return following string: <country> + "(" + <language> + ")"

Example: From the following locales:

Locale
German (Germany)
German (Switzerland)
German (Austria)
English (US)
English (Switzerland)
French (France)
French (Belgium)

---

return this:

Return Result
Germany(German)
Switzerland(German)
Switzerland(English)
Austria(German)
US(English)
France (French)
Belgium(French)

*Note: US is not in the return list, as it is not a MBconnect country.*

#### 6.5.19.2 Input

None.

#### 6.5.19.3 Output

Name	Type / Length / BOM	Description
reversedLocaleText	String	<country name> + "(" + <language name> + ")"
countryCode	String	The country code according to ISO 639-1
languageCode	String	The language code according to ISO 3166 Alpha 2

#### 6.5.19.4 Exceptions

None.

### 6.5.20 AF\_GenerateEmail

This AF generates an email (e.g. vehicle separation information) in the language of the given locale. The generated email is plain text only and does not contain font styles, templates or images.

#### 6.5.20.1 Sequence Description

##### **Step 1: Retrieve Dynamic Content**

Retrieve and expand each referenced localized `<CustomTag>` and `<DocumentBlock>`. The expansion is done by replacing the `<CustomTag>` and `<DocumentBlock>` with the content that was defined in the retrieved `<CustomTag>` and `<DocumentBlock>`.

If while resolving the content inside the email XML definition (e.g. `<block/>`) or if inside a referenced `<DocumentBlock>` a nested DocumentImage is referenced, than log the error GENDOC\_011 and continue.

Retrieve the dynamic content to substitute each `<DataField>`. Dynamic content can be vehicle data, customer data or contract data. If there is no dynamic content for a certain `<DataField>`, the field will be omitted.

If one of the XML tags has a condition attribute specified that does not match the given condition input, then omit this XML tag.

##### **Step 2: Generate Email**

---

With the dynamic content in place and any possible attachments included, the email header and body are now generated in accordance to the email XML definition.

The document's header is used for the email's subject field.

The document's body is used to compose the email's body.

### 6.5.20.2 Input

Name	Type / Length / BOM	Description
Email XML Definition	<DocumentXmlDefinition>	The email definition that describes how the email is to be built.
Locale	Locale	Specifies the language the email is supposed to be created in.
User	UserType	The user information is printed onto the document. This is optional, because not every document requires user information.
Vehicle	Vehicle	The vehicle information that is printed onto the document. This is optional, because not every document requires vehicle information.
Condition	String	Some XML tags are only resolved when a certain condition is fulfilled.
documentReferenceNumber	(optional) String	content for the documentReferenceNumber data field

### 6.5.20.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Subject	Mand.	String	-	-		The email's subject
Body	Mand.	String	-	-		The email's body.

### 6.5.20.4 Exceptions

None.

## 6.5.21 AF\_BuildLegalDocuments

This AF retrieves and returns the appropriate legal documents (user agreements including the terms and conditions) for the specified user. This is only for creating legal documents that need to be signed (DocumentUsageEnum.FOR\_SIGNATURE) or displayed for informational purposes (DocumentUsageEnum.FOR\_DISPLAY) only. This is not for creating legal documents that have been signed and sent out as a confirmation.

### 6.5.21.1 Sequence Description

#### **Step 1: Retrieve Customer Data**

Call the internal interface IIF\_GetUserProfile with the given user ID to retrieve the customer data <UserType>.

#### **Step 2: Determine User Agreement Version**

Repeat for each given user agreement ID:

If an entry exists for which <UserAgreementConsent.agreementStatus> is "ACCEPTED", then the user is considered an "existing customer" for the particular user agree-

---

ment, otherwise he is considered a “new customer”. (This information is passed to this AF by input parameter.)

If the customer is classified as a “new customer” for a given user agreement ID, retrieve the highest version of <UserAgreement> for which the <UserAgreement.validFrom> is equal or smaller than today and <UserAgreement.enabled> is TRUE.

If no productive form available (<UserAgreement> has no association to a ChangeSession), then retrieve the next highest version that fulfils the criteria’s above.

If the customer is classified as an “existing customer” for a given user agreement ID, retrieve the productive form of the associated <UserAgreement>.

### **Step 3: Determine DocumentTriggerEnum**

For each determined <UserAgreement> from step 3, the correct DocumentTriggerEnum needs to be determined. This is done with the help of the given <DocumentUsageEnum>.

If the given <DocumentUsageEnum> is:

- “FOR\_SIGNATURE”, then the DocumentTriggerEnum is US-ER AGREEMENT FOR SIGNING
- “FOR\_DISPLAY”, then the DocumentTriggerEnum is US-ER AGREEMENT LOOKUP

### **Step 4: Generate Documents**

For all determined <UserAgreement> from step 3, call AF\_GenerateDocuments with the parameters: - List of

- The <DocumentXmlDefinition> of the <UserAgreement>
- The given locale
- The <UserType>
- org city
- The determined DocumentTriggerEnum as the condition
- <sentFromCountry> = leave empty
- <localeForAddressCountry> = leave empty

and the actual input parameter requestDocumentBundle.

Return the list of documents as they have been retrieved from the previous call.

#### **6.5.21.2 Input**

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
User-Id	Mand.	String		User.userID		The user ID of the customer. The ID is necessary to retrieve customer information (e.g name) that is printed onto the documents.
Locale	Mand	String	5	Used as input pa-	Examples:	Locale (either a language or

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
	.			parameter to determine document language (AF_GeneratePDF)	de_DE, de_AT, fr_CH, de_CH, en	a language in combination with a country). Specifies which language the document is supposed to be created in.
Outlet City	Opt.	String	-	-	Example: "Stuttgart"	The city of the outlet, at which the customer signs the user agreements.
Document Usage	Mand.	String		DocumentUsageEnum	Example: "FOR_SIGNATURE", "FOR_DISPLAY"	The usage specifies the purpose of the document, for instance if the document is supposed to be signed or just used to look/display the document as an information.
RequestDocumentBundle	Mand.	Boolean		-	true, false	RequestDocumentBundle determines whether the requested documents are returned separately (false) or merged as one PDF (true).
<b>List of user agreement IDs</b>						
User Agreement ID	Mand.	String	50	Document.documentID	Example: "UserAgreement-BsicServices"	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.
User Agreement Version	Mand.	int		VersionedDocument.versionID	Examples: 1, 12356, 1234567890	The user agreement version that the binary PDF corresponds to.

Table 226: AF\_GetLegalDocuments Input

### 6.5.21.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of requested legal documents (user agreements incl. the terms and conditions + version)						
Document	Mand.	Binary PDF	-	Binary PDF	-	The requested legal document in PDF format. It contains the user agreement and the terms and conditions as one binary PDF. In case that the input parameter "RequestDocumentBundle" is TRUE, then only distinct versions of the terms and conditions will be appended at the end of the generated document.
<b>List of user agreement (id + version)</b>						
User Agreement ID	Mand.	String	50	Document.documentID	Examples: "UserAgreement-BsicServices"	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.
User Agreement Version	Mand.	int		VersionedDocument.versionID	Examples: 1, 12356, 1234567890	The user agreement version that the binary PDF corresponds to.

Table 227: AF\_GetLegalDocuments Output

#### 6.5.21.4 Exceptions

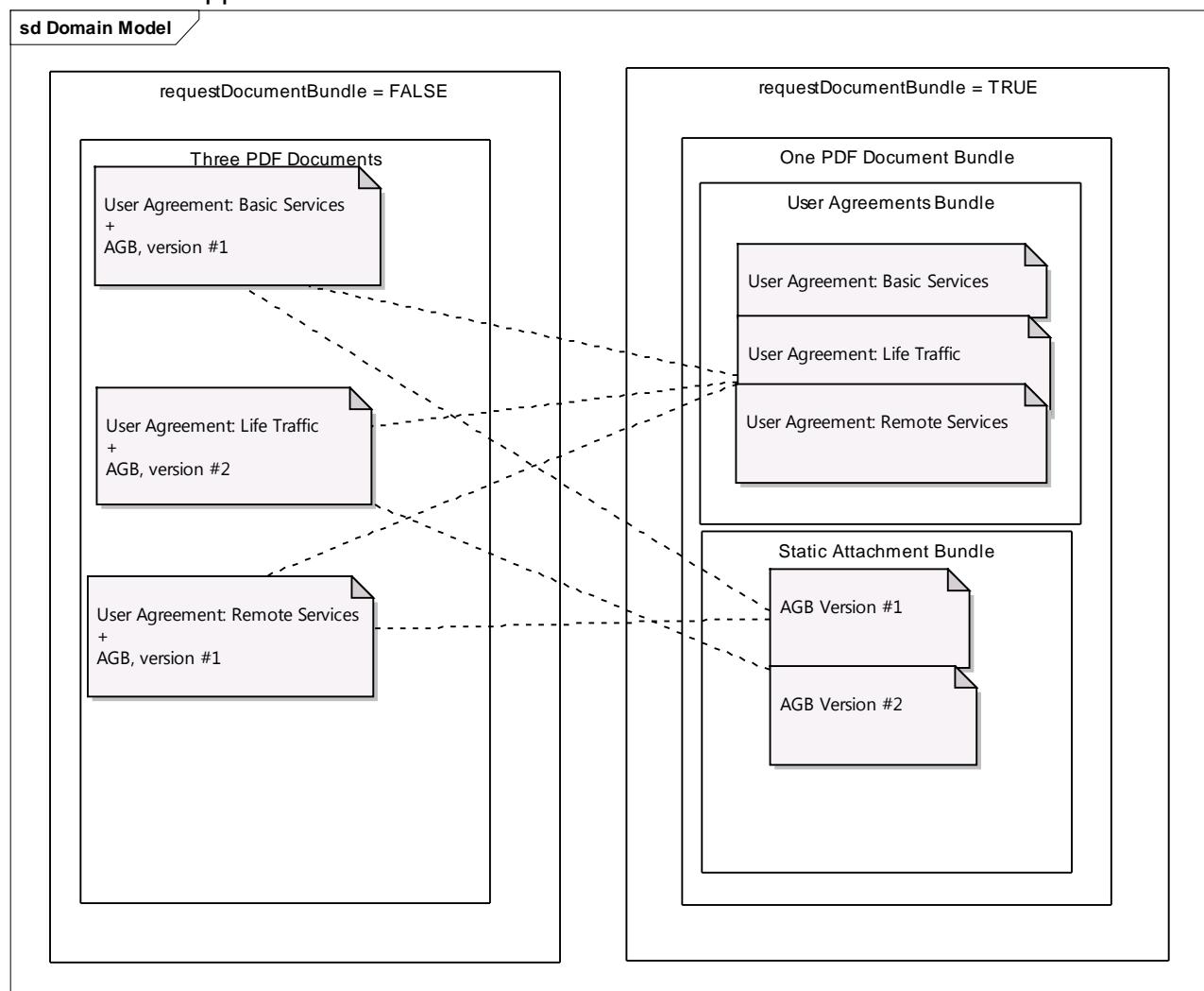
- If the user ID does not exist, the error ACCDAS\_002 is returned
- If the user agreement ID does not exist, the error GETDOC\_003 is returned

#### 6.5.22 AF\_GenerateDocuments

This AF generates the documents in the mode as they have been requested by the caller.

If the caller uses the parameter “requestDocumentBundle” = false, then each document will be merged only with the static attachment that it is referencing .

If the caller uses the parameter “requestDocumentBundle” = true, then all documents are merged into one PDF and only distinct static attachments referenced by all documents will be appended to the above PDF.



##### 6.5.22.1 Sequence Description

###### **Step 1: Retrieve Static Attachments as PDFs for each Document**

For each DocumentXMLDefinition (main document):

- call AF\_GeneratePDF (called with the input parameters of this AF) to generate the document and the associated static attachments documents.

In case the input parameter “requestDocumentBundle” is true then continue with step 2, otherwise for each main document and the list of attachments (returned by the previous call AF\_GeneratePDF) call AF\_MergePDFs. Return with the list of generated documents.

### **Step 2: Identify the static attachments that can be shared by multiple Documents**

In this step all distinct (unique) attachments that are being referenced from within the documents are determined. Two static attachments are identical, if their templateID, their versions (retrieved as output parameters from a previous call AF\_GeneratePDF) and the locale are equal.

### **Step 3: Merge all main documents and all static attachments to one PDF document**

Call AF\_MergePDFs with the list of all main documents and with all distinct static attachments binaries found at step 2.

#### **6.5.22.2 Input**

Name	Type / Length / BOM	Description
requestDocumentBundle	boolean	This parameter controls the output of this AF:  If set to TRUE, then all distinct static attachments across all documents are merged at the end of the merged PDF bundle If set to FALSE, then for each document, the associated static attachment will be returned
<b>List of attributes &lt; Document XML Definition, Locale&gt;</b>		
Document XML Definition	DocumentXmlDefinition	The document xml definition contains structural and content information for how to generate the document.
Locale	Locale	Specifies the language the document is supposed to be created in.
User	UserType	The user information is printed onto the document. This is optional, because not every document requires user information.
Vehicle	Vehicle	The vehicle information that is printed onto the document. This is optional, because not every document requires vehicle information.
Org City	String	The org. city is used to print the outlet location onto the document. This is optional, because not every document requires the outlet location.
Condition	String	Some XML tags are only resolved when a certain condition is fulfilled.
sentFromCountry	-	The country for which the address is formatted such that the address can be handled by the post in that country
localeForAddressCountry	-	The locale in which the users address country is translated, such that the address can be handled by the post in that country

#### **6.5.22.3 Output**

Name	Type / Length / BOM	Description
GeneratedAttachmentAcrossDocuments	Binary PDF	(optional) All static attachments referenced from the

		XMLDocumentDefinitions as a single PDF. This parameter is returned if the input parameter mergeStaticAttachmentsAcrossDocuments = TRUE
<b>List of Binary PDFs (List&lt;GeneratedMainPDF, GeneratedAttachment&gt;)</b>		<b>List of generated PDF Documents</b>
GeneratedMainPDF	Binary PDF	The generated document as a binary PDF. (optional)
GeneratedAttachment	Binary PDF	All static attachments referenced by the main document. This parameter is returned if the input parameter mergeStaticAttachmentsAcrossDocuments = FALSE

#### 6.5.22.4 Exceptions

None.

### 6.5.23 AF\_GetDocumentForVehicleSeparation

This AF returns a document which, when signed by the customer, authorize the retailer to separate the customer's vehicle from its linked account.

#### 6.5.23.1 Sequence Description

##### **Step 1: Determine Document**

Call AF\_GetDocumentsForDocTrigger, with the input parameters "VehicleRequestSeparationAuthorization" as <DocumentTriggerEnum> and "ONLINE\_PDF" as <OutputTypeEnum> to determine the documents.

##### **Step 2: Determine Document Version**

Retrieve the highest version <VersionedDocument> of the document, for which the <VersionedDocument.validFrom> is equal or smaller than today.

##### **Step 3: Retrieve Vehicle Information**

Call IIF\_FetchVehicleData with the given parameters FIN and locale to retrieve the vehicle data <Vehicle>.

##### **Step 4: Retrieve Customer Data**

Call the internal interface IIF\_GetUserProfile with the given user ID to retrieve the customer data <UserType>.

##### **Step 5: Generate Documents**

For all <VersionedDocument> entities from Step 2, call AF\_GenerateDocuments, with the input parameters <DocumentXmlDefinition>, locale, the user ID and <Vehicle> to generate the particular document and the input parameter requestDocumentBundle=TRUE to merge the documents to a single PDF.

#### 6.5.23.2 Input

See external interface IF\_SOE\_GetDocumentForVehicleSeparation.

#### 6.5.23.3 Output

See external interface IF\_SOE\_GetDocumentForVehicleSeparation.

---

#### 6.5.23.4 Exceptions

See external interface IF\_SOE\_GetDocumentForVehicleSeparation.

#### 6.5.24 AF\_GetDocumentsForDocTrigger

This AF returns each <Document> that is assigned to the given <DocumentTriggerEnum> and is of the given <OutputTypeEnum>.

##### 6.5.24.1 Sequence Description

###### **Step 1: Determine Output Assignments**

Determine the <OutputTypeAssignment>, where <OutputTypeAssignment.OutputTypeEnum> corresponds to the given <OutputTypeEnum> and <OutputTypeAssignment> is associated with <Trigger2DocAssignment>, where <Trigger2DocAssignment.triggerType> corresponds to the given <DocumentTriggerEnum>.

Depending on the <OutputTypeEnum>, each <OutputTypeAssignment> is either a <LetterOutputAssignment>, <EmailOutputAssignment> or <EDocumentOutputAssignment>.

###### **Step 2: Determine Documents**

Determine the associated <Document> for the <OutputTypeAssignment> from Step 1, which is returned as the main document.

If the <OutputTypeAssignment> is either a <LetterOutputAssignment> or <EMailOutputAssignment> and has the attribute “addRelevantUserAgreements” set to TRUE, then return any given user agreement as list of attachments.

#### 6.5.24.2 Input

Name	Type / Length / BOM	Description
DocumentTriggerEnum	DocumentTriggerEnum	The document trigger type defines the performed action that requires a certain set of documents. Necessary to determine the particular set of documents.
OutputTypeEnum	OutputTypeEnum	Specifies how the documents are assigned to the DocumentTriggerEnum.
List of User Agreements	List of UserAgreement	Specifies the possible list of user agreements, which are returned in the output if the configuration for DocumentTriggerEnum and OutputTypeEnum is set to include relevant user agreements.

Table 228: AF\_GetDocumentsForDocTrigger Input

#### 6.5.24.3 Output

Name	Type / Length / BOM	Description
Main Document	Document	The main document associated with the <DocumentTriggerEnum>. This can either be of the <DocumentTypeEnum> “PDF” or “PLAINTEXT/EMAIL”
List of Attachments	List of UserAgreement	(optional) Additional documents that come along with the main document. This is optional, because not every main document has attachments.

---

Table 229: AF\_GetDocumentsForDocTrigger Output

#### 6.5.24.4 Exceptions

If no matching output assignment where found, GENDOC\_009 is raised and the processing is aborted.

If no matching documents for an output assignment where found, GENDOC\_008 is raised and the processing is aborted.

#### 6.5.25 AF\_GetFormattedAddress

This AF returns the formatted address of the given customers based on their country.

##### 6.5.25.1 Sequence Description

###### *Step 1: Retrieve Custom Tag*

If the user type does not contain a country, then:

Retrieve the <CustomTag> where <CustomTag.tagType> is of <CustomTagType.CUSTOMER\_ADDRESS\_SHORT> and <CustomTag.country>=="" (i.e. "ALL") and <CustomTag.DocumentTypeEnum> is of <DocumentTypeEnum.PDF>.

If the given <AddressTypeEnum> is of <AddressTypeEnum.SHORT>:

Retrieve the <CustomTag> where <CustomTag.tagType> is of <CustomTagType.CUSTOMER\_ADDRESS\_SHORT> and <CustomTag.country> corresponds to the address country of the customer and <CustomTag.DocumentTypeEnum> is of <DocumentTypeEnum.PDF>.

If the given <AddressTypeEnum> is of <AddressTypeEnum.LONG>:

Retrieve the <CustomTag> where <CustomTag.tagType> is of <CustomTagType.CUSTOMER\_ADDRESS\_LONG> and <CustomTag.country> corresponds to the address country of the customer and <CustomTag.DocumentTypeEnum> is of <DocumentTypeEnum.PDF>.

If the country specific <CustomTag> cannot be found, return the error GENDOC\_006.

###### *Step 2: Retrieve Dynamic Content*

If the given <AddressTypeEnum> is of <AddressTypeEnum.SHORT>, then omit each name related <DataField>.

If the input parameters <sentFromCountry> and <localeForAddressCountry> are given (not NULL), then the following logic applies for the customtag <CustomTagType.CUSTOMER\_ADDRESS\_SHORT>:

If the country of the customer has the ISO CODE equal to "DE" and is the same as the country from which the letter is send from (UserType.Address\_Country == <sentFromCountry> parameter == DE) then omit from address the datafield DataFieldType.CUST\_ADDRESS\_COUNTRY

else

1. Get the localized name of the address country of the customer from the attribute MbcCountry.name. The entity is referenced by using this parameters:
  - MbcCountry.countryCode = UserType.Address\_Country
  - <I18N> = <localeForAddressCountry> (see input parameter of this AF)The name must be transformed to uppercase.
2. Transform the name of the city to uppercase.

Retrieve the dynamic content to substitute each <DataField> within the <CustomTag> from step 1. If there is no dynamic content for a certain <DataField>, the field will be omitted.

Use the input parameter “Locale” to find the correct internationalized output string.

#### **Step 3: Repeat**

For each customer from the input, remember the matchingCode and obtain the formatted address from steps 1-2. If no more customers are given, return the aggregated list. If the matchingCode from the input is empty, return an empty matchingCode (=no error).

#### **6.5.25.2 Input**

See external interface IF\_SOE\_GetFormattedAddress.

#### **6.5.25.3 Output**

See external interface IF\_SOE\_GetFormattedAddress.

#### **6.5.25.4 Exceptions**

See external interface IF\_SOE\_GetFormattedAddress.

### **6.5.26 AF\_GetFormattedDaimlerAddress**

This AF returns the formatted address of Daimler localized to the specified locale. The used address can be configured to the system.

#### **6.5.26.1 Sequence Description**

##### **Step 1: Retrieve Custom Tag**

Retrieve the <CustomTag> where <CustomTag.tagType> is of <CustomTagType.DAIMLER\_ADDRESS> and <CustomTag.country> corresponds to the passed country as input parameter and <CustomTag.DocumentTypeEnum> is of <DocumentTypeEnum.PDF>.

If the country specific <CustomTag> cannot be found, return the error GENDOC\_006.

##### **Step 2: Retrieve Dynamic Content**

Retrieve the dynamic content to substitute each <DataField> within the <CustomTag> from step 1. If there is no dynamic content for a certain <DataField>, the field will be omitted.

Use the input parameter “Locale” to find the correct internationalized output string.

## 6.5.26.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Country	Mand.	String	2	-	Example: “DE”	Specifies the country for which the address is needed. (Note: As there is only one Daimler address, this parameter influences the custom tag that is used for formatting.)
Locale	Mand.	String	5	Used as input parameter to determine output language	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country).

Table 230: IF\_SOE\_GetFormattedAddress Input

## 6.5.26.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of Address Lines (1..n)						
Address Line	Mand.	String	-	-	Example: “Daimler AG”, “Mercedesstr . 129”, “70599 Stuttgart”	Formatted address line.

Table 231: IF\_SOE\_GetFormattedAddress Output

## 6.5.26.4 Exceptions

- If the country specific <CustomTag> cannot be found, return the error GENDOC\_006.

## 6.5.27 AF\_SaveDocumentDefinition

This AF saves the document definition.

### 6.5.27.1 Sequence Description

#### Step 1: Check XML and save document

The XML is validated against the DocType specific XSD to insure the structural validity of the XML. Further the document definition will be saved.

#### Step 2: Save MBconnect countries where the user agreement is available in

**Step 2.1:** Restrict editing the list of countries for older versions of a user agreement (perform the check only if the user agreement is enabled).

Determine the next enabled version after the version of this user agreement. If such a version exists then throw error DOCMAS\_019.

**Step 2.2:** Restrict the withdrawal of a user agreement from countries where it is already available (perform the check only if the user agreement is enabled)

---

If the document represents a user agreement, retrieve the list of available countries for the previous and current version of this versioned document.

For both versions check if the set of countries is identical to or a subset of the list of countries provided as input parameter.

If this condition is violated throw the error DOCMAS\_018.

#### **Step 2.3 Save MBconnect countries**

Assign to the associated DocumentDefinition entity the references to the given MBconnect countries

#### **6.5.27.2 Input**

Name	Type / Length / BOM	Description
Document	Document	The document and its version.
<b>List of MBconnect countries where the user agreement is available in (0..*)</b>		
Country	MbcCountry	The ISO Code of the Country

#### **6.5.27.3 Output**

None.

#### **6.5.27.4 Exceptions**

- DOCMAS\_012: if a user agreement cannot be withdraw from specific countries
- DOCMAS\_013: if modifying country availability is not allowed

### **6.5.28 AF\_SaveDocumentTemplate**

This AF saves a DocumentTemplate, which is either an already maintained or new DocumentTemplate.

#### **6.5.28.1 Sequence Description**

##### **Step 1: Apply Changes**

If the given DocumentTemplate is an existing one, then apply the changes done to the corresponding instance of <DocumentTemplate>. Otherwise create a new instance of <DocumentTemplate>.

#### **6.5.28.2 Input**

Name	Type / Length / BOM	Description
PDF	DocumentTemplate	The PDF template to be added/changed.

#### **6.5.28.3 Output**

None.

#### **6.5.28.4 Exceptions**

None.

---

## 6.5.29 AF\_SaveDocumentBlock

This AF saves a DocumentBlock, which is either an already maintained or new DocumentBlock.

### 6.5.29.1 Sequence Description

#### **Step 1: Apply Changes**

If the given DocumentBlock is an existing one, then apply the changes done to the corresponding instance of <DocumentBlock>. Otherwise create a new instance of <DocumentBlock>.

### 6.5.29.2 Input

Name	Type / Length / BOM	Description
Document Block	DocumentBlock	The document black to be added/changed.

### 6.5.29.3 Output

None.

### 6.5.29.4 Exceptions

None.

## 6.5.30 AF\_SaveDocumentImage

This AF saves either a new image, or updates an already existing one.

### 6.5.30.1 Sequence Description

#### **Verify if the user is already inside a change session**

Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser. If the error SESSION\_001 is thrown executions stops here.

#### **Check for existing images that have duplicate names**

Retrieve the document image which match the given document image for the attributes DocumentImage.name.

If such a document image exists in a productive form and is not locked in the change session of the user abort with the error message DOCMAS\_022 with DocumentImage.name for placeholder <1>.

If such a document image exists and is in the change session of the requesting user, then:

- in case that a new version is being created for an existing image (<VersionedDocumentImage>) continue
- otherwise, abort with the error message DOCMAS\_022 with DocumentImage.name for placeholder <1>.

#### **Create or update the document image**

Create or update the document image bringing it in the change session of the user (see ApplyChangeSessionOnElement) and verify if the DocumentImage is already inside a change session of another user: if yes, throw error DOCSESSION\_001. Execution stops here.

---

If the document image is created or gets a new version:

- Persist a new instance of a <DocumentImage> or a <VersionedDocumentImage>.

If the document image is updated:

- Update or persist the new <VersionedDocumentImage>.

### 6.5.30.2 Input

Name	Type / Length / BOM	Description
image	DocumentImage	The image to be added/changed.

### 6.5.30.3 Output

None.

### 6.5.30.4 Exceptions

- DOCMAS\_022: the name of the document images already exists
- DOCSESSION\_001: DocumentImage is already inside a change session of another user

## 6.5.31 AF\_GetServicesByUserAgreements

For each given user agreement this AF retrieves the covered MBconnect services.

### 6.5.31.1 Sequence Description

For each given user agreement retrieve the covered MBconnect services. If there is no service found for a certain user agreement ID, an empty list is returned for this user agreement.

### 6.5.31.2 Input

Name	Type / Length / BOM	Description
UserAgreementList	List of UserAgreement	The list of user agreements to find the services for.

Table 232: AF\_GetServicesByUserAgreements Input

### 6.5.31.3 Output

Name	Type / Length / BOM	Description
<b>List of user agreement - service assignments (a list of services exists per UserAgreement) (1..*)</b>		
UserAgreement	UserAgreement	The user agreement to find the services for.
Services	List of Services	The services covered by the requested user agreement. Can be empty.

Table 233: AF\_GetServicesByUserAgreements Output

### 6.5.31.4 Exceptions

None.

---

### **6.5.32 AF\_GetUserAgreementsByDate**

This AF determines the enabled user agreements that either have an inform date or a valid from date that lies between the given date and today. Additionally, if a previous version of the user agreement in its productive and enabled form is available, it is also returned along with the determined user agreement.

#### **6.5.32.1 Sequence Description**

If more than one date is provided as input, throw the error GETDOC\_006.

##### ***Step 1: Retrieve all versions of a User Agreement within an interval***

If the parameter “inform date by email” was given as input:

Get the instances of <UserAgreement> where:

- <UserAgreement.informByEMailDate> is bigger than the given inform date and smaller or equal today
- <UserAgreement> is available in the country given as input parameter <countryCode> (if given)
- <UserAgreement.enabled> is TRUE
- UserAgreement is in its productive form or inside the ChangeSession of the user (if given)

and fetch the associated <Document>.

If the parameter “valid from date existing customer” was given as input:

Get the instances of <UserAgreement> where:

- <UserAgreement.validFromForExistingCustomer> is bigger than the given date and smaller or equal today
- <UserAgreement> is available in the country given as input parameter <countryCode> (if given)
- <UserAgreement.enabled> is TRUE and UserAgreement is in its productive form
- UserAgreement is in its productive form or inside the ChangeSession of the user (if given)

and fetch the associated <Document>.

If the parameter “valid from date” was given as input:

Get the instances of <UserAgreement> where:

- <UserAgreement.validFrom> is bigger than the given date and smaller or equal today
- <UserAgreement> is available in the country given as input parameter <countryCode> (if given)
- <UserAgreement.enabled> is TRUE and UserAgreement is in its productive form

- UserAgreement is in its productive form or inside the ChangeSession of the user (if given)

and fetch the associated <Document>.

### **Step 2: Retrieve Previous User Agreement**

If available, get the last two versions of the instances of the <UserAgreement> found in step 1. Mark that the two <UserAgreement> versions belong together, so that they can be returned as a tuple.

Note: The “previous version” of a UserAgreement with version id  $n$  is not necessarily  $n-1$ . If  $n-1$  is not “enabled” the version  $n-1$  is not relevant and then  $n-2$  has to be checked. If  $n-2$  also does not qualify,  $n-3$  has to be considered, etc.

#### **6.5.32.2 Input**

Note, only exactly one of the following dates must be provided as input.

Name	Type / Length / BOM	Description
Inform Date By Mail	UserAgreement.informByEMailDate	The starting point for the inform date interval for emails.
Valid From Date for existing customer	UserAgreement.validFromForExistingCustomer	The starting point for the valid from date interval for an existing customer.
Valid From Date	UserAgreement.validFrom	The starting point for the valid from date interval.
countryCode	String	Optional – used as filter criteria: Country for which all available user agreements need to be determined. If no country is given then retrieve the user agreement independently of the available countries.
ChangeSession	Boolean	Optional – used as filter criteria: If not given then retrieve only the productive form of the user agreements

#### **6.5.32.3 Output**

Name	Type / Length / BOM	Description
List of User Agreements (current and previous version if available)		
Match User Agreement	UserAgreement	The user agreement that matched the interval.
Previous User Agreement	UserAgreement	The previous user agreement of the matched one.

#### **6.5.32.4 Exceptions**

Only exactly one “Date”-input parameter is allowed. If more than one is passed as input, the error GETDOC\_006 occurs.

### **6.5.33 AF\_GeneratePDFDocumentForPreview**

This AF resolves all references of the legal document on the document elements by retrieving the modified document elements inside the change session given as input parameter.

If the current logged on user has made any changes inside his change session that impacts the version of the document, then the PDF document will include these changes.

---

### 6.5.33.1 Sequence Description

#### **Step 1: Retrieve “dummy” Printable Data Fields from Property file**

Loads the data specified in the property file PROP\_PREVIEW\_DATA. Based on this data instantiate the following entities: <UserType>, <Vehicle>

#### **Step 2: Retrieve further needed input data based on specific triggers**

Call AF\_GetDocumentsForDocTrigger with the parameters:

- DocumentTriggerType = “USER AGREEMENT\_CHANGED”
- OutputType = “PDF”
- List of User Agreements = leave empty

##### Case 1:

If the returned document (output parameter “MainDocument” of the call AF\_GetDocumentsForDocTrigger) is the same as the <document> input parameter of this AF (comparison is based on the attribute Document.documentID and OutputType=“PDF”), then collect all available user agreements that fulfill the conditions:

- Latest version number: max(VersionedDocument.versionID)
- Latest version number (determined previously) is bigger than 1(VersionedDocument.versionID > 1)
- UserAgreement.enabled = true

For all user agreements collected above, the associated summary changes need to be extracted. This is done by calling AF\_CollectSummaryOfLegalChanges (see 6.5.34) with the following parameters:

- list of user agreements (retrieved above)
- country (forward input parameter of this AF)
- locale (forward input parameter of this AF)
- docType = “PDF”

Also for all user agreements collected above, the associated document titles are summarized by calling AF\_CollectLegalTitles (see chapter 6.5.35) with the following parameters:

- list of user agreements (retrieved above)
- locale (forward input parameter of this AF)

and by saving the returned parameter in the contentField <titleOfLegalDocuments>.

##### Case 2:

If the returned document (output parameter “MainDocument” of the call AF\_GetDocumentsForDocTrigger) is not the same as the <document> input parameter of this AF (comparison is based on the attribute Document.documentID and OutputType=“PDF”), then collect all available user agreements that fulfill the conditions:

- Latest version number: max(VersionedDocument.versionID)
- UserAgreement.enabled = true

For all user agreements collected above, the associated document titles are summarized by calling AF\_CollectLegalTitles (see chapter 6.5.35) with the following parameters:

- list of user agreements (retrieved above)
  - locale (forward input parameter of this AF)
- and by saving the returned parameter in the contentField  
<titleOfLegalDocuments>.

### **Step 3: Load all referenced master data elements**

*Note: the data loaded here is retrieved based on the change session parameter! In case no change session given, then load the productive form of the document entities.*

Call algorithm described inside chapter “**Loading of master data elements**” to load the proper versions of the document entities from the database (aggregated by the entity <DocumentXMLDefinition>):

- **Step 3.1: Retrieve Templates:**

- a) **If document template parameter is NULL**

Use the locale of the <UserType> and the country from the input parameter to retrieve the correctly localized <VersionedDocumentTemplate>. If several templates are using the same document name, the one with the most matching language and country settings will be retrieved.

In case that no matching locale is found, the error GENDOC\_001 is returned.

- b) **If a document template parameter is passed**

Return the passed <VersionedDocumentTemplate>

- **Step 3.2: Retrieve Font Styles:**

Retrieve the font style sheet and look for each style used in the <DocumentXMLDefinition> based on the input “locale”. If the font style cannot be found, use the default font style specified in the font style sheet. If font style sheet cannot be found, the error GENDOC\_002 is returned.

- **Step 3.3: Retrieve Dynamic Content:**

Retrieve and expand each referenced localized <VersionedDocumentBlock>. In case that no matching locale is found, the error GENDOC\_010 is returned.

Retrieve and expand each referenced <CustomTag> based on the country given as input parameter.

In case that no matching country is found, the error GENDOC\_006 is returned.

Retrieve the dynamic content to substitute each <DataField> from PROP\_PREVIEW\_DATA. If there is no dynamic content for a certain <DataField>, the field will be omitted.

If while resolving the content inside the email XML definition (e.g. <block/>) or if inside a referenced <DocumentBlock> a nested DocumentImage is referenced, than log the error GENDOC\_012 and continue.

### **Step 4: Generate Document**

The retrieved <VersionedDocumentTemplate> is set as a background. All static and dynamic content, which was defined in the <DocumentXMLDefinition> is rendered upon the <DocumentTemplate> in accordance to the font style sheet.

### 6.5.33.2 Input

Name	Type / Length / BOM	Description
version	VersionedDocument.versionID	The version of the document for which the preview is requested.
document	Document	The document for which the preview is requested
changeSession	(optional) ChangeSession	The change session of the user that wants to preview the changes done inside the change session.
locale	String	Specifies the language in which the document is generated.
country	String	Used to format the address of the customer (needed in case any CustomTagType.CUSTOMER_ADDRESS_* are referenced inside the document)
documentTemplate	VersionedDocumentTemplate	The document template which should be used instead of the document's referenced. NULL, if document's template is not overridden.

### 6.5.33.3 Output

Name	Type / Length / BOM	Description
GeneratedPDF	Binary PDF	The generated document as a binary PDF.

### 6.5.33.4 Exceptions

- If font style sheet cannot be found, the error GENDOC\_002 is returned.
- In case that no matching locale is found for a Document Block, the error GENDOC\_010 is returned.
- In case that no matching locale is found for a Document Template, the error GENDOC\_001 is returned.
- In case that no matching country is found for a CustomTag, the error GENDOC\_006 is returned.
- If referenced pictures cannot be found, the error GENDOC\_12 is returned.

## 6.5.34 AF\_CollectSummaryOfLegalChanges

This AF collects the summary of changes from the legal documents that are given as input.

### 6.5.34.1 Sequence Description

**Step1: For each user agreement collect the following data:**

- “legal document name”:  
 $<\text{VersionedDocument.title}> + \text{"_v"} + <\text{VersionedDocument.versionId}>$   
The document title is determined using the locale.
- “summary of change”:  
Retrieve and expand/resolve the content of the  $<\text{DocumentBlock}>$  referenced by the attributes:
  - $<\text{UserAgreementLegalChanges.legalChangesBlockName}>$
  - $<\text{UserAgreementLegalChanges.legalChangesBlockVersion}>$

- locale
- country
- docType
- format the above strings using the pattern below, and store intermediate result inside <legalChanges> variable:
  - “-” + <legalDocumentName> + ":" + <cr> + <summaryOfChange> + <empty line>

#### **Step2:**

Store the variable <legalChanges> into the datafield: <legalChangesSummary>.

#### **6.5.34.2 Input**

Name	Type / Length / BOM	Description
Map<userAgreement, locale>	Map<UserAgreement, Locale>	list of user agreements together with the locales in which they were signed
locale	String	Specifies the locale in which the summary changes are extracted.
country	String	Specifies the country for which the summary of changes are extracted
docType	DocumentType	Type of the document (see DocumentType enum) in which the summary of legal changes are inserted.

#### **6.5.34.3 Output**

None.

#### **6.5.34.4 Exceptions**

None.

### **6.5.35 AF\_CollectLegalTitles**

#### **6.5.35.1 General Description**

This AF collects the titles of the legal documents that are given as input.

#### **6.5.35.2 Sequence Description**

For each <UserAgreement> given as input parameter determine the <VersionedDocument.title> in the <locale> given as input parameter.

Save each <VersionedDocument.title> separated through “, “ in the parameter titleOfLegalDocuments.

Example: If User Agreement Standard Services with locale de\_DE and User Agreement Tracking Services with locale de\_DE are given as input parameter, the output parameter titleOfLegalDocuments has to be “Basisdienste, Ortungsdienste”.

### 6.5.35.3 Input

Name	Type / Length / BOM	Description
Map<userAgreement, locale>	Map<UserAgreement, Locale>	list of user agreements together with the locales in which they were signed
locale	String	Specifies the locale in which the summary changes are extracted.

Table 234: AF\_CollectLegalTitles input

### 6.5.35.4 Output

Name	Type / Length / BOM	Description
titleOfLegalDocuments	String	This attribute summarizes the titles of all user agreements given as input parameter.

Table 235: AF\_CollectLegalTitles output

### 6.5.35.5 Exceptions

None.

## 6.5.36 AF\_GetUserAgreementsByIdAndVersion

This AF returns the user agreement that has the provided Id and matching version id.

### 6.5.36.1 Sequence Description

Search for the UserAgreement that matches:

- all given input parameters
- <UserAgreement.enabled> is TRUE and UserAgreement is in its productive form

Return it if one was found. If none was found, return an empty list.

### 6.5.36.2 Input

Name	Type / Length / BOM	Description
User Agreement Id	Document.documentId	The id to identify a user agreement
User Agreement Version	VersionedDocument.versionId	The version of the user agreement
countryCode	String	Mandatory: Country for which all available user agreements need to be determined

### 6.5.36.3 Output

Name	Type / Length / BOM	Description
List of User Agreements (current and previous version if available)		
Matching User Agreement	UserAgreement	The user agreement that matched the criteria. (or none if no one could be found)

### 6.5.36.4 Exceptions

None.

---

## **6.5.37 AF\_GetDocumentForConfirmationOfVehicleRegistration**

This AF returns a document with a confirmation code. With this confirmation code the customer can confirm the pre-existing vehicle registration to his MBconnect account inside of MyMercedes.

### **6.5.37.1 Sequence Description**

#### ***Step 1: Determine Document***

Call AF\_GetDocumentsForDocTrigger, with the input parameters “VEHICLE\_REGISTER\_CONFIRMATION” as <DocumentTriggerEnum> and “ONLINE\_PDF” as <OutputTypeEnum> to determine the documents.

#### ***Step 2: Determine Document Version***

Retrieve the highest version <VersionedDocument> of the document, for which the <VersionedDocument.validFrom> is equal or smaller than today.

#### ***Step 3: Generate Documents***

For all <VersionedDocument> entities from Step 2, call AF\_GenerateDocuments, with the input parameters <DocumentXmlDefinition>, locale, the user and <Vehicle> to generate the particular document. and the input parameter requestDocumentBundle=TRUE to merge the documents to a single PDF.

### **6.5.37.2 Input**

See external interface “IF\_SOE\_GetDocumentForConfirmationOfVehicleRegistration”.

### **6.5.37.3 Output**

See external interface “IF\_SOE\_GetDocumentForConfirmationOfVehicleRegistration”.

### **6.5.37.4 Exceptions**

See external interface “IF\_SOE\_GetDocumentForConfirmationOfVehicleRegistration”.

---

## **6.5.38 AF\_GenerateDocumentsPreviewArchive**

This AF generates an archive containing generated PDF previews of the latest versions of all productive document templates.

### **6.5.38.1 Sequence Description**

#### ***Step 1: Get locales for document generation***

- Retrieve locales for all supported MBconnect countries by calling AF\_GetSupportedMBconnectCountryLocales
- For each retrieved locale get all fallbacks and add it to a set of locales used for document preview generation

#### ***Step 2: Generate Document Previews***

- 
- Retrieve the latest versions of all documents. Generate a document preview AF\_GenerateDocumentsPreview for each document and each locale in Step 1.

### **Step 3: Archive Generated PDF previews**

- Archive the generated PDF files in a ZIP.

#### **6.5.38.2 Input**

None.

#### **6.5.38.3 Output**

None.

#### **6.5.38.4 Exceptions**

None.

### **6.5.39 AF.GetUserAgreementMasterData**

This AF provides access to the user agreement related master data. If there is no list of user agreements given as input parameter, the answer covers all user agreements existing in SOE.

#### **6.5.39.1 Sequence Description**

If the list of user agreements is empty retrieve all enabled <UserAgreement>.

If a list of user agreements is given, only consider those instances of <Document> where the pair <><Document>.<documentID>, <Document>.<versionID>>> match the given data.

For all relevant instances of <Document>, return <><<Document>.<documentID>, <Document>.<versionID>>>.

For each user agreement additionally determine:

all available translations, and attach the list of locales and localized user agreement names to the corresponding user agreement.

If there is no user agreement found for a given user agreement ID (and version), an empty list is returned for this user agreement.

#### **6.5.39.2 Input**

See internal interface “IF\_SOE.GetUserAgreementMasterData”.

#### **6.5.39.3 Output**

See internal interface “IF\_SOE.GetUserAgreementMasterData”.

#### **6.5.39.4 Exception**

See internal interface “IF\_SOE.GetUserAgreementMasterData”.

---

## **6.5.40 AF\_GetTermsOfUseMasterDataForCountry**

This AF provides access to the available user agreement related master data of a given country.

### **6.5.40.1 Sequence Description**

#### **Step 1: Get the User Agreement(s)**

For a given country, retrieve the latest enabled version of each <UserAgreement> for which the <UserAgreement.validFrom> date is equal or smaller than today.

If there is no user agreement found for a given country, abort here and return an empty list.

#### **Step 2: Get the User Agreement description**

If a locale is given, only determine the localized user agreement name corresponding to the user agreement in the given locale. Return the given locale as the locale of the localized user agreement name even if a fallback had to be applied.

If the locale is not given, determine all locales and localized user agreement names corresponding to the user agreement.

Note: The locales for a document are saved in the document blocks, which are referenced in the document definition. If a document block consists of multiple sub document blocks, the locales for each sub document block have to be part of the locales for the document.

#### **Step3: Get the languages**

If a locale is given, return the language from this locale as the list of available languages.

If the locale is not given, return the distinct languages from the locales for the user agreements from step 1 as the list of available languages.

### **6.5.40.2 Input**

See internal interface “IF\_SOE\_GetTermsOfUseMasterDataForCounty”.

### **6.5.40.3 Output**

See internal interface “IF\_SOE\_GetTermsOfUseMasterDataForCounty”.

### **6.5.40.4 Exceptions**

See internal interface “IF\_SOE\_GetTermsOfUseMasterDataForCounty”.

## **6.6 Batches**

Batchname	Called Application Function	Description
Batch “Generate Documents”	AF_GenerateDocumentsPreviewArchive	Execution time: night time Execution frequency: daily

Figure 66: Batches

## **6.7 Error Messages**

Message Id	Fault Title	Fault Message
GENDOC_001	Cannot find	Cannot find the localized Template <VersionedDocumentTemplate>.

---

	template	Please define the template in version <version> for the locale: <locale>.
GENDOC_002	Cannot find font style sheet	The font style sheet could not be found.
GENDOC_003	Cannot find font style	The font style <font_style> could not be found in the font style sheet.
GENDOC_004	Cannot find dynamic content	The dynamic content for the field <field> could not be found.
GENDOC_005	Cannot expand XML tag	The XML tag <tag> could not be expanded.
GENDOC_006	Cannot find CustomTag	Cannot find the CustomTag <CustomTag> for Country <Country>.
GENDOC_007	Document Trigger not defined	The document trigger <DocumentTriggerEnum> is not defined.
GENDOC_008	No document was found.	No Document has been found for documentTriggerEnum <DocumentTriggerEnum> and OutputTypeEnum <OutputTypeEnum>.
GENDOC_009	No document trigger assignment was found.	No DocTriggerTypeOutputTypeDocAssignment has been found for documentTriggerEnum <DocumentTriggerEnum> and OutputTypeEnum <OutputTypeEnum>.
DOCMAS_002	No service selected	At least one service must be selected.
DOCMAS_003	Email subject empty	The email subject cannot be empty.
DOCMAS_004	Localized content already exists	The localized content for <country> and <language> already exists.
DOCMAS_005	Localized content already exists	The localized content for <brand>, <country> and <language> in version <version> already exists.
DOCMAS_006	PDF already approved	The PDF in version <version> for the DocType <DocType> is already approved. Altering PDF templates is prohibited at this point.
DOCMAS_007	Document already approved	The document in version <version> for the DocType <DocType> is already approved. Altering documents is prohibited at this point.
DOCMAS_008	Field must not be empty	The field <field> must not be empty.
DOCMAS_009	File must be uploaded	A PDF file must be present to save a Document Template version.
DOCMAS_012	Cannot withdraw user agreement from countries	The edited version {0} of this document must contain at least all countries which are contained in the previous version {1} of this document. The countries with the country codes {2} are missing in version {0}.
DOCMAS_013	Modifying country availability is not allowed	You removed one or more countries from an already enabled document. This is not allowed. Please re-select the countries with the country codes {0}.
DOCMAS_022	Duplicate image detected	The image ("<1>") you are trying to save already exists.
ACCDAS_002	User ID does not exist	The given user ID does not exist.
GETDOC_003	User Agreement ID does not exist	The given user agreement ID does not exist.
VEHPRO_005	FIN does not exist	The given FIN does not exist.
GETDOC_005	Vehicle Configuration does not exist	The given vehicle configuration does not exist.
GETDOC_006	Both intervals specified	Only one interval date can be specified as input.
GENDOC_010	Cannot find	Cannot find the localized DocumentBlock

---

	Document Block	<VersionedDocumentBlock>. Please define the document block in version <version> for the locale <locale>.
GENDOC_011	Images are not supported	No support for embedding images inside e-mail text.
GENDOC_012	No image was found.	Cannot find the image <image>.
DOCSESSION_001	Document element locked by another user	The element you are trying to edit is already being edited by user <user>. Your changes will be discarded automatically.
DOCSESSION_002	Document element is currently being created	The element can't be loaded because it is currently being created.

# 7 Component „Contract Management“

## 7.1 Dialogs

None.

## 7.2 External View - Consumed Interfaces

### 7.2.1 IF\_CPD\_UpdateMbcServiceAvailability

Communication type: Synchronously

This interface informs the CPD system about the current status of a certain customer from a sales point of view. This includes the following information:

- which services are potentially possible for the customer's vehicles and available in the customers address country
- if the matching user agreement for a service was already accepted by the customer
- which vehicle a service is attached to (if applicable)
- list of all licenses for the given users including the license duration

Note: On a technical layer SOE must tell the TSB whether this call/message was sent in a BATCH context or a regular call.

#### 7.2.1.1 Input

Parameter Name	Man d./O pt.	Format	Length	Data Model	Possible Values	Annotation
List of services availabilities (0...*)						
Serviceld	Man d.	Integer	10	Service.Serviceld	Examples: 1, 2, 3, ...	The ID of the service.
userId	Man d.	String	50	Input parameter from calling AF.		The user ID of the customer account whose services are being transferred.
UserAgreementStatus	Man d.	String		UserAgreementConsent. userAgreementAcceptanceStatusEnum	{"ACCEPTED", "REQUIRED", "INAPPLICABLE"}	INAPPLICABLE: no user agreement needed for service  REQUIRED: user agreement needed, but not yet accepted  ACCEPTED: user agreement has been accepted
FIN	Man d.	String	17	Contract.Vehicle.FIN	Example: WDD1690 341J76450 7	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
LicenseID	Opt.	String	50	License.LicenseID		License ID
List of all licenses for given users						
LicenseID	Man d.	String	50	License.LicenseID		Die Liste der Lizenzen, die obere Lizenzen IDs refenzieren
LicenseS	Man	LicenseOrd		LicenseOrder.Start	LicenseSta	

Parameter Name	Man d./O pt.	Format	Length	Data Model	Possible Values	Annotation
tartType	d.	erStartEnum		Type	rtEnum [PREDEFINED_START_DATE, INITIAL_SERVICE_ACTIVATION]	
PredefinedStartDate	Opt.	Date		LicenseOrder.PrefefinedStartDate	Mand. for LicenseStartType PREDEFINED_START_DATE	
<b>List of license durations</b>						
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationMonths	Man d	Int	-	LicenseOrder.DurationMonth		
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationUnlimited	Man d	Boolean	-	LicenseOrder.DurationUnlimited		

Table 236: IF\_CPD\_updateMbcServiceAvailability Input

### 7.2.1.2 Output

None.

### 7.2.1.3 Exceptions

None.

## 7.3 External View - Offered Interfaces

### 7.3.1 IF\_SOE\_CreateLicenses

**Communication type:** Synchronously

This interface is called to create licenses for the given service masters which are ordered by a given user. The licenses need be created, to allow the activation and usage of services on vehicles. Since the licensed service may generally be used on only one vehicle (except e.g. concierge services) the information on which vehicle the license will be used is to provide. Additionally the information about validity period of licenses and their start type is captured. Also using the given transaction ids the license order entries are recorded.

After the license related changes are performed, the applied data needs to be sent to CPD. Finally the list of created licenses, affected fin and service master ids are to return.

Internally calls AF\_CreateLicenses.

### 7.3.1.1 Input

Parameter Name	Man d./O pt.	For-mat	Le ngt h	Data Model	Possible Values	Annotation
UserID	Man d.	String	50	UserProxy.userID		The user ID of the customer.
TransactionID	Man d.	String	60	LicenseOrder.TransactionID		UniqueReference to ensure idempotence
List of license durations						
o FIN	Man d.	String	17	VehicleProxy.FIN	Example: WDD1690 341J76450 7	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
o serviceMasterId	Man d.	Inte-ger	-	ServiceMas-ter.ServiceMasterID	-	The service master id.
o LicenseD urationUni t	Man d.	ENUM			[DAY,MON TH,YEAR]	
o LicenseD urationVal ue	Man d.	Int	-			
o LicenseSt artType	Man d.	Licens eOrde rStart Enum		LicenseOrder.StartT ype	LicenseSt artEnum [PREDE- FINED_ST ART_DAT E,INI- TIAL_SER VICE_ACT IVATION]	
o Predefine dStartDate	Opt.	Date		LicenseOrder.Predef inedStartDate	Mand. for LicenseSt artType PREDE- FINED_ST ART_DAT E	

### 7.3.1.2 Output

Parameter Name	Man d./O pt.	For-mat	Le ngt h	Data Model	Possible Values	Annotation
<b>List [1..20]</b>						
LicenseID	Man d.	String	50	License.LicenseID		
FIN	Opt.	String	17	VehicleProxy.FIN	Example: WDD1690 341J76450 7	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
serviceMaste rId	Man d.	Inte-ger	-	ServiceMas-ter.ServiceMasterID	-	The service master id.

### 7.3.1.3 Exceptions

Message Id	Fault Message	Error reason
------------	---------------	--------------

ACCDAS_002	User does not exist	The requested customer could not be found in the customer directory.
ACCDAS_004		An unexpected error occurred while connecting to the system CPD
CONMAN_009	Vehicle assignment does not exist	The requested combination of user ID and FIN does not exist.
LICENSE_003	Invalid license duration	The license duration must be greater than zero.
SERMAN_023	Service master does not exist	The requested service master ID does not exist.
VEHPRO_005	Vehicle does not exist	The given vehicle could not be found on the Daimler vehicle database.

### 7.3.2 IF\_SOE\_InformOfAccountDeletion

**Communication type:** Asynchronously

This interface is called by the system CPD upon the deletion of a user account and all information in SOE regarding the user of the given user ID. These are:

- any existing availabilities referencing the user
- any existing references to licenses
- any existing acceptances regarding user agreements

If the user is unknown in SOE, no error will be raised, as this would not generate any beneficiary information to the calling system.

Internally the AF\_DeleteCustomerSpecificData is called to delete all data related to the given customer.

#### 7.3.2.1 Input

Parameter Name	Mand./ Opt.	For- mat	Length	Data Model	Possible Values	Annotation
<b>Payload</b>						
userId	Mand.	String		Used as input parameter for AF_DeleteCustomerSpecificData		The user ID of the deleted customer account.
<b>Metadata</b>						
SystemId	Mand.	String	20	Used as input parameter for AF_DeleteCustomerSpecificData	Examples: „My Mercedes“, „MBC POS“, „CPD“	The name of the system the action was originally triggered in (not necessarily the one which sent the message).
RetailerId	Opt.	String	20	Used as input parameter for AF_DeleteCustomerSpecificData	Example: „TR0364178“	ID of the retailer who triggered the action (if applicable).
UserId	Opt.	String	50	Used as input parameter for AF_DeleteCustomerSpecificData	Examples: „D0X00107“, „mmeier“, „vorna-me.nachname“	User ID of the user who originally triggered the action (if applicable).

Table 237: IF\_SOE\_InformOfAccountDeletion Input

#### 7.3.2.2 Output

None.

#### 7.3.2.3 Exceptions

None.

### 7.3.3 IF\_SOE\_InformOfCustomerDataChange

**Communication type:** Synchronously

This interface informs SOE when the profile data of a customer has changed. It retrieves the two instances of the customer profile data: before and after the update operation. SOE determines the set of changed profile fields. The change is relevant only when the <ADDRESS COUNTRY> has changed: In case a user moves to a different country, all user agreements not supported in that country will be canceled.

The interface returns following information:

- which services are potentially possible for the customer's vehicles and available in the customers address country
- if the matching user agreement for a service was already accepted by the customer
- list of services available for user and his vehicle(s) (if applicable)
- list of all licenses for the given users including the license duration

Internally, AF\_InformOfCustomerDataChange (→ section 7.5.5) is called.

Note: The incoming customer data fields and the mapping of this interface are described centrally in AF\_InformOfCustomerDataChange (→ see chapter 7.5.5).

#### 7.3.3.1 Input

Parameter Name	Mand./O pt.	Format	Length	Data Model	Possible Va-lues	Annotation
<b>Customer Profile (See Handling of Profile Data within SOE chapter 2.3.18)</b>						
OldData	Mand.	UserTyp e		UserType		The old user data. The whole profile is requested.
NewUserData	Mand.	UserTyp e		UserType		The new user data. The whole profile is requested
<b>Metadata</b>						
ChangeAction	Opt.	Custome rDataCha ngeActio nEnum		CustomerDataCh angeActionEnum	“Add”, “UP-DATE” or “RE-MOVE”	One of the following Action of the data change: <ul style="list-style-type: none"><li>• ADD</li><li>• UPDATE</li><li>• Remove</li></ul> Hint: the <change action> is ignored by SOE
System	Mand.	String	20	originatingSystem	Examples: “My Mercedes”, “MBC POS”, “CPD”	The name of the system the action was originally triggered in (not necessarily the one which sent the message).
RetailerId	Opt.	String	20	organisationID	Example: “TR0364178”	ID of the retailer who triggered the action (if applicable).
UserId	Opt.	String	50	userID	Examples: „D0X00107“, „mmeier“, „vorna-me.nachname“	User ID of the user who originally triggered the action (if applicable).

Table 238: IF\_SOE\_InformOfCustomerDataChange Input

### 7.3.3.2 Output

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of services availabilities (0...*)						
ServiceId	Man d.	Integer	10	Service.ServiceId	Examples: 1, 2, 3, ...	The ID of the service.
userId	Man d.	String	50	Input parameter from calling AF.		The user ID of the customer account whose services are being transferred.
UserAgreementStatus	Man d.	String		UserAgreementConsent. userAgreementAcceptanceStatusEnum	{"ACCEPTED", "REQUIRED", "INAPPLICABLE"}	INAPPLICABLE: no user agreement needed for service  REQUIRED: user agreement needed, but not yet accepted  ACCEPTED: user agreement has been accepted
FIN	Man d.	String	17	Contract.Vehicle.FIN	Example: WDD1690341J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
LicenseID	Opt.	String	50	License.LicenseID		License ID
List of all licenses for given users						
LicenseID	Man d.	String	50	License.LicenseID		Die Liste der Lizenzen, die obere Lizenzen IDs refenzieren
LicenseStartType	Man d.	LicenseOrderStartEnum		LicenseOrder.StartType	LicenseStartEnum [PREDEFINED_START_DATE, INITIAL_SERVICE_ACTIVATION]	
PredefinedStartDate	Opt.	Date		LicenseOrder.PrefixedStartDate	Mand. for LicenseStartType PREDEFINED_START_DATE	
List of license durations						
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationMonths	Man d	Int	-	LicenseOrder.DurationMonth		
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationUnlimited	Man d	Boolean	-	LicenseOrder.DurationUnlimited		

---

Table 239: IF\_SOE\_InformOfCustomerDataChange Output

### 7.3.3.3 Exceptions

Message Id	Fault Message	Error reason
ACCDAS_002	User does not exist	if the given user could not be resolved in the CPD.
ACCDAS_004		if any unexpected, technical errors occur when connecting to CPD.
GENDOC_006	Cannot find the CustomTag {0} for Country {1}.	Only exactly one "Date"-input parameter is allowed. If more than one is passed as input, the error GENDOC_006 occurs
GENDOC_007	No Document has been found for documentTriggerEnum {0} and OutputTypeEnum {1}.	If the given <DocumentTriggerEnum> is unknown
VEHPRO_005	The given vehicle could not be found on the Daimler vehicle database.	if the given vehicle cannot be found on ODC
VEHPRO_006	An unexpected error occurred when trying to retrieve the requested vehicle from the ODC system.	if things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors)

Table 240: IF\_SOE\_InformOfCustomerDataChange exceptions

### 7.3.4 IF\_SOE\_InformOfInitialServiceActivation

**Communication type:** Asynchronously

This interface is not active anymore. If it will be called, then an error message CON-MAN\_010 will be thrown.

#### 7.3.4.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
userID	Mand.	String	-	User.userID		The user ID of the customer.
FIN	Mand.	String	17	Vehicle.fin	Example: WDD169007 1J236589	The vehicle identification number is vehicle specific. This information is printed onto the document.
serviceID	Mand.	Int	10	Service.serviceID	Examples: 1, 12356, 1234567890	The service ID identifies the specific service.
activationDate	Mand.	DateTime	-	Contract.startDate	Example: "01.01.2014 00:00"	Date and time that specifies from which point on the service can be used.
additionalActivationReference	Mand.	String	50	Contract.additionalActivationReference		<p>Describes the additional activation reference for the given service.</p> <p>For a service with contract start trigger "initial service activation", the start date for the given service will be set to the given activation date each time a new additional activation reference for a specific user and FIN is given.</p> <p>E.g.: For the service</p>

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						LiveTraffic SD , the additional activation reference will be the ID of the sd card.

Table 241: IF\_SOE\_InformOfInitialServiceActivation Input

### 7.3.4.2 Output

None.

### 7.3.4.3 Exceptions

None.

## 7.3.5 IF\_SOE\_InformOfVehicleRegistration

**Communication type:** Synchronously

This interface is called by the CPD upon the assignment of a customer or business partner to a vehicle. This usually happens when a vehicle is (re-)sold. SOE will connect the given customer or business partner to the vehicle and determine the services that are potentially available for the given vehicle.

If the vehicle is registered to a regular customer, the licenses for the give-away services are created or transferred to the new user account.

The list of services that are potentially available for the given vehicle and, for each service, the status of the matching user agreement is returned. Additionally a list of licenses including the license durations is returned.

If the vehicle is currently connected to a different user account, the connection will be separated and no error will be raised, because CPD is the master system for the vehicle registration. However, a warning is logged.

Internally the AF\_InformOfVehicleRegistration is called to register the vehicle to the given customer.

### 7.3.5.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Customer Profile (See Handling of Profile Data within SOE chapter 2.3.18)</b>						
User	Opt.	-		UserType		<p>The whole profile is requested.</p> <p><u>XOR:</u> Either the User or the BusinessPartner has to be provided.</p>
Business-Partner	Opt.	String	20	BusinessPartner-Proxy.BusinessPartnerId	Example: "MBC.DFS"	<p>The ID of the business partner.</p> <p><u>XOR:</u> Either the User or the BusinessPartner has to be provided.</p>
<b>Payload</b>						
Fin	Mand.	String	17	Used as input parameter for AF_RegisterVehicleToUser	Example: WDD1690341 J764507	The FIN of the vehicle which shall be registered to a customer.
FirstRegistrat	Mand	Date	-	Used as input	Example:	The first registration date

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
ionDate	.			parameter for AF_RegisterVehicleToUser	01.01.2014	of the vehicle to register.
<b>Metadata</b>						
SystemId	Man d.	String	20	Used as input parameter for AF_RegisterVehicleToUser	Examples: "My Mercedes", "MBC POS", "CPD"	The name of the system the action was originally triggered in (not necessarily the one which sent the message).
RetailerId	Opt.	String	20	Used as input parameter for AF_RegisterVehicleToUser	Example: "TR0364178"	ID of the retailer who triggered the action (if applicable).
UserId	Opt.	String	50	Used as input parameter for AF_RegisterVehicleToUser	Examples: „D0X00107“, „mmeier“, „vorna-me.nachname“	User ID of the user who originally triggered the action (if applicable).

Table 242: IF\_SOE\_InformOfVehicleRegistration Input

### 7.3.5.2 Output

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of services availabilities (0...*)						
ServiceId	Man d.	Integer	10	Service.ServiceId	Examples: 1, 2, 3, ...	The ID of the service.
userId	Opt..	String	50	Input parameter from calling AF.		The user ID of the customer account whose services are being transferred.  <u>XOR</u> : Either the UserId or the BusinessPartnerId will be provided.
business PartnerId	Opt.	String	50			The user ID of the business partneraccount whose services are being transferred.  <u>XOR</u> : Either the UserId or the BusinessPartnerId will be provided.
UserAgreementStatus	Man d.	String		UserAgreementConsent. userAgreementAcceptanceStatusEnum	{"ACCEPTED", "REQUIRED", "INAPPLICABLE"}	INAPPLICABLE: no user agreement needed for service  REQUIRED: user agreement needed, but not yet accepted  ACCEPTED: user agreement has been accepted
FIN	Man d.	String	17	Contract.Vehicle.FIN	Example: WDD1690341J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
Licencel D	Opt.	String	50	License.LicenseID		License ID

Parameter Name	Man d./O pt.	Format	Length	Data Model	Possible Values	Annotation
List of all licenses for given users						
LicenseID	Man d.	String	50	License.LicenseID		Die Liste der Lizenzen, die obere Lizenzen IDs refenzieren
LicenseStartType	Man d.	LicenseOrderStartEnum		LicenseOrder.StartType	LicenseStartEnum [PREDEFINED_START_DATE, INITIAL_SERVICE_ACTIVATION]	
PredefinedStartDate	Opt.	Date		LicenseOrder.PrefefinedStartDate	Mand. for LicenseStartType PREDEFINED_START_DATE	
<b>List of license durations</b>						
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationMonths	Man d	Int	-	LicenseOrder.DurationMonth		
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationUnlimited	Man d	Boolean	-	LicenseOrder.DurationUnlimited		

Table 243: IF\_SOE\_InformOfVehicleRegistration Output

### 7.3.5.3 Exceptions

Message Id	Fault Message	Error reason
GENDOC_007	No Document has been found for documentTriggerEnum {0} and OutputTypeEnum {1}.	If the given <DocumentTriggerEnum> is unknown
VEHPRO_005	The given vehicle could not be found on the Daimler vehicle database.	if the given vehicle cannot be found on ODC
VEHPRO_006	An unexpected error occurred when trying to retrieve the requested vehicle from the ODC system.	if things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors)

### 7.3.6 IF\_SOE\_InformOfVehicleSeparation

**Communication type:** Synchronously

This interface is called by the system CPD upon the separation of a customer or business partner from a vehicle. This usually happens when a vehicle is re-sold as a used

vehicle and so gets a new owner. SOE will delete all its information regarding the connection between the vehicle and the user.

If the connection is unknown in SOE, no error will be raised, as this would not generate any beneficiary information to the calling system. However, a warning is logged. The output of this interface will always be empty.

Internally, the AF\_InformOfVehicleSeparation is called.

### 7.3.6.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Customer Profile (See Handling of Profile Data within SOE chapter 2.3.18)</b>						
User	Opt.	-		UserType		The whole profile is requested.  <u>XOR</u> : Either the UserId or the BusinessPartnerId has to be provided.
Business-Partner	Opt.	String	20	BusinessPartner-Prox.BusinessPartnerId	Example: "MBC.DFS"	The ID of the business partner  <u>XOR</u> : Either the UserId or the BusinessPartnerId has to be provided.
<b>Payload</b>						
Fin	Mand.	String	17	-	Example: WDD1690341 J764507	The FIN of the vehicle which shall be separated from a customer.
Role	Opt.	Enum		ExternalRolesEnum		The role of the user that originally triggered the action
SystemId	Man d.	String	20	-	Examples: "My Mercedes", "MBC POS", "CPD"	The name of the system the action was originally triggered in (not necessarily the one which sent the message).
RetailerId	Opt.	String	20	-	Example: "TR0364178"	ID of the retailer who triggered the action (if applicable).
UserId	Opt.	String	50	-	Examples: „D0X00107“, „mmeier“, „vor-name.nachname“	User ID of the user who originally triggered the action (if applicable).

Table 244: IF\_SOE\_InformOfVehicleSeparation Input

### 7.3.6.2 Output

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of services availabilities (0...*)						
Serviceld	Man d.	Integer	10	Service.Serviceld	Examples: 1, 2, 3, ...	The ID of the service.
UserId	Opt.	String	50	Input parameter from calling AF.		The user ID of the customer account whose services are being transferred.  <u>XOR</u> : Either the UserId or the BusinessPartnerId will be pro-

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
BusinessPartnerId	Opt.	String	20	BusinessPartnerProxy.BusinessPartnerId	Example: "MBC.DFS"	vided. The user ID of the business partner account whose services are being transferred.  <u>XOR:</u> Either the UserId or the BusinessPartnerId will be provided.
UserAgreementStatus	Man d.	String		UserAgreementConsent. userAgreementAcceptanceStatusEnum	{"ACCEPTED", "REQUIRED", "INAPPLICABLE"}	INAPPLICABLE: no user agreement needed for service  REQUIRED: user agreement needed, but not yet accepted  ACCEPTED: user agreement has been accepted
Fin	Man d.	String	17	VehicleProxy.Fin	Example: "WDD1690341J764507"	The European identifier of the vehicle.
LicenseID	Opt.	String	50	License.LicenseID		License ID
<b>List of all licenses for given users</b>						
LicenseID	Man d.	String	50	License.LicenseID		Die Liste der Lizenzen, die obere Lizenzen IDs refenzieren
LicenseStartType	Man d.	LicenseOrderStartEnum		LicenseOrder.StartType	LicenseStartEnum [PREDEFINED_START_DATE, INITIAL_SERVICE_ACTIVATION]	
PredefinedStartDate	Opt.	Date		LicenseOrder.PrefefinedStartDate	Mand. for LicenseStartType PREDEFINED_START_DATE	
<b>List of license durations</b>						
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationMonths	Man d	Int	-	LicenseOrder.DurationMonth		
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationUnlimited	Man d	Boolean	-	LicenseOrder.DurationUnlimited		

Table 245: IF\_SOE\_InformOfVehicleSeparation output

### 7.3.6.3 Exceptions

Message Id	Fault Message	Error reason
ACCDAS_002	User does not exist	if the given user could not be resolved in the CPD.
ACCDAS_004		if any unexpected, technical errors occur when connect-

		ing to CPD.
GENDOC_006	Cannot find the CustomTag {0} for Country {1}.	Only exactly one "Date"-input parameter is allowed. If more than one is passed as input, the error GETDOC_006 occurs
GENDOC_007	No Document has been found for documentTriggerEnum {0} and OutputTypeEnum {1}.	If the given <DocumentTriggerEnum> is unknown
VEHPRO_005	The given vehicle could not be found on the Daimler vehicle database.	if the given vehicle cannot be found on ODC
VEHPRO_006	An unexpected error occurred when trying to retrieve the requested vehicle from the ODC system.	if things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors)

Table 246: IF\_SOE\_InformOfVehicleSeparation exceptions

### 7.3.7 IF\_SOE\_SetUserAgreementStateOfConsent

**Communication type:** Synchronously

This interface sets the state of consent for a user concerning a certain user agreement. This happens either when a user agreement is signed or revoked by the user.

Note: If the address country of the customer is not supported by MBconnect, an exception is thrown. Further, if a user tries to sign a user agreement in an outdated version or the user agreement is not available in the country where the user lives in, the system will refuse to set this state of consent.

Internally, the AF\_SetUserAgreementStateOfConsent is called to set the state of consent.

#### 7.3.7.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
User-Id	Mand.	String		User.userID		The user ID of the customer. The ID is necessary to retrieve customer information (e.g name) that is printed onto the documents.
List of user agreements						
uaID	Mand.	String		Document.documentID	Examples: "Basic Services"	The user agreement ID identifies the specific user agreement for which the status is to be recorded.
uaAcceptanceStatus	Mand.	String	12	UserAgreementConsent.AcceptanceStatus (the model deviates slightly from what possible values are returned in this IF. Only relevant states are "accepted" and "not accepted". Other states are derived to the former two.)	"Accepted", "Not Accepted"	The acceptance status of the user agreement for the specific user.
uaVersion	Mand.	int	10	Document.VersionID	Examples: 1, 12356,	The user agreement version that the customer signed or

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
					1234567890	revoked.
uaAcceptanceLocale	Mand.	String	5	UserAgreementConsent. uaAcceptanceLocale	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the signed user agreement.
<b>Metadata</b>						
SystemId	Man d.	String	20	Used as input parameter for AF_SetUserAgreementStateOfConsent	Examples: "My Mercedes", "MBC POS", "CPD"	The name of the system the action was originally triggered in (not necessarily the one which sent the message).
Orgid (RetailerId)	Opt.	String	20	Used as input parameter for AF_SetUserAgreementStateOfConsent (Market.GSSNOutletOutletID)	Example: "TR0364178", "GS0008205"	ID of the retailer who triggered the action (if applicable). The organization ID of the retailer, which is recorded for safekeeping purposes to know where the consent was given.
UserId	Opt.	String	50	Used as input parameter for AF_SetUserAgreementStateOfConsent	Examples: „D0X00107“, „mmeier“, „vorname.nachname“	User ID of the user who originally triggered the action (if applicable). The ID of the retailer user, if the call came from POS. Else it's the user ID from the customer if the call came from MyMercedes.

Table 247: IF\_SOE\_SetUserAgreementStateOfConsent input

### 7.3.7.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
AllUserAgreementsConsentsSet	Mand.	Boolean		-	True, False	Returns true if the state of all consents was set successfully.
<b>List of user agreement which consents could not be set</b>						
ualD	Opt.	String		Document.documentID	Examples: 1, 12356, 1234567890	The user agreement ID for the user agreements which consents could not be set due to an outdated version.

Table 248: IF\_SOE\_SetUserAgreementStateOfConsent output

### 7.3.7.3 Exceptions

Message Id	Fault Message	Error reason
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
ACCDAS_004	An unexpected error occurred while connecting to the system CPD.	if any unexpected, technical errors occur when connecting to CPD.
ACCDAS_005	Action is not possible, because the customer is notified in an unsupported country.	Action is not possible, because the customer lives in an unsupported country.
GETDOC_003	The requested User Agreement '{0}' does not exist.	If the combination of document ID and version ID cannot be found, the error GETDOC_003 is returned
GENDOC_006	Cannot find the CustomTag {0} for Coun-	Only exactly one "Date"-input parameter is allowed. If more than one is passed as input, the error GENDOC_006 occurs

---

	try {1}.	
GENDOC_007	No Document has been found for documentTriggerEnum {0} and OutputTypeEnum {1}.	If the given <DocumentTriggerEnum> is unknown
VEHPRO_005	The given vehicle could not be found on the Daimler vehicle database.	if the given vehicle cannot be found on ODC
VEHPRO_006	An unexpected error occurred when trying to retrieve the requested vehicle from the ODC system.	if things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors)

Table 249: IF\_SOE\_SetUserAgreementStateOfConsent exceptions

## 7.4 Internal View - Offered Interfaces

None.

## 7.5 Implementation

### 7.5.1 AF\_InformOfVehicleRegistration

#### 7.5.1.1 General Description

See external interface **Error! Reference source not found..**

#### 7.5.1.2 Sequence Description

If the User is given as input parameter, call the application function **Error! Reference source not found..** Omit the BusinessPartnerId as output parameter. If the BusinessPartner is given as input parameter, call the application function AF\_RegisterVehicleToBusinessPartner. Omit the UserId as output parameter. Furthermore, return an empty list of licenses and, for each B2B service availability, set the UserAgreementStatus to "INAPPLICABLE" and omit the Licenseld.

#### 7.5.1.3 Input

See external interface **Error! Reference source not found..**

#### 7.5.1.4 Output

See external interface **Error! Reference source not found..**

#### 7.5.1.5 Exceptions

See external interface **Error! Reference source not found..**

### 7.5.2 AF\_CreateLicenses

See external interface IF\_SOE\_CreateLicenses.

#### 7.5.2.1 Sequence Description

Step 1: If the license duration of the input (LicenseDurationValue) is smaller or equal to "0", the error message LICENSE\_003 is thrown.

---

Step 2: Determine the UserProxy by calling IIF\_GetOrCreateOrUpdateUserProxy with the given userID.

Step 3: Determine the VehicleProxy by calling IIF\_GetVehicleProxiesForUser with the given userID. Check if the given fin / list of fins is/are contained in the retrieved VehicleProxies. If the fin is not contained in the retrieved list of VehicleProxies then the error message CONMAN\_009 is thrown. For each matched VehicleProxy continue with next step.

Step 4: Create temporal integer variables LicenseDurationYears, LicenseDurationMonths, LicenseDurationDays and Boolean variable LicenseDurationUnlimited which has value “FALSE” by default.

Step 5: From the given input parameters check the value of LicenseDurationUnit:

- If it equal to “DAY”, then set temporal attributes LicenseDurationYears to 0, LicenseDurationMonths to 0 and LicenseDurationDays to parameter LicenseDurationValue from.
- If it equal to “MONTH”, then set temporal attributes LicenseDurationYears to 0, LicenseDurationMonths to parameter LicenseDurationValue from input and LicenseDurationDays to 0.
- Else if it equal to “YEAR”, then set temporal attributes LicenseDurationYears to parameter LicenseDurationValue from input, LicenseDurationMonths to 0, LicenseDurationDays to 0.

Step 6: Call the IIF\_CreateLicensesAndLicenseOrders and provide following parameters:

- UserProxy (from step 2),
- TransactionID (given in input),
- orderSource.ORDERED
- list of matched VehicleProxy (from step 3),
- serviceMasterId (given in input),
- LicenseDurationYears (from step 5),
- LicenseDurationMonths (from step 5),
- LicenseDurationDays (from step 5),
- LicenseDurationUnlimited (from step 5),
- LicenseStartType (given in input),
- PredefinedStartDate (given in input),
- actual date for licenseOrderDate

Remember the returned result.

Step 7: For each matched VehicleProxy call AF\_DetermineSummaryForUser with the given userId and fin and pass in the returned result to AF\_UpdateMbcServiceAvailability to inform CPD about affected changes.

Step 8: Provide the results returned from the IIF\_CreateLicensesAndLicenseOrders (step 6) to output.

### 7.5.2.2 Input

See external interface IF\_SOE\_CreateLicenses.

---

### **7.5.2.3 Output**

See external interface IF\_SOE\_CreateLicenses.

### **7.5.2.4 Exceptions**

See external interface IF\_SOE\_CreateLicenses.

## **7.5.3 AF\_DeleteCustomerSpecificData**

See external interface IF\_SOE\_InformOfAccountDeletion (→ section 7.3.2).

### **7.5.3.1 Sequence Description**

Step 1: Delete all the availabilities referencing the user

Call **IIF\_DeleteAvailabilitiesForUserAndVehicle** for given user.

Step 2: Remove the user from the licenses

In the case of deleting an user account, all the references to the licenses has to be removed. Therefore call **IIF\_SeparateLicensesFromUser** with user.

Step 3: Remove the references from license orders (LicenseOrder.OrderingUser)

Call **IIF\_SeparateLicenseOrdersFromUser** with the given userId to delete the reference of user in license orders.

Step 4: Remove user agreement consents

Call **IIF\_GetUserAgreementConsents** to get all the users agreements. Make sure that this result is stored on the database for historization before deleting the user agreement consent. Now delete them by calling **IIF\_DeleteUserAgreementConsents**.

Step 5: Delete UserProxy

Call **IIF\_DeleteUserProxy** and pass in the given userID to delete its UserProxy.

### **7.5.3.2 Input**

See external interface IF\_SOE\_InformOfAccountDeletion (→ section 7.3.2).

### **7.5.3.3 Output**

See external interface IF\_SOE\_InformOfAccountDeletion (→ section 7.3.2).

### **7.5.3.4 Exceptions**

See external interface IF\_SOE\_InformOfAccountDeletion (→ section 7.3.2).

## **7.5.4 AF\_DetermineSummaryForUser**

This application function returns the services, current acceptance status of the matching user agreement and licenses that are related for the given user and the user's vehicles if it is given.

### **7.5.4.1 Sequence Description**

**Step 1: Determine all service availabilities for user**

---

Determine all service availabilities that are associated with the given user by calling IIF\_GetAvailabilitiesForUser. Pass in the given userId. If the fin is given, then select from the returned output (e.g. this list of availabilities contains following datas: *Availability.userId*, *Availability.fin* and *Availability.serviceId*) availabilities related to the given vehicle.

Remember all referenced availabilities.

#### **Step 2: Retrieve service relevant data.**

Call IIF.GetServiceMasterData and pass in the list of service ids from step 1. The returned result will be used to extend the list of availabilities from step 1 with additional information: Availability.userId, Availability.fin, Availability.serviceId, Service.licenseRequired and Service.serviceMasterId.

#### **Step 3: Determine current status of user agreement**

For each available service from step 1 determine the current acceptance status of the matching user agreement. This is achieved by calling IIF\_GetStateOfUserAgreementForServices with these services. Merge the result with the list of available services remembered in step 2 (e.g. this list of availabilities extends as follows: Availability.userId, Availability.fin, Availability.serviceId, Service.licenseRequired, Service.serviceMasterId. and *UserAgreementStatus*).

#### **Step 4: Determine all licenses for user**

Determine all licenses and license related data such as license orders that are associated with the given user by calling IIF\_GetLicensesForUser with the given user. If the fin is given, then select from the returned output licenses related to the given vehicle. Remember all referenced licenses and license orders.

#### **Step 5: Join together licenses and availabilities**

For each available service from step 1 determine the related license (if exists) from the list of licenses retrieved in step 2. Use service master reference contained in both entities as a linkage between them (Service.ServiceMasterID = License.ServiceMasterID).

#### **Step 6: Return the data builded in step 5.**

##### **7.5.4.2 Input**

Name	Type / Length / BOM	Description
UserId	String	User ID of the customer whose contracts shall be read.
FIN	String	Optional parameter, describes the european vehicle identification number.

Table 250: AF\_DetermineSummaryForUser Input

##### **7.5.4.3 Output**

Parameter Name	Man d./O pt.	Format	Len gth	Data Model	Possible Values	Annotation
List of services (0...*)						
ServiceId	Man d.	Integer	10	Service.ServiceId	Examples: 1, 2, 3, ...	The ID of the service.
userId	Man d.	String	50	Input parameter from calling AF.		The user ID of the customer account whose services are being transferred.

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
UserAgreementStatus	Man d.	String		UserAgreementConsent. userAgreementAcceptanceStatusEnum	{"ACCEPTED", "REQUIRED", "INAPPLICABLE"}	INAPPLICABLE: no user agreement needed for service REQUIRED: user agreement needed, but not yet accepted ACCEPTED: user agreement has been accepted
FIN	Man d.	String	17	Contract.Vehicle.FIN	Example: WDD1690341J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
LicenseID	Opt.	String	50	License.LicenseID		License ID
<b>List of all licenses for given users</b>						
LicenseID	Man d.	String	50	License.LicenseID		Die Liste der Lizenzen, die obere Lizenzen IDs refenzieren
LicenseStartType	Man d.	LicenseOrderStartEnum		LicenseOrder.StartType	LicenseStartEnum [PREDEFINED_START_DATE, INITIAL_SERVICE_ACTIVATION]	
PredefinedStartDate	Opt.	Date		LicenseOrder.PrefefinedStartDate	Mand. for LicenseStartType PREDEFINED_START_DATE	
<b>List of license durations</b>						
LicenseDurationYears	Man d	Int	-	LicenseOrder.DurationYears		
LicenseDurationMonths	Man d	Int	-	LicenseOrder.DurationMonth		
LicenseDurationDays	Man d	Int	-	LicenseOrder.DurationDays		
LicenseDurationUnlimited	Man d	Boolean	-	LicenseOrder.DurationUnlimited		

Table 251: AF\_DetermineSummaryForUser Output

#### 7.5.4.4 Exceptions

None.

## **7.5.5 AF\_InformOfCustomerDataChange**

This AF will be called whenever the customer data is changed. It returns the user status regarding services and user agreements.

In case, the customers address country changes, the user's SOE status is updated, including availabilities, licenses and user agreement consents. Those user agreements that are not supported in the new country will be canceled. If the new address country is not supported by MBconnect all user agreements will be cancelled.

(Hint: If the user moved to a non-Mbc Country he will explicitly not be informed about the loss of his user agreement consents.)

### **7.5.5.1 Sequence Description**

Identify the list of changed user profile attributes. Remember them in the <ChangedFields> list.

#### ***Step 1: Check MBconnect Support of Address Country***

- If <ChangedFields> does not contain the field <ADDRESS\_COUNTRY> (no update of contracts necessary), call AF\_DetermineSummaryForUser and return the output result of the application function. Proceeding stops here.
- If <ChangedFields> contains the field <ADDRESS\_COUNTRY>, call IIF\_IsMBconnectCountry with <User>.<Address\_Country> as Country and remember the result. Additionally, call the internal interface IIF\_GetOrCreateOrUpdateUserProxy with the userId and the User.addressCountry as input parameters. Remember retrieved UserProxy.

#### ***Step 2: Gather all user agreements that are no longer valid for the user in the new country***

Determine the user agreements signed by the user in his old country by calling IIF.GetUserAgreementConsents . Pass in userId and UserAgreementAcceptanceStatus.AcceptanceState. Remember the output in set A. Get the list of all user agreements available for the user in his new address country by calling IIF GetUserAgreementsByDate (remember them in set B). Determine the list of user agreements that are not available in the new address country by subtracting the two sets: A \ B.

#### ***Step 3 Revoke all accepted User Agreements that are no longer valid in the new country***

For all user agreements determined in step 2 delete all instances by calling IIF\_DeleteUserAgreementConsents with the list of user agreements and the user.

#### ***Step 4: Inform Customer about automatically revoked user agreements***

If the user moved to a non-Mbc Country he will explicitly not be informed about the loss of his user agreement consents.

For the affected list of <UserAgreement> from step 2, build a map consisting from each user agreement and the locale in which the user agreement has been signed.

Call the internal interface IIF\_InformCustomer to inform the customer about his revoked user agreements, with the parameters:

- 
- <PerformedActionEnum> = USER AGREEMENT REVOKED
  - Map<UserAgreement, locale>
  - User
  - Omit vehicle configuration

#### **Step 5: Get Customer Vehicles**

Call the internal interface IIF\_GetVehicleProxiesForUser with the User.userId as input parameter to retrieve the vehicles that are assigned to the user. Remember the retrieved VehicleProxies. For each FIN, call IIF\_FetchVehicleDataWithoutLocale in order to determine the vehicle's configuration (including the model series and the first registration date). If that internal interface aborts with an error, abort with VEHPRO\_006.

#### **Step 6: Updating availabilities and licenses for user**

With the given UserProxy (from step 1) and the list containing vehicles (including their configurations, their model series and first registration dates) and VehicleProxies retrieved in previous step, call AF\_UpdateAvailabilititesAndLicensesForUser in order to update the user's availabilitites and licenses.

If that application function aborts with an error, abort with error message VEHPRO\_006.

#### **Step 7: Determine availabilities and licenses for user**

Retrieve the current user status regarding services and user agreements by calling AF\_DetermineSummaryForUser with the userID of the given user and return the retrieved data.

#### **7.5.5.2 Input**

See external interface "IF\_SOE\_InformOfCustomerDataChange".

#### **7.5.5.3 Output**

See external interface "IF\_SOE\_InformOfCustomerDataChange".

#### **7.5.5.4 Exceptions**

See external interface "IF\_SOE\_InformOfCustomerDataChange".

### **7.5.6 AF\_InformOfInitialServiceActivation**

See external interface.

#### **7.5.6.1 Sequence Description**

##### **Step 1: Retrieve Customer Data**

Call the internal interface IIF\_GetUserProfile with the given user ID to retrieve the customer data <User>.

##### **Step 2: Check contract activation trigger of service**

For a service with contractStartTrigger not equal „INITIAL\_SERVICE\_ACTIVATION“, exit the application function without any warning. For a service with contractStartTrigger „INITIAL\_SERVICE\_ACTIVATION“, continue with the next step.

---

**Step 3: Check if corresponding contract entity already exists**

Check if there is a contract entity associated to the given FIN, userID and serviceID. If such a contract entity does not exist, the vehicle has probably not been registered to the SOE or the service is not available for the address country of the user. Log an error message with the given parameters ("The service was activated without being available for this vehicle / in this country. The service duration could not be set.") and abort this function here.

**Step 4: Check changes of additional activation reference**

Compare the attribute additionalActivationReference from the contract entity with the given one in this function. If they are the same, return and log a warning ("A service was activated with the same additional activation reference for the same user and vehicle."). In case they are different or the attribute of the contract entity was not set yet, continue with the next step.

**Step 5: Change start date of contract and calculate duration**

Set the attribute startDate of the contract entity to the given parameter activationDate and additionalActivationReference of the contract entity to the identical parameter given in this function.

If any duration for the service (Service.duration) is set, calculate the attribute endDate of the contract entity with the following formula: Contract.endDate = Contract.startDate + Service.duration. In case no duration is set, set Contract.endDate to nothing.

**Step 6: Determine current contracts and inform CPD**

Determine the currently possible services for the customer by calling AF\_DetermineContractsForUser. Pass in the given user profile returned in step 1 as parameter. With the output data call AF\_UpdateMbcServiceAvailability in order to inform CPD about the current status of available services for the customer and his vehicles and the status of the related user agreements.

**7.5.6.2 Input**

See external interface IF\_SOE\_InformOfInitialServiceActivation.

**7.5.6.3 Output**

See external interface IF\_SOE\_InformOfInitialServiceActivation.

**7.5.6.4 Exceptions**

See external interface IF\_SOE\_InformOfInitialServiceActivation.

**7.5.7 AF\_InformOfMasterDataChangesAffectedByServiceEnablement**

This application function identifies each master data change affected by enabling or disabling the given services and informs adjacent systems.

**7.5.7.1 Sequence Description**

1. Call AF\_GetRequiredUserAgreementForService for each given service.

---

Aggregate the returned user agreements to a list of unique < Document.documentIDs >.

If the list is not empty, call IIF\_InformOfMasterDataChange with the following parameter:

- < MasterDataTypeEnum > = DOCUMENTS
- <dataType> = USERAGREEMENTSERVICEASSIGNMENT
- <subSet> = List of distinct < Document.documentIDs >

## 2. Call IIF.GetServiceAssignmentRulesForService for each given service.

For each returned service assignment rule call

IIF\_RetrieveCountryCodeForVehicleProductMasterData.

Aggregate the returned lists of countryCodes to a list of unique <countryCode>.

If the list is not empty, call IIF\_InformOfMasterDataChange with the following parameter:

- <MasterDataResourceTypeEnum> = SERVICE
- <dataType> = VEHICLECOUNTRYMASTERDATA
- <subSet> = List of distinct <countryCode>

### 7.5.7.2 Input

BO-Name	Type / Length / BOM	Description
ServiceList	List of <Service>	The list of services

Table 252: AF\_InformOfMasterDataChangesAffectedByServiceEnablement Input

### 7.5.7.3 Output

None.

### 7.5.7.4 Exceptions

None.

## 7.5.8 AF\_MigrateUserToNewUserAgreementBatch

This AF migrates all customers that currently have user agreement consents in effect, for which the user agreement is now valid in a newer version and for which the print approval was accepted. If the user agreement is now valid in a newer version and the customer was not informed, the AF logs a warning in the batch log file.

Should the batch abort during the run, all successfully migrated users are marked as such. The batch will continue from where it stopped after a restart.

### 7.5.8.1 Sequence Description

#### **Step 1: Get latest user agreements**

Call IIF.GetUserAgreementsByDate to retrieve all enabled <UserAgreement> with “the smallest value of Date” as input. From the set of retrieved user agreements, discard all versions but the latest productive form for each user agreement.

#### **Step 2: Determine affected user agreement consents**

---

Call IIF\_GetAllUserAgreementConsentsAndAssociations with input parameter AcceptanceState = ACCEPTED to get all user agreement consents corresponding to an accepted user agreement, and its relations to the user proxy, the user agreement, and if available the user agreement consent batch meta data. From these user agreement consents select those which satisfy each of the conditions below:

- An entity object of type UserAgreementConsentBatchMetaData is attached to it.
- The user agreement consent is associated to none of the user agreements determined in step 1.

Group the filtered user agreement consents by the associated user proxy.

### **Step 3: Migrate User**

Iterate over the user proxies determined in step 2:

- Iterate over the user agreement consents of the currently selected user proxy determined in step 2:
  - If the date “validFromExistingCustomer” of the newest version of the user agreement the currently selected user agreement consent is associated to is smaller than today or equal to today:
    - Migrate the user agreement consent to the newest version of the user agreement.
    - If the flag *isInformed* of the user agreement consent batch meta data associated to the migrated user agreement consent is FALSE, then log a warning in the batch log file.
    - Delete the user agreement consent batch meta data associated to the migrated user agreement consent.
    - Call AF\_DetermineSummaryForUser with the user-ID as input to get aggregated information about the user’s service availability.
    - For each of the user’s vehicles:
      - Call AF\_UpdateMBcServiceAvailability to inform CPD about the updated service availability.

#### **7.5.8.2 Input**

None.

#### **7.5.8.3 Output**

None.

#### **7.5.8.4 Exceptions**

None.

### **7.5.9 AF\_RegisterVehicleToUser**

See external interface IF\_SOE\_InformOfVehicleRegistration (→ section 7.3.5)

#### **7.5.9.1 General Description**

This application function registers the given customer with the given vehicle. Additionally the first registration date is set, the available services are determined, the licenses for the give-away services are created and the customer is informed.

---

The list of services that are potentially available for the vehicle and, for each service, the status of the matching user agreement is returned. Additionally a list of relevant licenses including the license durations is returned.

### 7.5.9.2 Sequence Description

#### Step 1: Assign vehicle to user

Call the internal interface IIF\_AssignUserProxyToVehicleProxy with the given userId, addressCountry and Fin as input parameters to assign the vehicle to the customer. Remember the output.

#### Step 2: Get vehicle configuration

Call the internal interface IIF\_FetchVehicleData to retrieve the vehicle. Use the given input parameter fin, the preferred language of the previously retrieved customer (field PREFERRED\_LANGUAGE) and the address country of the customer as parameters. If that AF aborts with an error, abort with that error, too.

#### Step 3: Set first registration date

Call the internal interface IIF\_UpdateFirstRegistrationDate with input parameters fin and FirstRegistrationDate to set the first registration date, if it is not set so far. If that IIF aborts with an error, abort with that error, too.

#### Step 4: Update availabilities and licenses

Call AF\_UpdateAvailabilititesAndLicensesForUser with the UserProxy (from step 1) and the list of VehicleProxy (from step 1) and vehicle configuration (from step 2) to update the availabilities and the licenses.

#### Step 5: Trigger document/email sending

Trigger the user information by calling IIF\_InformCustomer with the following parameters:

- DocumentTrigger: VEHICLE\_REGISTER
- with the given UserType
- vehicle: the retrieved vehicle

If that AF aborts with an error, abort with that error, too.

#### Step 6: Return the availabilities, licenses and user agreement consents

Call AF\_DetermineSummaryForUser with the given userId and fin and move the result to output.

### 7.5.9.3 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Customer Profile (See Handling of Profile Data within SOE chapter 2.3.18)</b>						
User	Mand.	-		UserType		The whole profile is requested.
<b>Payload</b>						
Fin	Mand.	String	17		Example: WDD1690341 J764507	The FIN of the vehicle which shall be registered to a customer.
FirstRegistrationDate	Mand.	Date	-		Example: 01.01.2014	The first registration date of the vehicle to register.

Table 253: **Error! Reference source not found.** input

### 7.5.9.4 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of services availabilities (0...*)</b>						
Serviceld	Mand.	Integer	10	Service.Serviceld	Examples: 1, 2, 3, ...	The ID of the service.
UserId	Mand.	String	50	Input parameter from calling AF.		The user ID of the customer account whose services are being transferred.
UserAgreementStatus	Mand.	String		UserAgreementConsent. userAgreementAcceptanceStatusEnum	{"ACCEPTED", "REQUIRED", "INAPPLICABLE"}	INAPPLICABLE: no user agreement needed for service  REQUIRED: user agreement needed, but not yet accepted  ACCEPTED: user agreement has been accepted
FIN	Mand.	String	17	Contract.Vehicle.FIN	Example: WDD169034 1J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
Licenceld	Opt.	String	50	License.LicenseID		License ID
<b>List of all licenses for given users</b>						
Licenceld	Mand.	String	50	License.LicenseID		Die Liste der Lizenzen, die obere Lizenzen IDs referenzieren
LicenseStartType	Mand.	LicenseOrder.StartTypeEnum		LicenseOrder.StartType	LicenseStartEnum [PREDEFINED_START_DATE, INITIAL_SERVICE_ACTIVATION]	
PredefinedStartDate	Opt.	Date		LicenseOrder.PredefinedStartDate	Mand. for LicenseStart Type PREDEFINED_START_DATE	
<b>List of license durations</b>						
LicenseDurationYears	Mand	Int	-	LicenseOrder.DurationYears		
LicenseDurationMonths	Mand	Int	-	LicenseOrder.DurationMonth		
LicenseDurationYears	Mand	Int	-	LicenseOrder.DurationYears		
LicenseDurationUnlimited	Mand	Boolean	-	LicenseOrder.DurationUnlimited		

Table 254: Error! Reference source not found. output

### 7.5.9.5 Exceptions

Message Id	Fault Title	Fault Message
VEHPRO_005	The given vehicle could not be found on the Daimler vehicle database.	If the given vehicle could not be found on ODC.

VEHPRO_006	An unexpected error occurred when trying to retrieve the requested vehicle from ODC.	If things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors).
GENDOC_007	No Document has been found for documentTriggerEnum {0} and OutputTypeEnum {1}.	If the given DocumentTriggerEnum is unknown.

Table 255: Exceptions of **Error! Reference source not found.**

## 7.5.10 AF\_SeparateVehicleFromUser

### 7.5.10.1 General Description

This application function separates the given customer from the given vehicle. The application function deletes the related availabilities, separates the customer from the related licenses, detaches the customer from the vehicle and informs the customer about the separation.

The list of services that are potentially available for the remaining vehicles of the customer and, for each service, the status of the matching user agreement is returned. Additionally a list of licenses including the license durations is returned.

### 7.5.10.2 Sequence Description

#### Step 1: Delete all availabilities

Call the internal interface IIF\_DeleteAvailabilitiesForUserAndVehicle with the given User.UserId and Fin to delete the B2C service availabilities.

#### Step 2: Separate the user from the licenses

In case of separation of the user from vehicle, the licenses stay attached to the vehicle ("Vignette"). Therefore call the internal interface IIF\_SeparateLicensesFromUser with the given User.UserId and Fin in order to separate the user from the existing licenses.

#### Step 3: Detach vehicle from customer

Call the internal interface IIF\_DetachUserProxyFromVehicleProxy with the given User.UserId and Fin as input parameters to separate the vehicle from the user.

#### Step 4: Trigger document/email sending

Call IIF\_FetchVehicleData (see chapter 3.4.9) to retrieve the vehicle that should also be printed in the document. Use the given FIN, the preferred language of the given customer (field PREFERRED\_LANGUAGE) and the address country of the customer. If that AF aborts with an error, abort with that error, too.

Trigger the customer information by calling IIF\_InformCustomer with the following parameters:

- DocumentTrigger: VEHICLE\_SEPARATE
- the given UserType
- Vehicle: the retrieved vehicle

### 7.5.10.3 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Customer Profile</b>						
User	Mand.	-		UserType		The whole profile is requested.

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Payload						
Fin	Mand.	String	17	-	Example: WDD1690341 J764507	The FIN of the vehicle which shall be separated from a customer.

Table 256 AF\_SeparateVehicleFromUser – Input

#### 7.5.10.4 Output

None.

#### 7.5.10.5 Exceptions

Message Id	Fault Title	Fault Message
VEHPRO_005	The given vehicle could not be found on the Daimler vehicle database.	If the given vehicle could not be found on ODC.
VEHPRO_006	An unexpected error occurred when trying to retrieve the requested vehicle from ODC.	If things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors).
GENDOC_007	No Document has been found for documentTriggerEnum {0} and OutputTypeEnum {1}.	If the given DocumentTriggerEnum is unknown.

Table 257 AF\_SeparateVehicleFromUser – Exceptions

### 7.5.11 AF\_SeparateVehicleFromBusinessPartner

#### 7.5.11.1 General Description

This application function separates the given business partner from the given vehicle. The application function deletes the related availabilities and detaches the business partner from the vehicle.

#### 7.5.11.2 Sequence Description

##### Step 1: Delete the availabilities for the vehicle

Call the internal interface **Error! Reference source not found.** with the given BusinessPartnerId and Fin to delete the B2B service availabilities.

##### Step 2: Detach business partner from vehicle

Call the internal interface IIF\_DetachBusinessPartnerProxyFromVehicleProxy with the given BusinessPartnerId and Fin as input parameters to separate the business partner from the vehicle.

#### 7.5.11.3 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
BusinessPartnerId	Mand.	String	20	BusinessPartner-Proxy.BusinessPartnerId	Example: "MBC.DFS"	The ID of the business partner who shall be separated.
Fin	Mand.	String	17	VehicleProxy.Fin	Example: "WDD1690341 J764507"	The European FIN of the vehicle which shall be separated.

Table 258: Table 257 AF\_SeparateVehicleFromUser – Exceptions

---

## AF\_SeparateVehicleFromBusinessPartner input

### 7.5.11.4 Output

None.

### 7.5.11.5 Exceptions

None.

## 7.5.12 AF\_RegisterVehicleToBusinessPartner

### 7.5.12.1 General Description

This application function returns the services that are available for the given business partner in the given vehicle.

### 7.5.12.2 Sequence Description

#### Step 1: Determine availabilities

Call the internal interface **Error! Reference source not found.** with the given BusinessPartner and Vehicle as input parameters to determine the B2B service availabilities.

#### Step 2: Create the summary

For each of the B2B service availabilities, return the corresponding service:

- Set the ServiceId to ServiceAvailabilityB2B.ServiceId
- Set the BusinessPartnerId to the BusinessPartner.BusinessPartnerId from the input
- Set the FIN to ServiceAvailabilityB2B.Fin

Return the result.

### 7.5.12.3 Input

Name	Type / Length / BOM	Description
BusinessPartner	BusinessPartnerProxy	The proxy object for the business partner.
Vehicle	VehicleProxy	The proxy object for the vehicle

Table 259: AF\_ input

### 7.5.12.4 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of services (0...*)						
ServiceId	Mand.	Integer	10	Service.ServiceId	Examples: 1, 2, 3, ...	The ID of the service.
BusinessPartnerId	Opt.	String	20	BusinessPartnerProxy.BusinessPartnerId	Example: "MBC.DFS"	The ID of the business partner.
FIN	Mand.	String	17	Contract.Vehicle.FIN	Example: WDD1690341J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the ser-

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						vice is person specific.

Table 260: AF\_ output

### 7.5.12.5 Exceptions

None.

## 7.5.13 AF\_DetermineSummaryForBusinessPartnerAndVehicle

### 7.5.13.1 General Description

This application function pairs the given business partner with the given vehicle. Additionally the first registration date is set and the available services are determined. The list of services that are potentially available for the vehicle and, for each service, the status of the matching user agreement is returned.

### 7.5.13.2 Sequence Description

#### Step 1: Retrieve proxy objects and assign business partner to vehicle

Call the internal interface IIF\_AssignBusinessPartnerProxyToVehicleProxy with the given BusinessPartnerId and Fin as input parameters to assign the business partner to the vehicle and to retrieve the proxy objects for the business partner and the vehicle.

#### Step 2: Set the first registration date

Call the internal interface IIF\_FetchVehicleDataWithoutLocale with the Fin from the input to retrieve the vehicle configuration.

Call the internal interface IIF\_UpdateFirstRegistrationDate with the Fin and the FirstRegistrationDate from the input as input parameters to update the first registration date.

#### Step 3: Update availabilities

Call the internal interface IIF\_UpdateAvailabilitiesForBusinessPartnerAndVehicle with the proxy object for the business partner, the proxy object for the vehicle and the vehicle configuration to update the services that are available for the business partner in the vehicle.

#### Step 4: Return availabilities

Call the AF\_ with the proxy object for the business partner and vehicle 1 and return the result as output.

### 7.5.13.3 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Customer Profile (See Handling of Profile Data within SOE chapter 2.3.18)</b>						
BusinessPartnerId	Mand.	String	20	BusinessPartner-Proxy.BusinessPartnerId	Example: "MBC.DFS"	The ID of the business partner.
<b>Payload</b>						
Fin	Mand	String	17	VehicleProxy.Fin	Example:	The FIN of the vehicle

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
.	.				WDD1690341 J764507	which shall be registered to a customer.
FirstRegistrationDate	Mand.	Date			Example: 01.01.2014	The first registration date of the vehicle to register.

Table 261: **Error! Not a valid bookmark self-reference.** input

## Output

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of B2B services availabilities (0...*)						
Serviceld	Mand.	Integer	10	Service.Serviceld	Examples: 1, 2, 3, ...	The ID of the service.
BusinessPartnerId	Mand.	String	20	BusinessPartnerProxy.BusinessPartnerId	Example: "MBC.DFS"	The ID of the business partner.
FIN	Mand.	String	17	Contract.Vehicle.FIN	Example: WDD1690341J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.

Table 262: **Error! Not a valid bookmark self-reference.** output

### 7.5.13.4 Exceptions

Message Id	Fault Title	Fault Message
VEHPRO_005	The given vehicle could not be found on the Daimler vehicle database.	If the given vehicle could not be found on ODC.
VEHPRO_006	An unexpected error occurred when trying to retrieve the requested vehicle from ODC.	If things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors).

Table 263: Exceptions of **Error! Not a valid bookmark self-reference.**

## 7.5.14 AF\_InformOfVehicleSeparation

### 7.5.14.1 General Description

See external interface **Error! Reference source not found..**

### 7.5.14.2 Sequence Description

If the User is given as input parameter, call the application function AF\_SeparateVehicleFromUser.

If the BusinessPartner is given as input parameter, call the application function Table 257 AF\_SeparateVehicleFromUser – Exceptions

AF\_SeparateVehicleFromBusinessPartner.

Return an empty list of service availabilities and an empty list of licenses as output.

### 7.5.14.3 Input

See external interface **Error! Reference source not found..**

---

#### **7.5.14.4 Output**

See external interface **Error! Reference source not found..**

#### **7.5.14.5 Exceptions**

See external interface **Error! Reference source not found..**

### **7.5.15 AF\_SetUserAgreementStateOfConsent**

See external interface IF\_SOE\_SetUserAgreementStateOfConsent (→ section 7.3.7)

#### **7.5.15.1 Sequence Description**

Do once in the beginning:

##### ***Step 1: Retrieve Customer Data***

Call the internal interface IIF\_GetUserProfile with the given user ID to retrieve the customer data <User> and UserProxy.

##### ***Step 2: Check Address Country of Customer***

Call IIF\_IsMBconnectCountry with <User>.<Address\_Country> as Country. In case the country is supported by MBconnect, proceed. Otherwise, exit with exception ACCDAS\_005.

##### ***Step 3: Perform Validity Check for given User Agreements***

Call IIF\_GetUserAgreementsByIdAndVersion to get the matching user agreement using the input from the input parameters. If one of the given user agreements is not found the input data is corrupt: Abort the processing and return GETDOC\_003.

For all User Agreements do:

##### ***Step 4: Check User Agreement Version (only when accepting a user agreement)***

If the user agreement is going to be accepted:

Call IIF\_GetAllUserAgreementsWithStateForUser with the parameters <User.userID>, <User.addressCountry> to retrieve all valid user agreements for the user. If the current user agreement is not contained in the retrieved list, mark the agreement as not able to set and skip step 5 to 10 for this document. “Contained in the list” means that the list contains a document whose DocumentId and Version match the currently examined document.

If the user agreement is going to be revoked: Skip this check.

##### ***Step 5: Determine the change type of acceptance status for each User Agreement***

Compare for each user agreement of a user the new status with the corresponding old status. If the acceptance status of a user agreement will be changed, the user agreement will be assigned to one of the following <UserAgreement> list:

- <userAgreementIdsFromAcceptedToRevoked>: The old acceptance status of user agreement was accepted, and the new status will be revoked.
- <userAgreementIdsFromRevokedToAccepted>: The old acceptance status of user agreement was revoked, and the new status will be accepted.

- 
- <userAgreementIdsNewAccepted>: The user agreement will be new associated with the user, and the acceptance status will be accepted.

If the acceptance status of a user agreement will not be changed, the user agreement will be assigned none of these <UserAgreement> lists.

#### **Step 6: Set State of Consent**

Call IIF\_GetUserAgreementConsents

See AF.GetUserAgreementConsents(→ section 9.5.11).

IIF\_SetUserAgreementConsent with UserProxy (step 1), uaID, uaAcceptanceStatus, uaVersion and uaAcceptanceLocale of user agreement and futher required metadata (systemId, orgId and userId). This function sets the desired acceptance status for the given user agreement.

#### **Step 7: Get Customer Vehicles**

Call the internal interface IIF\_GetVehicleProxiesForUser with the UserProxy.userId as input parameter to retrieve the vehicles that are assigned to the user.

#### **Step 8: Determine current contracts & Inform CPD**

Determine the currently available services and licenses for the customer by calling AF\_DetermineSummaryForUser with userId as input parameter (no updates of contracts necessary). Using the output data call AF\_UpdateMbcServiceAvailability in order to inform CPD about the current status of available services for the customer and his vehicles and the according acceptance of the related usage agreements.

#### **Step 9: Determine Performed Action**

If the given user agreement acceptance status is “ACCEPTED”, then <PerformedActionEnum> is “USER AGREEMENT ACCEPTED”. If the given user agreement acceptance status is “REVOKED”, then <PerformedActionEnum> is “USER AGREEMENT REVOKED”.

#### **Step 10: Inform Customer**

For each affected list of <UserAgreement> from step 5 build a map consisting from each user agreement and the locale in which the user agreement has been signed. Call the internal interface IIF\_InformCustomer to inform the customer about his newly accepted/revoke user agreements, with the parameters:

- The determined <PerformedActionEnum> from step 9
- Map<UserAgreement, locale>
- The given User (determined in step 1)
- Omit vehicle configuration

User agreements, which acceptance status was not changed , was not assigned to any of the <UserAgreement> list in step 5, so customer will not be informed for these user agreement.

Do once in the end:

#### **Step 11: Return Success Status**

---

If all user agreements consents could be set, return with  
<AllUserAgreementsConsentsSet> = true. Otherwise, set  
<AllUserAgreementsConsentsSet> = false and attach a list of all user agreements  
which consents could not be set.

### 7.5.15.2 Input

See external interface IF\_SOE\_SetUserAgreementStateOfConsent (→ section 7.3.7)

### 7.5.15.3 Output

See external interface IF\_SOE\_SetUserAgreementStateOfConsent (→ section 7.3.7)

### 7.5.15.4 Exceptions

See external interface IF\_SOE\_SetUserAgreementStateOfConsent (→ section 7.3.7)

## 7.5.16 AF\_UpdateAvailabilititesAndLicensesForUser

This application function updates the service availabilities and the licenses for the given vehicles and the given customer.

### 7.5.16.1 Sequence Description

For each given vehicle do the following:

#### Step 1: Get all available services for user and vehicle

Call IIF\_GetAvailableServicesByConfiguration with parts of the given vehicle configuration and <UserProxy.custAddressCountry> to retrieve the enabled services available for the vehicle and the users address country.

#### Step 2: Update availabilitties

Call IIF\_UpdateAvailabilitiesForUserAndVehicle with UserProxy, VehicleProxy and the list of available services for the combination of user and FIN.

#### Step 3: Get service masters for the list of service ids

Call IIF.GetServiceMasterData and pass in the list of service ids (from step 1) to get all relevant service masters.

#### Step 4: Update licenses

Call IIF\_UpdateLicensesForUserAndVehicle with UserProxy, VehicleProxy, FirstRegistrationDate and the list of following service attributes:  
licenseRequired, ContractStartTrigger, contractDuration, serviceMasterID.

### 7.5.16.2 Input

Name	Type / Length / BOM	Description
UserProxy	UserProxy	The user proxy of the user whose contracts will be updated.
List of Vehicles and VehicleProxies		
Vehicle	Vehicle	Detailed information of vehicle (including its configuration, model serie and first registration date).
VehicleProxy	VehicleProxy	Vehicle proxy assigned to the given UserProxy and used in next steps to link availability or license entities to VehicleProxy

Table 264: AF\_UpdateAvailabilititesAndLicensesForUser input

### **7.5.16.3 Output**

None.

### **7.5.16.4 Exceptions**

Message Id	Fault Message	Error reason
ACCDAS_002	User does not exist	if the given user could not be resolved in the CPD.
ACCDAS_004		if any unexpected, technical errors occur when connecting to CPD.

Table 265: AF\_UpdateAvailabilititesAndLicensesForUser exceptions

## **7.5.17 AF\_UpdateMbcServiceAvailability**

Informs CPD about the current status of available services for the customer, his vehicles and the according acceptance of the related user agreements.

### **7.5.17.1 Sequence Description**

Call the external interface IF\_CPD\_UpdateMbcServiceAvailability with the given input parameters.

### **7.5.17.2 Input**

See external interface IF\_CPD\_UpdateMbcServiceAvailability.

### **7.5.17.3 Output**

None.

### **7.5.17.4 Exceptions**

None.

## **7.5.18 AF\_UpdateServiceAvailabilityForUsersBatch**

### **7.5.18.1 General Description**

This batch updates availabilities and licenses if the service availability for a customer has been changed caused by

- disablement or enablement of services
- adding or removing the service availability for particular MBconnect countries
- creating, deleting, modifying service assignment rules

and informs CPD about changed service availability.

### **7.5.18.2 Sequence Description**

#### **Step 1: Retrieve all existing B2C-services and determine the set of potentially affected vehicles and users**

1. Call IIF\_GetAllServices with input parameters onlyEnabledServices=false and BusinessArea = B2C in order to retrieve all instances of B2C-Service entity that exist in SOE.
2. Call IIF\_GetAllVehiclesProxiesAndAssociatedUserProxy to get each vehicle proxy and for each vehicle proxy the associated user proxy

- 
3. For each vehicle proxy call IIF\_RetrieveModelSeriesForFIN to determine the model series for each FIN
  4. Store the tuple <FIN, ModelSeries; UserID, AddressCountry> in the data structure **POTENTIALLY\_AFFECTED\_VEHICLES\_AND\_USERS**

### **Step 2: Determine types of changes: affected model series-country combinations**

**Step 2.0: Initialize an empty list *CHANGES* of model series with an inner list of countries per model series**

**Step 2.1: Model series and countries affected by enablement or disablement of services**

1. From the list of services from step 1 determine the list of those services that need to be enabled or disabled:
  1. For disabled services: Service.enabledTo is given and is smaller than today and greater than the date of the last batch run.
  2. For enabled services: Service.enabledFrom is equal to or smaller than today and greater than the date of the last batch run and Service.enabledTo is empty or greater than today.
2. For each service call IIF\_GetModelSeriesForService to determine a list of unique model series.
3. For each service call IIF\_GetMBcCountriesForServices to determine the countries each service is available in.
4. Call AF\_JoinModelSeriesAndCountriesByService with the lists from steps 2.1.2 and 2.1.3.
5. Add the result of step 2.1.4 to *CHANGES*.

**Step 2.2: Model series and countries affected by changes of service country assignments**

1. Call IIF\_RetrieveServiceCountryAssignmentChangeLog twice with input parameter “DELETE” and “NEW” in order to determine each ServiceCountryAssignmentChangeLog entity that logs the deletion or creation of an assignment between an enabled Service and an MBcCountry.
2. From the returned data, create a list of unique ServiceCountryAssignmentChangeLog.serviceIDs and for each service an inner list of unique ServiceCountryAssignmentChangeLog.countryCodes.
3. For each service call IIF\_GetModelSeriesForService to determine a list of unique model series.
4. Call AF\_JoinModelSeriesAndCountriesByService with the lists from steps 2.2.2 and 2.2.3.
5. Merge the result of step 2.2.4 with the content of *CHANGES*.
6. Call IIF\_DeleteServiceCountryAssignmentChangeLog with the list of retrieved ServiceCountryAssignmentChangeLog entities.

**Step 2.3: Model series and countries affected by changes of service assignment rules**

While the service assignment rule change log is not empty:

- 
1. Call IIF\_RetrieveOldestServiceAssignmentRuleChangeLog in order to get the list of model series and services that are affected by the oldest modification of a service assignment rule that is logged in SOE (ServiceAssignmentRuleChangeLog.ModelSeries and ServiceAssignmentRuleChangeLog.ServiceID).
  2. From the returned data, create a list of unique ServiceAssignmentRuleChangeLog.serviceIDs and for each service an inner list of unique ServiceAssignmentRuleChangeLog.modelSeries.
  3. For each service call IIF\_GetMBcCountriesForServices in order to determine the list of MBc countries each service is available in.
  4. Call AF\_JoinModelSeriesAndCountriesByService with the lists from steps 2.2.3 and 2.2.4.
  5. Merge the result of step 2.3.3 with the content of *CHANGES*.
  6. Call IIF\_DeleteServiceAssignmentRuleChangeLog with the ServiceAssignmentRuleChangeLog retrieved in step 1 in order to delete the log entry.

### **Step 3: Filter potentially affected vehicle proxy-user proxy combinations**

Call AF\_FilterPotentiallyAffectedVehiclesAndUsers to filter the vehicles and users stored in *POTENTIALLY\_AFFECTED\_VEHICLES\_AND\_USERS* by model series and address country applying the filter defined by the data structure *CHANGES*.

### **Step 4: Update service availability and determine the pre-update post-update difference**

1. Iterate over the list of pairs of vehicle proxy and user proxy *POTENTIALLY\_AFFECTED\_VEHICLES\_AND\_USERS*
  1. Call AF\_DetermineSummaryForUser for the current combination of vehicle proxy and user proxy in to create a snapshot of the service availability before the update.
  2. Call AF\_UpdateAvailabilitiesAndLicensesForUser for the current combination of vehicle proxy and user proxy to update the service availability.
  3. Call AF\_DetermineSummaryForUser for the current combination of vehicle proxy and user proxy in to create a snapshot of the service availability after the update.
2. Compare the snapshots of the service availabilities and licenses before and after the update.
  - o During the comparison, log the type of changes by calling the IIF\_LogChangedServiceAvailabilityOrLicense with the respective FIN and user-ID, service-ID, as well as the type of change. The type of change is determined as follows:
    - If a service availability was removed, log SERVICE\_DISABLED.
    - If a service availability was added, log SERVICE\_ENABLED.
    - If a license was removed, log LICENSE\_REMOVED.
    - If a license was added, log LICENSE\_ADDED.
3. If the snapshots before and after the update differ, call AF\_UpdateMBcServiceAvailability to inform CPD of the updated service availability.

---

### **Step 5: Inform adjacent systems of enablement or disablement of services**

In order to inform adjacent systems of each master data change caused by the enablement or disablement of services, call

AF\_InformOfMasterDataChangesAffectedByServiceEnablement with the list of services determined in step 2.1.1.1 and step 2.1.1.2.

#### **7.5.18.3 Input**

None.

#### **7.5.18.4 Output**

None.

#### **7.5.18.5 Exceptions**

None.

### **7.5.19 AF\_JoinModelSeriesAndCountriesByService**

This application function joins two lists of services with an inner list (1) of model series and (2) of countries by the service-ID and groups the result to get a list of unique model series with an inner list of aggregated countries.

#### **7.5.19.1 Sequence Description**

1. Join the two lists given as input by the ID of the services.
2. For each model series collect all countries associated to it to get a list of unique model series and for each model series a list of unique countries.
3. Return the result.

#### **7.5.19.2 Input**

Name	Type / Length / BOM	Description
List of service		
ServiceID	Integer	The ID of the service.
Inner list of model series		
ModelSeriesID	ModelSeriesDT	The code of a model series.
List of service		
ServiceID	Integer	The ID of the service.
Inner list of country		
CountryCode	CountryCodeDT	The ID of a country.

Table 10: AF\_JoinModelSeriesAndCountriesByService Input

#### **7.5.19.3 Output**

Name	Type / Length / BOM	Description
List of model series		
ModelSeriesID	ModelSeriesDT	The code of a model series.
Inner list of country		
CountryCode	CountryCodeDT	The ID of a country.

---

Table 10: AF\_JoinModelSeriesAndCountriesByService Output

#### 7.5.19.4 Exceptions

None.

### 7.5.20 AF\_FilterPotentiallyAffectedVehiclesAndUsers

This application function filters a list of vehicles and associated user proxies given as tuples <FIN, ModelSeries; UserID, AdressCountry> applying a filter containing a list of unique model series with a nested list of countries.

#### 7.5.20.1 Sequence Description

Select all tuples for which

1. the model series matches one of the model series contained in the filter
2. the nested list of countries associated with the model series in the filter contains the address country of the user.

#### 7.5.20.2 Input

Name	Type / Length / BOM	Description
List of vehicle proxies with model series and associated user proxy		
VehicleProxy	ContextProxy::VehicleProxy	A vehicle proxy
ModelSeries	ModelSeriesDT	The model series of the vehicles proxy
UserProxy	ContextProxy::UserProxy	User Proxy associated to the vehicle proxy
List of model series		
ModelSeries	ModelSeriesDT	The code of a model series
List of countries per model series		
CountryCode	CountryCodeDT	The identifier of a country

Table 10: AF\_FilterPotentiallyAffectedVehiclesAndUsers Input

#### 7.5.20.3 Output

Name	Type / Length / BOM	Description
List of <u>filtered</u> vehicle proxies with model series and associated user proxy		
VehicleProxy	ContextProxy::VehicleProxy	A vehicle proxy
ModelSeries	ModelSeriesDT	The model series of the vehicles proxy
UserProxy	ContextProxy::UserProxy	User Proxy associated to the vehicle proxy

Table 10: AF\_FilterPotentiallyAffectedVehiclesAndUsers Output

#### 7.5.20.4 Exceptions

None.

## 7.6 Batches

Batchname	Called Application Function	Description
Batch "Migrate User to a new User Agreement"	AF_MigrateUserToNewUserAgreementBatch	Execution time: night time Execution frequency: daily
Batch "Update Service availability for Users"	AF_UpdateServiceAvailabilityForUsersBatch	Execution time: night time, <b>after</b> "Migrate User to a new User

---

		Agreement"Batch Execution frequency: daily
--	--	---

Figure 67: Component "Contract management" - Batches

## 7.7 Error Messages

Message Id	Fault Title	Fault Message
CONMAN_008	Inconsistent MasterData	The service given is associated to multiple User Agreements.
CONMAN_009	Vehicle assignment does not exist	The requested combination of user ID and FIN does not exist.
CONMAN_010	The interface is not supported	The interface is not supported.

Table 266: Component "Contract management" - error messages

---

## 8 Component “Licenses”

### 8.1 Dialogs

None.

### 8.2 External View - Consumed Interfaces

None.

#### 8.2.1 IF\_CPD\_InformOfLicenseModification

**Communication type:** Synchronously

This interface informs the CPD system about the modification of a license. Therefore SOE sends a list of all license duration (this can be a reduction as well as an extension of a license).

Only for release 2.0: Modification of licenses ist not allowed in Release R2.0. If this interface is called, write an error message “Modification of licenses is not allowed for release 2.0” to the log file and abort with a general functional error.

##### 8.2.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of licenses						
LicenseID	Mand.	String	50	License.LicenseID		The ID of the license.
Inner list of license durations						
LicenseDurationUnlimited	Mand.	Boolean	-	LicenseOrder.DurationUnlimited		
LicenseDurationYears	Mand.	Integer	-	LicenseOrder.DurationYears		
LicenseDurationMonths	Mand.	Integer	-	LicenseOrder.DurationMonths		
LicenseDurationDays	Mand.	Integer	-	LicenseOrder.DurationDays		

Table 267: IF\_CPD\_InformOfLicenseModification Input

##### 8.2.1.2 Output

None.

##### 8.2.1.3 Exceptions

None.

#### 8.2.2 IF\_CPD\_InformOfLicenseRevocation

**Communication type:** Synchronously

This interface informs the CPD system about the revocation of a license.

### 8.2.2.1 Input

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of licenses						
LicenseID	Mand .	String	50	License.LicenseID		The ID of the license.

Table 268: IF\_CPD\_InformOfLicenseRevocation Input

### 8.2.2.2 Output

None.

### 8.2.2.3 Exceptions

None.

## 8.3 External View - Offered Interfaces

### 8.3.1 IF\_SOE\_GetActiveLicensesByID

#### 8.3.1.1 General Description

**Communication type:** Synchronously

This interface retrieves all licenses that match one of the given license IDs and have not yet expired. Invalid license IDs that do not match any licenses will be ignored.

Internally, the AF\_GetActiveLicensesByID is called.

#### 8.3.1.2 Input

Parameter Name	Man d./Opt.	Format	Length	Data Model	Possible Values	Annotation
List [1..20]						
LicenseID	Mand .	String	50	License.LicenseID		

Table 269: IF\_SOE\_GetActiveLicensesByID Input

#### 8.3.1.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List [1..20]						
LicenseID	Mand .	String	50	License.LicenseID		
UserID	Opt.	String	50	UserProxy.UserID		The user ID of the customer.
FIN	Opt.	String	17	VehicleProxy.FIN	Example: WDD169034 1J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.
serviceMasterId	Mand .	Integer	-	ServiceMaster.ServiceMasterID	-	The service master id.
isLicenseRevocable	Mand .	Boolean				

Table 270: IF\_SOE\_GetActiveLicensesByID Output

### **8.3.1.4 Exceptions**

None.

## **8.3.2 IF\_SOE\_ModifyLicenses**

### **8.3.2.1 General Description**

**Communication type:** Synchronously

This interface updates the licenses if the customer makes or revokes a purchase of a license. In case of revocation, the duration is given as negative input value. If the durations of all license orders neutralize themselves, the license will be deleted. Additionally, the CPD is informed about all modifications and revocations of licenses.

Hint: For release 2.0, it will not be possible to extend a license. Instead, an exception will be thrown.

Internally, the AF\_ModifyLicenses is called.

### **8.3.2.2 Input**

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
UserID	Mand.	String	50	UserProxy.userID		The user ID of the customer.
Transactio nID	Mand.	String	60	LicenseOrder.Transac tionID		UniqueReference to ensure idempotence
List [1..20]						
LicenseID	Mand.	String	50	License.LicenseID		
LicenseDu rationUnit	Mand.	ENUM			[DAY,MONT H,YEAR]	
LicenseDu rationValue	Mand.	Integer				In case of revocation of licences the duration has to be negative

Table 271: IF\_SOE\_ModifyLicenses Input

### **8.3.2.3 Output**

None.

### **8.3.2.4 Exceptions**

Message Id	Fault Title	Fault Message
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
LICENSE_001	License does not exist	if the requested license ID does not exist or has expired.
LICENSE_002	License assignment does not exist	If the requested combination of user ID and license ID does not exist.

## **8.3.3 IF\_SOE\_InformOfServiceLicenseExpiration**

**Communication type:** Synchronously

---

This interface informs the SOE about licenses that have expired and need to be deleted. Invalid license IDs that do not match any licenses will be ignored.

Internally, the AF\_InformOfServiceLicenseExpiration is called.

### 8.3.3.1 Input

Parameter Name	Man d./Op t.	Format	Length	Data Model	Possible Values	Annotation
List of licenses						
LicenseID	Mand	String	50	License.LicenseID		The ID of the license.

Table 272: IF\_SOE\_InformOfServiceLicenseExpiration Input

### 8.3.3.2 Output

None.

### 8.3.3.3 Exceptions

None.

## 8.4 Internal View - Consumed Interfaces

## 8.5 Internal View - Offered Interfaces

### 8.5.1 IIF\_CreateLicensesAndLicenseOrders

See AF\_CreateLicensesAndLicenseOrders.

### 8.5.2 IIF\_GetLicensesForUser

See AF\_GetLicensesForUser.

### 8.5.3 IIF\_SeparateLicenseOrdersFromUser

See AF\_SeparateLicenseOrdersFromUser.

### 8.5.4 IIF\_SeparateLicensesFromUser

See AF\_SeparateLicensesFromUser.

### 8.5.5 IIF\_UpdateLicensesForUserAndVehicle

See AF\_UpdateLicensesForUserAndVehicle.

## 8.6 Implementation

### 8.6.1 AF\_CreateLicensesAndLicenseOrders

The application function creates the licenses and persists license related data for the given userId, fin and service master.

#### 8.6.1.1 Sequence Description

Do the following for the given input parameters:

- 1) Create the license instances based on the given parameters:

- a. Generate license id, where next sequential number will be created and assigned to the attribute License.LicenseID.
  - b. Set the attribute License.User to UserProxy.userId.
  - c. Set the attribute License.Vehicle to VehicleProxy.fin.
  - d. Set the attribute License.ServiceMaster to serviceMasterId.
  - e. Set the value of License.licenseStartType to licenseStartType.
  - f. Set the value of License.predefinedStartDate to PredefinedStartDate.
- 2) Create the license order:
- a. Set the value of LicenseOrder.TransactionID to given Transaction ID.
  - b. Set the attribute LicenseOrder.licenseOrders to License.LicenseID created above.
  - c. Set the attribute LicenseOrder.OrderingUser to UserProxy.userId.
  - d. Set the attribute LicenseOrder.ReferencedVehicle to VehicleProxy.fin.
  - e. Set the value of enumeration LicenseOrder.orderSource to the given OrderSourceEnum value.
  - f. Set the value of LicenseOrder.StartType to licenseStartType.
  - g. Set the value of LicenseOrder.predefinedStartDate to PredefinedStartDate.
  - h. Set the value of LicenseOrder.licenseOrderDate to the current date.
  - i. Set the value of LicenseOrder.durationDays to the LicenseDurationDays.
  - j. Set the value of LicenseOrder.durationMonths to the LicenseDurationMonths.
  - k. Set the value of LicenseOrder.durationYears to the LicenseDurationYears.
  - l. Set the value of LicenseOrder.durationUnlimited to the durationUnlimited.
  - m. Set the value of LicenseOrder.additionalActivationReference to given additionalActivationReference.
- 3) Return the list of created license ids, fin and service master ids.

### 8.6.1.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
UserProxy	Mand.	UserProxy		UserProxy		The user proxy of the customer.
TransactionID	Mand.	String	60	LicenseOrder.TransactionID		UniqueReference to ensure idempotence
orderSource	Mand.	OrderSourceEnum		LicenseOrder.orderSource	[GIVEAWAY, ORDERED]	
List of license durations						
o VehicleProxy	Mand.	VehicleProxy		VehicleProxy	Example: WDD169034 1J764507	The vehicleproxy assigned to the given userproxy.
o serviceMasterId	Mand.	Integer	-	ServiceMaster.ServiceMasterID		The service master id.
o LicenseDurationYears	Mand.	Int	-	LicenseOrder.DurationYears		
o LicenseDurationMonths	Mand.	Int	-	LicenseOrder.DurationMonth		
o LicenseDurationDays	Mand.	Int	-	LicenseOrder.DurationDays		
o durationUnlimited	Mand.	Boolean		LicenseOrder.durationUnlimited		
o LicenseStartType	Mand.	LicenseOrderStartTypeEnum		LicenseOrder.StartType	LicenseStartEnum [PREDEFINED_START_DATE, INITIAL_SERVICE_ACTIVE]	

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
				TION]		
o PredefinedStartDate	Opt.	Date		LicenseOrder.PredefinedStartDate	Mand. for LicenseStart Type PRE-DE-FINED_STA RT_DATE	
o additionalActivationReference	Opt.	String	50	LicenseOrder.additionalActivationReference		

Table 273: AF\_CreateLicenses Input

### 8.6.1.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List [1..20]</b>						
LicensesID	Mand.	String	50	License.LicenseID		
FIN	Opt.	String	17	VehicleProxy.FIN	Example: WDD169034 1J764507	The European FIN of the vehicle the service is specific to. Empty for person related services.  If a FIN is given, the service is vehicle specific. If a FIN is not given the service is person specific.
serviceMasterId	Mand.	Integer	-	ServiceMaster.ServiceMasterID	-	The service master id.

Table 274:AF\_CreateLicensesAndLicenseOrders Output

### 8.6.1.4 Exceptions

Message Id	Fault Title	Fault Message
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
ACCDAS_004	An unexpected error occurred while connecting to the system CPD.	if any unexpected, technical errors occur when connecting to CPD.

## 8.6.2 AF\_GetActiveLicensesByID

### 8.6.2.1 General Description

See external interface IF\_SOE\_GetActiveLicensesByID.

### 8.6.2.2 Sequence Description

Retrieve the licenses that match the license IDs from the input. If none of the license IDs from the input matches a license, return an empty list (= no error).

Set the output parameters according to the values from the licenses. Set the output parameter isLicenseRevocable to "True" if the License.UserProxy.UserId is not empty and matches the LicenseOrder.UserProxy.UserId from the last LicenseOrder in the list of License.licenseOrders and if the LicenseOrder.orderDate of the last LicenseOrder in the list of License.licenseOrders is not older than 14 days or to "False" otherwise.

### **8.6.2.3 Input**

See external interface IF\_SOE\_GetActiveLicensesByID.

### **8.6.2.4 Output**

See external interface IF\_SOE\_GetActiveLicensesByID.

### **8.6.2.5 Exceptions**

See external interface IF\_SOE\_GetActiveLicensesByID.

## **8.6.3 AF\_GetLicensesForUser**

This application function retrieves all licenses referencing the given input parameters. Also the data from the related table LicenseOrder will be retrieved.

### **8.6.3.1 Sequence Description**

Do the following for the given userId:

- Select all licenses that match the given input parameters (License.User (and License.Vehicle if fin is given)).
- Determine the LicenseOrder entries referencing the selected licenses.
- Return the required fields from all these tables as an output.

### **8.6.3.2 Input**

Name	Type / Length / BOM	Description
UserId	String	Mandatory;
Fin	String	Optional; the European FIN.

Table 275: AF\_GetLicensesForUser Input

### **8.6.3.3 Output**

Name	Type / Length / BOM	Description
List of all licenses for given users		
Licences	List of Licenses	
Inner List of license orders (1..*)		
LicenseOrders	List of LicenseOrders	

Table 276: AF\_GetLicensesForUser Output

### **8.6.3.4 Exceptions**

None.

## **8.6.4 AF\_InformOfServiceLicenseExpiration**

### **8.6.4.1 General Description**

See external interface IF\_SOE\_InformOfServiceLicenseExpiration.

### **8.6.4.2 SequenceDescription**

Delete all licenses that match the license IDs from the input including their license orders. If a license ID from the input does not match any of the existing licenses, then ignore this license ID.

---

#### **8.6.4.3 Input**

See external interface IF\_SOE\_InformOfServiceLicenseExpiration.

#### **8.6.4.4 Output**

See external interface IF\_SOE\_InformOfServiceLicenseExpiration.

#### **8.6.4.5 Exceptions**

See external interface IF\_SOE\_InformOfServiceLicenseExpiration.

### **8.6.5 AF\_ModifyLicenses**

#### **8.6.5.1 General Description**

See external interface IF\_SOE\_ModifyLicenses.

#### **8.6.5.2 Sequence Description**

##### Step 1: Validate the licenses

Retrieve the licenses that match the LicenseIDs from the input. Assert that all licenses exist. Otherwise, abort with error code LICENSE\_001.

##### Step 2: Create the license orders and determine the total durations of the licenses

For the given UserId, determine the corresponding UserProxy by calling IIF\_GetOrCreateOrUpdateUserProxy.

For each license determined in step 1, append a new LicenseOrder to the ordered list of License.LicenseOrders:

- Connect the LicenseOrder to the UserProxy, i.e. set LicenseOrder.OrderingUser = UserProxy
- Connect the LicenseOrder to the same VehicleProxy as the License
- Leave the LicenseOrder.AdditionalActivationReference empty
- Set LicenseOrder.DurationUnlimited to “False”
- Set the duration that corresponds to the LicenseDurationUnit from the input to the LicenseDurationValue from the input and set all other durations to “0”, e.g. set the LicenseOrder.DurationYears to the LicenseDurationValue from the input if the LicenseOrderUnit is set to “YEARS”
- Set the LicenseOrder.LicenseOrderDate to the current date
- Leave the LicenseOrder.PredefinedStartDate empty
- Set the LicenseOrder.StartType to “MODIFICATION\_OF\_LICENSE”
- Set the LicenseOrder.OrderSource to “ORDERED”
- Set the LicenseOrder.TransactionId to the TransactionID from the input
- Connect the LicenseOrder to the License

Calculate the total duration of the license as follows: For each LicenseOrder in License.LicenseOrders, add the LicenseOrder.DurationYears, LicenseOrder.DurationMonths and LicenseOrder.DurationDays to the total duration in years, months or days, i.e. calculate a separate “total” duration per unit. If the LicenseOrder.DurationUnlimited of one of the license orders is set to “True”, the total duration of the license is unlimited.

---

If the total duration is unlimited or one of the total durations in years, months or days is positive, add the license to the list of modified licenses. Otherwise, add the license to the list of revoked licenses.

### Step 3: Inform the CPD about modifications and revocations of licenses

If the list of modified licenses is not empty, call the interface **IF\_CPD\_InformOfLicenseModification**. For each License in the list of modified licenses

- Use the License.LicenseId as input parameter LicensId
- For each LicenseOrder in the License.licenseOrders, append the LicenseOrder.durationUnlimited, LicenseOrder.durationYears, LicenseOrder.durationMonths and LicenseOrder.durationDays to a list of durations and use this list as the input parameter for the license durations

If the list of revoked licenses is not empty, call the interface **IF\_CPD\_InformOfLicenseRevocation** with a list of all License.licenseId from the list of revoked licenses as input parameter. Afterwards, delete all licenses from the list of revoked licenses including their license orders.

#### **8.6.5.3 Input**

See external interface **IF\_SOE\_ModifyLicenses**.

#### **8.6.5.4 Output**

See external interface **IF\_SOE\_ModifyLicenses**.

#### **8.6.5.5 Exceptions**

See external interface **IF\_SOE\_ModifyLicenses**.

### **8.6.6 AF\_ReassignLicenses**

#### **8.6.6.1 General Description**

This application function is used to reassign the given licenses to the given user.

#### **8.6.6.2 Sequence Description**

Do the following for the given list of licenses and user:

1. Select all licenses that match the given license ids.
2. Replace existing user reference (License.User) with the given UserProxy.

#### **8.6.6.3 Input**

Name	Type / Length / BOM	Description
UserProxy	UserProxy	The user proxy object of the user to whom licenses will be assigned
LicensIds	List of LicensId	List of licenses ids need to be assigned to new user

Table 277: AF\_ReassignLicenses Input

#### **8.6.6.4 Output**

None.

---

### **8.6.6.5 Exceptions**

None.

### **8.6.7 AF\_SeparateLicenseOrdersFromUser**

This application function deletes the UserProxy reference from LicenseOrders for the given userId.

#### **8.6.7.1 Sequence Description**

Step 1: Select all license orders that match the given input parameters (LicenseOrder.OrderingUser = userId).

Step 2: For each of the license orders remove the user reference (LicenseOrder.OrderingUser).

#### **8.6.7.2 Input**

Name	Type / Length / BOM	Description
UserId	String	The user which has to be separated from the licenses.

Table 278: AF\_SeparateLicenseOrdersFromUser Input

#### **8.6.7.3 Exceptions**

None.

### **8.6.8 AF\_SeparateLicensesFromUser**

This application function separates the user from vehicles licenses.

#### **8.6.8.1 Sequence Description**

Step 1: Select all licenses that match the given input parameters (License.vehicle and License.user). Note: If no FIN is given, select all the licenses referencing the user.

Step 2: For each of the licenses remove the user reference (License.User).

#### **8.6.8.2 Input**

Name	Type / Length / BOM	Description
UserId	String	The user which has to be separated from the licenses.
Fin	String	Optional; The vehicle, which licenses has to be separated from the user.

Table 279: AF\_GetLicenses Input

#### **8.6.8.3 Output**

None.

#### **8.6.8.4 Exceptions**

None.

### **8.6.9 AF\_UpdateLicensesForUserAndVehicle**

The AF reassigns the existing licenses to a new user. Afterwards it creates new licenses for the given services which require licenses and are available for user and his vehicle (e. g. "GiveAway" case by vehicle registration).

### **8.6.9.1 Sequence Description**

For the given input parameters do the following:

#### Step 1: Get existing Licenses for the vehicle

Select all licenses that match the FIN of the given VehicleProxy .

#### Step 2: Reassign licenses

Call the AF\_ReassignLicenses and pass in UserProxy and the list of license ids retrieved from step 1 to replace existing user reference (License.User) with the given user.

#### Step 3: Create licenses for “GiveAway”-services

Do the following for the given UserProxy, VehicleProxy and list of service attributes:

1. Check the value of the attribute “licenseRequired” of given service entity: if it is set to “false”, then no creation of license for this service is needed. If the value is set to “true” go to the next step.
2. Check the value of enumeration “ContractStartTrigger” of service: if it is set to “LICENSE\_PURCHASE”, then no license is created.  
If the value is set to “FIRST\_REGISTRATION\_VEHICLE” or “INITIAL\_SERVICE\_ACTIVATION” go to the next step.
3. If there is already a license existing (see step 1) for the given service master (License.serviceMasterId = Service.serviceMasterId), no license is created.
4. If “ServiceGiveAwayLogBook” contains the given vehicle and service master id (the service have been offered to user during the customer loyalty programs → “Dreingabe”), then no license is to create. Otherwise go to next step 3.5:
5. Create temporally integer variable LicenseDurationMonth and Boolean variable LicenseDurationUnlimited.
6. If a service contract duration is given (contractDuration is not NULL) then set LicenseDurationMonth (created in step 5) to contractDuration and LicenseDurationUnlimited (created in step 5) to “FALSE”, if contract duration is not given (contractDuration is NULL) then set the LicenseDurationMonth to “0” and LicenseDurationUnlimited to “TRUE”.
7. Call AF\_CreateLicensesAndLicenseOrders with the following input values:
  - UserProxy (given in input),
  - TransactionID: Generate unique id with the prefix “SOE”.
  - orderSource.GIVEAWAY
  - VehicleProxy (given in input),
  - serviceMasterId (given in input),
  - “0” (as LicenseDurationYears parameter),
  - LicenseDurationMonth (from step 6),
  - “0” (as LicenseDurationDays parameter),
  - LicenseDurationUnlimited (from step 6),
  - contractStartTrigger (given in input; provide as LicenseStartType parameter)
  - If contractStartTrigger = “FIRST\_REGISTRATION\_VEHICLE” then Vehicle.firstRegistrationDate as PredefinedStartDate else No PredefinedStartDate is provided.
  - actual date ( as licenseOrderDate parameter)

Ignore the data returned from AF\_CreateLicenses.

#### Step 3: Log the “GiveAway”-services

---

Persist the services that are licensed as “GiveAway” services in “ServiceGiveAwayLogBook” table. Do therefore following:

- set into ServiceGiveAwayLogBook.Vehicle the VehicleProxy.fin(given in input),
- set into ServiceGiveAwayLogBook.ServiceMasterId the serviceMasterId (given in input)

### 8.6.9.2 Input

Name	Type / Length / BOM	Description
UserProxy	UserProxy	UserProxy of the customer for whom the new licenses will be created.
VehicleProxy	VehicleProxy	VehicleProxy assigned to given UserProxy
FirstRegistrationDate	Date	The first registration date of the vehicle to register.
Inner list of service attributes		
licenseRequired	Boolean	Tells whether a license is required or not in order to use this service.
ContractStartTrigger	ContractStartTriggerEnum	Indicates what event triggers the beginn of a license creation related to this service.
contractDuration	Integer	Describes the contract duration in months, in case a contract for this service is established. This attribute is optional.
serviceMasterID	Integer	Points to the ServiceMaster this Service is assigned to.

Table 280: AF\_UpdateLicensesForUserAndVehicle input

### 8.6.9.3 Output

None.

### 8.6.9.4 Exceptions

Message Id	Fault Title	Fault Message
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
ACCDAS_004	An unexpected error occurred while connecting to the system CPD.	if any unexpected, technical errors occur when connecting to CPD.

## 8.7 Batches

None.

## 8.8 Error Messages

Message Id	Fault Title	Fault Message
LICENSE_001	License does not exist	The requested license ID does not exist or has expired.
LICENSE_002	License assignment does not exist	The requested combination of user ID and license ID does not exist.
LICENSE_003	Invalid license duration	The licens duration must be greater then zero.

Table 281: Messages of the component Licenses

## 9 Component „Consents“

### 9.1 Dialogs

#### 9.1.1 DLG\_PrintApproval

This dialog shows the number of letters and emails, which should be sent to the customers due to changed user agreements. The number of documents will be selected by a batch every night and accumulated each night until the print approval is done. Thereafter, counting starts again from the beginning.



Figure 68: DLG\_PrintApproval

#### 9.1.1.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		Load the number of documents that are to be printed. This is the number of <UserAgreementConsentBatchMetadata.isInformed>="FALSE" and <UserAgreementConsentBatchMetadata.isApproved>="FALSE". Accumulate each night the number of documents after the batch is running until the print approval is done.
"Print Approval"	Button	Call AF_PrintApproval (-> see chapter 9.5.12). Refresh the dialog and start again with accumulation from zero.

Table 282: Buttons and functions (new dialogue)

#### 9.1.1.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Dialog Description-Label	Label	"Number of changed documents through which the user should be informed."	-
Number-Label	Label	"Print approval for <number> emails and letters is requested."	-
<number>	String	The Number is determined through the initialization of the dialog.	Count UserAgreementConsentBatchMetadata with UserAgreementConsentBatchMetadata.isInformed = FALSE and UserAgreementConsentBatchMetadata.isApproved = FALSE

Table 283: Form fields and front-end data objects (new dialogue)

### 9.1.1.3 Dialogue field validation

None.

### 9.1.1.4 Configurability (incl setting for roles)

None.

### 9.1.1.5 Dialog Elements State

None.

## 9.2 External View - Consumed Interfaces

None.

## 9.3 External View - Offered Interfaces

### 9.3.1 IF\_SOE\_GetAvailableUserAgreementsForUser

**Communication type:** Synchronously

This interface returns a list of user agreements available for the particular user, including the info whether he signed the user agreement. If the user' address country is not supported by MBconnect, or if no user agreements are available in the MBconnect country where the user lives in, then an empty list is returned.

Internally, the AF\_GetAvailableUserAgreementsForUser is called to retrieve the user agreements.

### 9.3.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
userID	Mand.	String		User.userID		The user ID of the customer. The ID is necessary to derive the information, which user agreements the user has signed.
Locale	Mand.	String	5	Used as input to determine user agreement name.	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the user agreement name.

---

Table 284: IF\_SEO\_GetAvailableUserAgreementsForUser Input

### 9.3.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of user agreements</b>						
uaID	Mand.	String		Document.documentID	Examples: "Basic Services"	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.
uaName	Mand.	String	-	UserAgreement.Name	Example: "User Agreement – Basic Services"	The name of the user agreement.
uaVersion	Mand.	Int	10	Document.VersionID	Example: 1, 213123	The user agreement version that the customer signed or revoked.
uaAcceptanceStatus	Mand.	String	12	UserAgreementConsent.AcceptanceStatus (the model deviates slightly from what possible values are returned in this IF. Only relevant states are "accepted" and "not accepted". Other states are derived to the former two.)	"Accepted", "Not Accepted"	The acceptance status of the user agreement for the specific user.
uaAcceptanceLocale	Opt.	String	5	UserAgreementConsent.uaAcceptanceLocale	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the signed user agreement.

Table 285: IF\_SEO\_GetAvailableUserAgreementsForUser Output

### 9.3.1.3 Exceptions

None.

## 9.3.2 IF\_SOE\_GetAvailableUserAgreementsForVehicle

**Communication type:** Synchronously

This interface returns a list of all user agreements for a particular vehicle that are needed to use all services offered by this vehicle, along with the information whether the user signed the user agreement. If the user' address country is not supported by MBconnect, or if no user agreements are available in the country where the user lives in, then an empty list is returned.

Internally, the AF\_GetAvailableUserAgreementsForVehicle is called to retrieve the user agreements.

### 9.3.2.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
userID	Mand.	String		User.userID		The user ID of the customer. The ID is necessary to derive the information, which user agreements the user has signed.
Locale	Mand.	String	5	Used as input to determine user agreement name.	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the user agreement name.
ModelSeries	Mand. If FIN not available	String	3	ModelSeries.ModelSeriesID	Example: "205"	The model series.
baumuster	Mand. If finOrVin not given	Baumuster DT		SalesType.baumuster	Example: "2050041"	The baumuster as part of the sales type.
NST	Opt.	NstDT		SalesType.Nst	Example: "CH1"	The NST code (for passenger cars, i.e. product group 'P') of the sales type.
ModelYearCode	Mand. If FIN not available	String	5	YearCodeCombination.ModelYearCode	Example: "804"	The code of the model year. <u>Mapping:</u> Add the YearCodeCombination.ModelYearCode to the set of equipments.
ChangeYear Code	Opt.	String	5	YearCodeCombination.ChangeYearCode	Example: "054"	The code of the change year.
FIN	Opt.	String	17	Vehicle.fin	Example: WDD1690071 J236589	The vehicle identification number is vehicle specific and is required to determine the vehicle.
<b>List of codes (optional)</b>						
EquipCode	Opt.	String	5	Equipment.Code	Example: "809", "820"	The equipment code of the vehicle.

Table 286: IF\_SOE\_GetAvailableUserAgreementsForVehicle Input

### 9.3.2.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of user agreements						
uaid	Mand.	String		Document.documentID	Examples: "Basic Services"	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
uaName	Mand.	String	-	UserAgreement.Name	Example: "User Agreement – Basic Services"	The name of the user agreement.
uaVersion	Mand.	Int	10	Document.VersionID	Example: 1, 213123	The user agreement version that the customer signed or revoked.
uaAcceptanceStatus	Mand.	String	12	UserAgreementConsent.AcceptanceStatus (the model deviates slightly from what possible values are returned in this IF. Only relevant states are "accepted" and "not accepted". Other states are derived to the former two.)	"Accepted", "Not Accepted"	The acceptance status of the user agreement for the specific user.
uaAcceptanceLocale	Opt.	String	5	UserAgreementConsent.uaAcceptanceLocale	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the signed user agreement.

Table 287: IF\_SOE\_GetAvailableUserAgreementsForVehicle Output

### 9.3.2.3 Exceptions

Code	Message
VEHPRO_005	The given vehicle cannot be found on UVS
VEHPRO_006	Error message if things go wrong that the calling system/AF cannot influence (= wrong configuration of SOE or technical errors)

Table 288: IF\_SOE\_GetAvailableUserAgreementsForVehicle exceptions

### 9.3.3 IF\_SOE\_GetContractData

**Communication type:** Synchronously

This interface returns a list of contractual information regarding a certain user. This includes information about the user agreements and services.

Internally, the AF\_GetContractData is called to retrieve the contract data.

*Remark: Each MBconnect service existing in the user's address country is returned by this interface.*

### 9.3.3.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
userID	Mand.	String		UserProxy.userID		The user ID of the customer.
Locale	Mand.	String	5		Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country).

---

Table 289: IF\_SOE\_GetContractData Input

### 9.3.3.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of user agreements (0..*)						
uaID	Mand.	String		Document.documentID	Examples: 1, 12356, 1234567890	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.
uaVersion	Mand.	Int		Document.VersionID	Example: 1, 213123	The user agreement version that the customer signed or revoked.
uaAcceptanceStatus	Mand.	Boolean		UserAgreementConsent.AcceptanceStatus mapped to a Boolean value	"true" (for accepted) and "false" (for not accepted)	The acceptance status of the user agreement for the specific user.
uaSignedDate	Opt.	Date		UserAgreementConsent.changeDate	Example: "01.01.2015"	Specifies when the user agreement was signed by the customer. This does not mean that the user agreement is still signed. Please refer to uaStatus for that. In case that the uaAcceptanceStatus is "not accepted", it is not recommended to use this date. If the UserAcceptance was never signed, this field will be omitted.
uaAcceptanceLocale	Opt.	String	5	UserAgreementConsent.uaAcceptanceLocale	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the signed user agreement.
List of services (0..*)						
ServiceID	Mand.	Int		Service.ServiceID	Examples: 1, 12356, 1234567890	The service ID identifies the specific service.

Table 290: IF\_SOE\_GetContractData Output

### 9.3.3.3 Exceptions

None.

## 9.3.4 IF\_SOE\_GetLegalDocuments

**Communication type:** Synchronously

This interface returns a list of legal documents (user agreements + terms and conditions). If the request leads to the generation of several documents, the documents can be requested as document bundle i.e. all documents are merged into one resulting doc-

ument. If the input parameter “RequestDocumentBundle” is set to true, then only distinct versions of the terms and conditions will be appended at the end of the generated document.

If the country of the user is not supported by MBconnect, an exception will be thrown. Internally, the AF\_GetLegalDocuments is called to retrieve the documents.

### 9.3.4.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
User-Id	Opt.	String		User.userID		The user ID of the customer. The ID is necessary to retrieve customer information (e.g name) that is printed onto the documents.
Locale	Mand.	String	5	Used as input parameter to determine document language	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies which language the document is supposed to be created in.
Outlet City	Opt.	String	-	-	Example: “Stuttgart”	The city of the outlet, at which the customer signs the user agreements.
Document Usage	Mand.	String		DocumentUsageEnum	Example: “FOR_SIGNATURE”, “FOR_DISP_LAY”	The usage specifies the purpose of the document, for instance if the document is supposed to be signed or just used to look/display the document as an information.
RequestDocumentBundle	Mand.	Boolean		-	true, false	RequestDocumentBundle determines whether the requested documents are returned separately (false) or merged as one PDF (true).
List of user agreement IDs						
User Agreement ID	Mand.	String	50	Document.documentID	Example: “UserAgreement-BsicServices”	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.

Table 291: IF\_GetLegalDocuments Input

### 9.3.4.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of requested legal documents (user agreements incl. the terms and conditions + version)						
Document	Mand.	Binary PDF	-	Binary PDF	-	The requested legal document in PDF format. It contains the user agreement and the terms and conditions as one binary PDF. In case that the input parameter “RequestDocumentBundle” is TRUE, then only distinct versions of the terms and conditions will be appended at the end of the generated

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						document.
List of user agreement (id + version)						
User Agreement ID	Mand.	String	50	Document.documentID	Examples: "UserAgreement-BsicServices"	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.
User Agreement Version	Mand.	int	10	VersionedDocument.versionID	Examples: 1, 12356, 1234567890	The user agreement version that the binary PDF corresponds to.

Table 292: IF\_GetLegalDocuments Output

### 9.3.4.3 Exceptions

Message Id	Fault Title	Fault Message
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
GETDOC_003	The requested User Agreement '{0}' does not exist.	The requested user agreement ID does not exist.

Table 293: Table of exceptions

## 9.3.5 IF\_SOE\_GetRequiredUserAgreementForService

**Communication type:** Synchronously

This interface returns the user agreement that needs to be signed before the given service can be used by the customer.

Internally, the AF\_GetRequiredUserAgreementForService is called to retrieve the user agreement.

### 9.3.5.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
userID	Opt.	String		User.userID		The user ID of the customer. The ID is necessary to derive the information, which user agreements the user has signed.
User	Opt.	UserType		UserType		The user profile of the customer.
Locale	Mand.	String	5	Used as input to determine user agreement name.	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the user agreement name.
ServiceID	Mand.	Int	10	Service.ServiceID	Examples: 1, 12356, 1234567890	The service ID identifies the specific service.

Table 294: IF\_SOE\_GetRequiredUserAgreementForService Input

### 9.3.5.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
uaID	Mand.	String		Document.documentID	Examples: "Basic Ser-	The user agreement ID identifies the specific doc-

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
					vices”	ument (e.g. user agreement for basic services) that is supposed to be returned.
uaName	Mand.	String	-	UserAgreement.Name	Example: “User Agreement – Basic Services”	The name of the user agreement.

Table 295: IF\_SOE\_GetRequiredUserAgreementForService Output

### 9.3.5.3 Exceptions

Code	Message
SERMAN_008	If such a service does not exist, a functional error SERMAN_008 is returned.
CONMAN_008	If multiple User Agreements exist, a functional error CONMAN_008 is returned.

Table 296: Exceptions

## 9.3.6 IF\_SOE\_GetServicesCoveredByUserAgreement

**Communication type:** Synchronously

This interface returns the services that belong to a given user agreement.

If the userID is not given, the services are fetched for a potentially new user. This means the latest, valid version of the user agreement will be considered.

If the userID is given, the services for the signed user agreement will be considered. In case the user did not sign the user agreement, the services are also fetched like for a potentially new user.

Internally, the AF\_GetServicesCoveredByUserAgreement is called to retrieve the user agreement.

### 9.3.6.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
userID	Opt.	String		User.userID		The user ID of the customer. The ID is necessary to derive the information, which user agreements the user has signed.
Locale	Mand.	String	5	Used as input to determine user agreement name.	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the user agreement name.
ualID	Mand.	String		Document.documentID	Examples: “Basic Services”	The user agreement ID identifies the specific document (e.g. user agreement for basic services) that is supposed to be returned.

Table 297: IF\_SOE\_GetServicesCoveredByUserAgreement Input

### 9.3.6.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of services</b>						
ServiceID	Mand.	Int	10	Service.ServiceID	Examples: 1, 12356, 1234567890	The service ID identifies the specific service.
ServiceName	Mand.	String	-	Service.Name	Example: “Remote Door Lock/Unlock ”	The name of the Service

Table 298: IF\_SOE\_GetServicesCoveredByUserAgreement Output

### 9.3.6.3 Exceptions

Code	Message
GETDOC_003	If the user Agreement ID does not exist, the error GETDOC_003 is returned

Table 299: IF\_SOE\_GetServicesCoveredByUserAgreement exceptions

## 9.4 Internal View - Offered Interfaces

### 9.4.1 IIF\_GetAllUserAgreementConsentsAndAssociations

This internal interface is implemented by the application function AF\_GetAllUserAgreementConsentsAndAssociations.

### 9.4.2 IIF\_DeleteUserAgreementConsents

See AF\_DeleteUserAgreementConsents (→ section 9.5.2).

### 9.4.3 IIF\_GetAllUserAgreementsWithStateForUser

See AF\_GetAllUserAgreementsWithStateForUser (→ section 9.5.3).

### 9.4.4 IIF\_GetRequiredUserAgreementForService

See AF\_GetRequiredUserAgreementForService (→ section 9.5.8).

### 9.4.5 IIF\_GetStateOfUserAgreementForServices

See AF\_GetStateOfUserAgreementForServices (→ section 9.5.10).

### 9.4.6 IIF.GetUserAgreementConsents

See AF.GetUserAgreementConsents(→ section 9.5.11).

---

## **9.4.7 IIF\_SetUserAgreementConsent**

See AF\_SetUserAgreementConsent (→ section 9.5.15).

## **9.5 Implementation**

### **9.5.1 AF\_GetAllUserAgreementConsentsAndAssociations**

This application function returns all user agreement consents and the associated user proxy, user agreement, and if available, the associated user agreement batch meta data.

#### **9.5.1.1 Sequence Description**

##### **Step 1: Determination of user agreements**

Select all user agreement consents. For each user agreement consent, get the associated user proxy, user agreement, and if available, the associated user agreement consent batch meta data.

##### **Step 2: Filter the user agreement acceptance status**

Filter the selected user agreement consents regarding the optional input parameter “AcceptanceState”. If no “Acceptance Status” is given as input, return all user agreement consents determined in step 1.

#### **9.5.1.2 Input**

Name	Type / Length / BOM	Description
AcceptanceState	UserAgreementAcceptanceStatus Enum	Optional

Table 300: AF\_GetAllUserAgreementConsentsAndAssociations input

#### **9.5.1.3 Output**

Name	Type / Length / BOM	Description
List of user agreement consents and associations		
UserAgreementConsent	UserAgreementConsent	The user agreement consent
UserProxy	UserProxy	The associated user proxy
UserAgreement	UserAgreement	The associated user agreement
UserAgreementConsentBatchMetaData	UserAgreementConsentBatchMetaData	If available, the associated user agreement consent batch meta data

Table 301: AF\_GetAllUserAgreementConsentsAndAssociations output

#### **9.5.1.4 Exceptions**

None.

## **9.5.2 AF\_DeleteUserAgreementConsents**

### **9.5.2.1 Sequence Description**

Step 1: Delete all instances of that <UserAgreementConsent> which are related to the user and the given user agreements.

---

Step 2: Delete all existing <UserAgreementConsentBatchMetadata> associated with the deleted user user agreement consents.

### 9.5.2.2 Input

Name	Type / Length / BOM	Description
userId	String	
User agreements	List of UserAgreement	

Table 302: AF\_DeleteUserAgreementConsents input

### 9.5.2.3 Output

Name	Type / Length / BOM	Description
UserAgreementConsents	List of UserAgreement Consent	

Table 303: AF\_DeleteUserAgreementConsents output

### 9.5.2.4 Exceptions

None.

## 9.5.3 AF\_GetAllUserAgreementsWithStateForUser

Internal application function to retrieve a list of all user agreements in the correct version including the agreement state and the locale the user accepted the agreement in.

### 9.5.3.1 Sequence Description

#### **Step 1: Determination of User Agreements**

Call IIF\_GetUserAgreementsByDate to get all available user agreements, with the input:

- Smallest Java Date value as the Valid From Date
- countryCode

#### **Step 2: Determination of User Agreement Acceptance State and Version**

Afterwards perform a check against <UserAgreementConsent> to retrieve the acceptance state for the given user ID and the Document ID of the user agreement retrieved in step 1

- If an entry exists and the acceptance state is ACCEPTED the productive form of the user agreement, the user agreement state <TRUE> and its signed version are remembered.
- If the user has rejected the user agreement (<UserAgreementConsent.acceptanceStatus> is REVOKED) or if there is no <UserAgreementConsent> corresponding the given user id and user agreement then productive form of the user agreement, the user agreement state <FALSE> and its highest version are remembered.

#### **Step 3: Return the User Agreements**

Return all determined user agreements with its acceptance state and its retrieved version.

### 9.5.3.2 Input

Name	Type / Length / BOM	Description
userId	String	Mandatory: User ID of the customer whose contract relations and accepted user agreements shall be deleted.
countryCode	String	Mandatory: Country for which all available user agreements need to be determined

Table 304: AF\_GetAllUserAgreementsWithStateForUser input

### 9.5.3.3 Output

Name	Type / Length / BOM	Description
<b>UserAgreements: A list of all user agreements. Each element has the following entries:</b>		
ID	Integer	ID of the User Agreement
Version	Integer	Version of the User Agreement
Name	String	Name / Description of the User Agreement
AcceptanceState	Boolean	TRUE if user agreement is accepted, FALSE if user agreement is revoked or if there is no consents corresponding this user agreement and user
ChangeDate	Date	Specifies when the user agreement was signed by the customer. This does not mean that the user agreement is still signed. Please refer to uaStatus for that. In case that the uaAcceptanceStatus is "not accepted", it is not recommended to use this date. If the UserAcceptance was never signed, this field will be omitted.
uaAcceptanceLocale	Locale	Locale (either a language or a language in combination with a country). Specifies the language of the signed user agreement.

Table 305: AF\_GetAllUserAgreementsWithStateForUser output

### 9.5.3.4 Exceptions

None.

## 9.5.4 AF\_GetAvailableUserAgreementsForUser

The service evaluates all user agreements which are related to a user, but independent of vehicles. Checks whether the user is located in a valid MBConnect country.

### 9.5.4.1 Sequence Description

#### **Step 1: Retrieve Customer Data**

Call the internal interface IIF\_GetOrCreateOrUpdateUserProxy with the given userId as input parameter to retrieve the UserProxy for the user.

#### **Step 2: Check MBconnect Status of Customer's Country**

Call the internal interface IIF\_IsMBconnectCountry with <UserProxy.custAddressCountry> as country to check if the user's address country is supported by MBconnect or not. If the result is true, continue. If the result is false, return with an empty list.

#### **Step 3: Determination of a User Agreement list**

---

The AF\_DeleteUserAgreementConsents is called with the input parameters <UserProxy.userID> and <UserProxy.addressCountry> to get the list of all User Agreements for a user.

#### 9.5.4.2 Input

See external Interface “IF\_SOE\_GetAvailableUserAgreementsForUser”.

#### 9.5.4.3 Output

See external Interface “IF\_SOE\_GetAvailableUserAgreementsForUser”.

#### 9.5.4.4 Exceptions

See external Interface “IF\_SOE\_GetAvailableUserAgreementsForUser”.

### 9.5.5 AF\_GetAvailableUserAgreementsForVehicle

Determines all user agreements which are associated (over at least one Service) to a given Vehicle. Checks whether the user is located in a valid MBConnect country.

#### 9.5.5.1 General Description

See external interface **Error! Reference source not found..**

#### 9.5.5.2 Sequence Description

##### **Step 1: Retrieve Customer Data**

Call the internal interface IIF\_GetOrCreateOrUpdateUserProxy with the given UserId as input parameter to retrieve the UserProxy for the user.

##### **Step 2: Check MBconnect Status of Customer's Country**

Call the internal interface IIF\_IsMBconnectCountry with <UserProxy.custAddressCountry> as country to check if the user's address country is supported by MBconnect or not. If the result is true, continue. If the result is false, return with an empty list.

##### **Step 3: Determination of Services**

If the FIN is given **Error! Reference source not found.** is used (with <UserProxy.custAddressCountry> and the business area “B2C” as input parameter) to retrieve a service list.

If a configuration is given, then IIF\_GetAvailableServicesByConfiguration (with <UserProxy.custAddressCountry> and the business area “B2C” as input parameter) is used to retrieve a service list.

##### **Step 4: Determination of User Agreements**

The AF\_GetAllUserAgreementsWithStateForUser with the input parameters <UserProxy.userID> and <UserProxy.custAddressCountry> is called to get the list of all User Agreements for a user.

For each User Agreement, it is checked whether a correlated Service exists. If not, then the User Agreement is ignored. If yes, the UserAgreement is incorporated in the output. If that AF quits with an error, quit with that error too.

---

### **9.5.5.3 Input**

See external interface IF\_SOE\_GetAvailableUserAgreementsForVehicle (→ section 9.3.2).

### **9.5.5.4 Output**

See external interface IF\_SOE\_GetAvailableUserAgreementsForVehicle (→ section 9.3.2).

### **9.5.5.5 Exceptions**

See external interface IF\_SOE\_GetAvailableUserAgreementsForVehicle (→ section 9.3.2).

## **9.5.6 AF\_GetContractData**

### **9.5.6.1 General Description**

See external interface **Error! Reference source not found..**

### **9.5.6.2 Sequence Description**

#### ***Step 1: Determine the address country of the user***

Call the internal interface IIF\_GetOrCreateOrUpdateUserProxy with the given UserId to retrieve the UserProxy for the user.

#### ***Step 2: Determine all available user agreements with their acceptance status***

Call the AF\_GetAllUserAgreementsWithStateForUser with the UserProxy.UserId and the UserProxy.custAddressCountry as input parameters.

#### ***Step 3: Determine all available services***

Call the internal interface IIF\_GetAllServices with with onlyEnabledServices="True", BusinessArea="B2C" and the UserProxy.custAddressCountry as input parameters.

Return the results from step 2 and step 3 as output.

### **9.5.6.3 Input**

See external Interface "IF\_SOE\_GetContractData".

### **9.5.6.4 Output**

See external Interface "IF\_SOE\_GetContractData".

### **9.5.6.5 Exceptions**

See external Interface "IF\_SOE\_GetContractData".

## **9.5.7 AF\_GetLegalDocuments**

This AF retrieves and returns the appropriate legal documents (user agreements including the terms and conditions) for the specified user or retrieves and returns user independent legal documents, if no user is specified. This is only for creating legal documents that need to be signed (DocumentUsageEnum.FOR\_SIGNATURE) or displayed

---

for informational purposes (DocumentUsageEnum.FOR\_DISPLAY) only. This is not for creating legal documents that have been signed and sent out as a confirmation.

### **9.5.7.1 Sequence Description**

Start with Step 1 if the user ID is given. If no user ID is given, start with Step 4.

#### **Step 1: Retrieve Customer Data**

Call the internal interface IIF\_GetUserProfile with the given user ID to retrieve the customer data <UserType> with the <ADDRESS\_COUNTRY>.

#### **Step 2: Check MBconnect Status of Customer's Country**

Call IIF\_IsMBconnectCountry with < ADDRESS\_COUNTRY> as country to check if the user's address country is supported by MBconnect or not. If the result is true, continue. If the result is false, quit the AF with the exception ACCDAS\_005.

#### **Step 3: Determine Customer Classification**

A check is performed against <UserAgreementConsent> to retrieve the acceptance state for the given user ID and the user agreement ID. If an entry exists for which <UserAgreementConsent.agreementStatus> is "ACCEPTED", then the user is considered an "existing customer" for the particular user agreement, otherwise he is considered a "new customer". Step 3 is repeated for each given user agreement ID.

#### **Step 4: Determine User Agreement Version**

If the customer is classified as a "new customer" or no user ID is given for a given user agreement ID, retrieve the highest version of <UserAgreement> for which the <UserAgreement.validFrom> is equal or smaller than today and <UserAgreement.enabled> is TRUE.

If the customer is classified as an "existing customer" for a given user agreement ID, retrieve the associated <UserAgreement>.

#### **Step 5: Determine DocumentTriggerType**

For each determined <UserAgreement> from step 4, the correct DocumentTriggerType needs to be determined. This is done with the help of the given <DocumentUsageType>.

If the given <DocumentUsageType> is:

"FOR\_SIGNATURE", then the DocumentTriggerType is USER AGREEMENT FOR SIGNING

"FOR\_DISPLAY", then the DocumentTriggerType is USER AGREEMENT LOOKUP

#### **Step 6: Generate Documents**

For each determined <UserAgreement> from step 3, call AF\_GeneratePDF with the parameters:

The <DocumentXmlDefinition> of the <UserAgreement>

The given locale

The <UserType> if the user ID is given

org city

The determined DocumentTriggerType as the condition to generate the requested user agreements.

---

#### **(Optional Step 7: Merge Documents)**

If requested, the generated documents can be returned as one merged document (`requestDocumentBundle == true`). In that particular case, `AF_MergePDFs` is called with the given documents from Step 5 as a list of binary PDFs (listed in their order of appearance). The result from `AF_MergePDFs` (one merged document) will be returned.

#### **9.5.7.2 Input**

See external interface “`IF_SOE_GetLegalDocuments`” (see 9.3.4 `IF_SOE_GetLegalDocuments`).

#### **9.5.7.3 Output**

See external interface “`IF_SOE_GetLegalDocuments`” (see 9.3.4 `IF_SOE_GetLegalDocuments`).

#### **9.5.7.4 Exceptions**

See external interface “`IF_SOE_GetLegalDocuments`” (see 9.3.4 `IF_SOE_GetLegalDocuments`).

### **9.5.8 AF\_GetRequiredUserAgreementForService**

The service returns the current User Agreement, which is responsible for a given MBconnect service.

#### **9.5.8.1 Sequence Description**

##### **Step 1: Retrieve Customer Data**

If the `userId` is given, call the internal interface `IIF_GetOrCreateOrUpdateUserProxy` with the given `userId` as input parameter to retrieve the `UserProxy` for the user.

##### **Step 2: Determination of “User Agreements”**

The `AF_GetAllUserAgreementsWithStateForUser` is called with the parameters `<UserProxy.UserID, UserProxy.CustAddressCountry>`, to get the list of all available user agreements.

If the `userID` and the user profile is not given, the latest enabled version of the productive form of the user agreements will be determined as follows: the entries with the highest version are retrieved for which `<UserAgreement.enabled>` is TRUE and has a productive form. The current date must be after or equal to the `<UserAgreement.validFrom>`.

##### **Step 3: Determination of correct User Agreement**

With the given service ID, the corresponding User Agreements are selected from the DB and intersected with the list of User Agreements (ID and version):

- If such a Service does not exist, a functional error SERMAN\_008 is returned.
- If multiple User Agreements exist, a functional error CONMAN\_008 is returned.
- If no User Agreement exists, return nothing.

---

Otherwise, the User Agreement is returned.

### **9.5.8.2 Input**

See external Interface “IF\_SOE\_GetRequiredUserAgreementForService”.

### **9.5.8.3 Output**

See external Interface “IF\_SOE\_GetRequiredUserAgreementForService”.

### **9.5.8.4 Exceptions**

See external Interface “IF\_SOE\_GetRequiredUserAgreementForService”.

## **9.5.9 AF\_GetServicesCoveredByUserAgreement**

The service returns all MBconnect services, which are covered by a single user agreement. The optional user ID is used to determine the correct user agreement.

If the userID is not given, the services are fetched for a potentially new user. This means the latest and valid version of the user agreement will be considered.

If the userID is given, the services for the signed user agreement will be considered. In case the user did not sign the user agreement, the services are also fetched like for a potentially new user.

### **9.5.9.1 Sequence Description**

#### ***Step 1: Determination of “User Agreements”***

If the userID is given call the internal interface IIF\_GetOrCreateOrUpdateUserProxy with the given UserId as input parameter to retrieve the UserProxy for the user.

The AF\_GetAllUserAgreementsWithStateForUser is called with the parameters <UserProxy.userID, UserProxy.addressCountry>, to get the list of all available user agreements.

If the given user agreement does not exist in the list, a functional error GETDOC\_003 is returned.

If the userID is not given, the latest enabled version of the productive form of the user agreement will be determined as follows: the entry with the highest version is retrieved for which <UserAgreement.enabled> is TRUE and has a productive form. The current date must be after or equal to the <UserAgreement.validFrom>.

#### ***Step 2: Determination of Services***

With this given user agreement, IIF\_GetServicesByUserAgreements from Documents is called to receive the service list.

### **9.5.9.2 Input**

See external interface IF\_SOE\_GetServicesCoveredByUserAgreement (→ section 9.3.6).

---

### **9.5.9.3 Output**

See external interface IF\_SOE\_GetServicesCoveredByUserAgreement (→ section 9.3.6).

### **9.5.9.4 Exceptions**

See external interface IF\_SOE\_GetServicesCoveredByUserAgreement (→ section 9.3.6).

## **9.5.10 AF\_GetStateOfUserAgreementForServices**

For each service in the given list of services, this application function determines the current acceptance status of the matching user agreement.

### **9.5.10.1 Sequence Description**

1. For all existing user agreement consents (entity UserAgreementConsent) which have the status accepted (attribute UserAgreementConsent.AcceptanceStatus == ACCEPTED) and relate to the given customer (userId) find out which MBconnect services they cover. To do so, call IIF\_GetServicesByUserAgreements with the respective versioned user agreements as parameter (UserAgreementConsent.UserAgreement).
2. Match the services, determined in Step 1 against the services that are given as input parameter:
  - a. The services that are covered by any user agreement consents which have the status accepted (as determined in step 1) and given as input parameter will be returned with the UserAgreementStatus “ACCEPTED”.
  - b. The services that are not covered by any consent with status accepted (not returned by IIF\_GetServicesByUserAgreements in step 1) but given as input parameter, check if a user agreement is needed for that service. Call AF\_GetRequiredUserAgreementForService with these services, the user ID and any valid locale (only the ID of the user agreement is relevant).

If that AF returns with a user agreement, this service will be returned with the UserAgreementStatus “REQUIRED”. If that AF returns with no user agreement (no user agreement is needed for that service), this service will be returned with the UserAgreementStatus “INAPPLICABLE”.

### **9.5.10.2 Input**

Name	Type / Length / BOM	Description
User	UserType	User profile of the customer.
ServiceList	List of <Service.ServiceID>	The services for which the acceptance status of the matching user agreement needs to be returned.

### 9.5.10.3 Output

Name	Type / Length / BOM	Description
List of services (1...*)		
ServiceId	Service.ServiceId	The ID of the service.
UserAgreementStatus	UserAgreementConsent.userAgreementAcceptanceStatusEnum	INAPPLICABLE: no user agreement needed for service REQUIRED: user agreement needed, but not yet accepted ACCEPTED: user agreement has been accepted

### 9.5.10.4 Exceptions

None.

## 9.5.11 AF\_GetUserAgreementConsents

This AF gets all available user agreements related to the given user. If desired acceptance state of user agreement is also given as input parameter, then this state of user agreements will be taken into consideration by this AF too. Get all available user agreement consents determined by the user and the optional acceptance status of the corresponding user agreement.

### 9.5.11.1 Sequence Description

#### *Step 1: Determination of User Agreement consents*

Select all instances of user agreement consent, which are related to the given user id (UserAgreementConsent.UserID = UserId).

#### *Step 2: Filter the user agreement acceptance status*

Filter all the selected user agreement consents regarding the optional input parameter "AcceptanceState". If no "Acceptance Status" is given as input, return all user agreement consents.

### 9.5.11.2 Input

Name	Type / Length / BOM	Description
UserId	String	
AcceptanceState	UserAgreementAcceptanceStatus Enum	optional

Table 306: AF\_GetUserAgreementConsents input

### 9.5.11.3 Output

Name	Type / Length / BOM	Description
UserAgreementConsents	List of UserAgreement Consent	

Table 307: AF\_GetUserAgreementConsents output

---

#### **9.5.11.4 Exceptions**

None.

### **9.5.12 AF\_PrintApproval**

This AF marks in the UserAgreementConsentBatchMetadata these all documents that should be printed.

#### **9.5.12.1 Sequence Description**

For all <UserAgreementConsentBatchMetada.isInformed> = “FALSE” and <UserAgreementConsentBatchMetada.isApproved>=“FALSE” set <UserAgreementConsentBatchMetada.isApproved>=“TRUE”.

#### **9.5.12.2 Input**

None.

#### **9.5.12.3 Output**

None.

#### **9.5.12.4 Exceptions**

None.

### **9.5.13 AF\_SelectNewLegalDocumentsBatch**

This AF checks all user agreements and their “inform Date by Mail” date to collect the customers, who have to be informed about the availability of new legal documents. All these collected customers will be marked.

Additionally, the functional support SOE will be informed by email, including the request of print approval, about the number of emails which have to be sent to inform the customer about changes of user agreements.

#### **9.5.13.1 Sequence Description**

##### ***Step 1: Retrieve User Agreements to inform of***

Call IIF\_GetUserAgreementsByDate to retrieve the upcoming enabled <UserAgreement> productive form that the user needs to be informed of, along with the previous <UserAgreement> if available with the input:

- Date of the last successful batch run as the informDateByMail or if the batch was never run, use the smallest value of Date

##### ***Step 2: Determine affected users***

Go through each previous <UserAgreement> from step 2a and check if there is a related <UserAgreementConsent> that has the agreement status “ACCEPTED” and also there is still no record for the entity <UserAgreementConsentBatchMetadata>.

Attach a new entity of type <UserAgreementConsentBatchMetadata> to each <UserAgreementConsent> satisfying this condition. Set its attributes

<UserAgreementConsentBatchMetadata.isInformed> and  
<UserAgreementConsentBatchMetadata.isApproved> to FALSE.

### **Step 3: Send email to functional support SOE**

If records of the entity <UserAgreementConsentBatchMetadata> exist with the attributes <UserAgreementConsentBatchMetadata.isInformed>="FALSE" and <UserAgreementConsentBatchMetadata.isApproved>="FALSE", then send an English written email to the functional support SOE. In this email should be the number of <UserAgreementConsentBatchMetadata.isInformed>="FALSE" and <UserAgreementConsentBatchMetadata.isApproved>="FALSE", i.e. the number of letters and emails that needed to inform the customer, and the request for print approval. The functional support SOE must be contactable by mailing list.

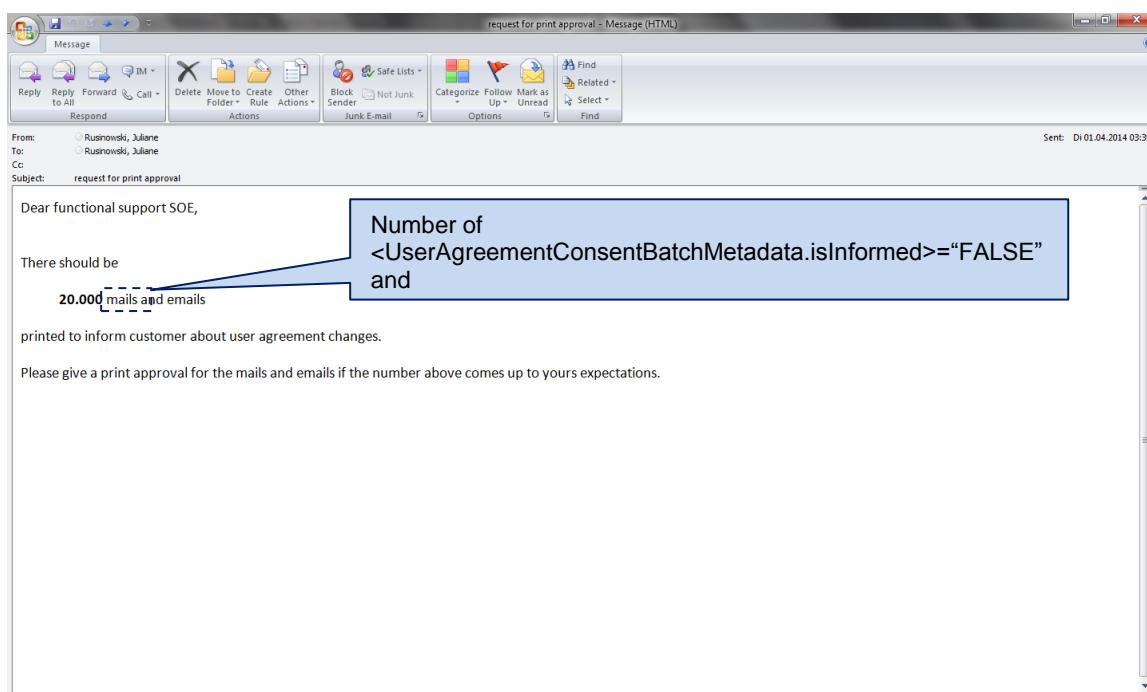


Figure 69: Email for functional support SOE

#### **9.5.13.2 Input**

None.

#### **9.5.13.3 Output**

None.

#### **9.5.13.4 Exceptions**

None.

### **9.5.14 AF\_SendNewLegalDocumentsBatch**

This AF checks UserAgreementConsentBatchMetadata for user agreements that were changed and for these the customer has to be informed.

---

Additionally with the UserAgreementConsentBatchMetadata it is possible to find out the customers who weren't informed by the batch after the batch is stopped and not finish.

#### **9.5.14.1 Sequence Description**

##### **Iterate over all <UserAgreementConsentBatchMetaData>**

###### Step 1: Retrieve users and changed user agreements

- If <UserAgreementConsentBatchMetadata.isInformed> = FALSE and <UserAgreementConsentBatchMetadata.isApproved> = TRUE
  - Retrieve the <UserAgreement> and the user which are linked to the <UserAgreementConsentBatchMetadata> via the corresponding <UserAgreementConsent>
  - Retrieve the language (locale) in which the user agreement was accepted (<UserAgreementConsent>.uaAcceptanceLocale)

###### Step 2: Group the affected user agreement consents by users

- For each affected user call IIF\_GetUserProfile to get the user's profile

###### Step 3: Inform the affected users of changed user agreements

- Call IIF\_InformCustomer for each affected user to inform about the upcoming user agreements with the following inputs:
  - USER AGREEMENT CHANGED as <DocumentTriggerEnum>
  - The user's profile to be informed
  - Omit vehicle configuration
  - Retrieved list of user agreement and locales as <UserAgreement, locale>

###### Step 4: Update batch meta data

- Set <UserAgreementConsentBatchMetadata.isInformed> = TRUE

#### **9.5.14.2 Input**

None.

#### **9.5.14.3 Output**

None.

#### **9.5.14.4 Exceptions**

None.

### **9.5.15 AF\_SetUserAgreementConsent**

This AF sets the state of consent for a user concerning a certain user agreement in the particular version. This happens either when a user agreement is signed or revoked by the user.

### 9.5.15.1 Sequence Description

Call AF.GetUserAgreementConsents (section 9.5.11)with the given userId to get all existing user agreement consents for the given user. Determine in the result list the consents associated with the documentId of the given user agreement. In case:

- if a <UserAgreementConsent> (corresponding the given userId and documentId) does not exist and the new uaAcceptanceStatus for the given UserAgreement is equal to “ACCEPTED” then create a new <UserAgreementConsent> and associate it with the <UserAgreement> and <UserProxy>, where:
  - <UserAgreementConsent.AcceptanceStatus> is the given user agreement acceptance status
  - <UserAgreementConsent.DateOfConsentChange> is the current date
  - <UserAgreementConsent.systemID> is the given systemID
  - <UserAgreementConsent.orgID> is the given orgID
  - <UserAgreementConsent.userID> is the given UserProxy.userID
  - <UserAgreementConsent.acceptanceLocale> is the given acceptanceLocale
- if a <UserAgreementConsent> (corresponding the given userId and documentId) exists, check the versionId of the document: If the existing entry has a version id that is:
  - (a) less than the version id provided as input parameter or
  - (b) equal and acceptance status has changed (where existing UserAgreementConsent.AcceptanceStatus != new uaAcceptanceStatus of user agreement provided as input parameter) then update the entry:
    - <UserAgreementConsent.AcceptanceStatus> is the given user agreement acceptance status.
    - <UserAgreementConsent.DateOfConsentChange> is the current date
    - <UserAgreementConsent.systemID> is the given systemID
    - <UserAgreementConsent.orgID> is the given orgID
    - <UserAgreementConsent.userID> is the given UserProxy.userID
    - <UserAgreementConsent.acceptanceLocale> is the given acceptanceLocale

If UserAgreementConsent.AcceptanceStatus is REVOKED, then delete all existing <UserAgreementConsentBatchMetadata>.

### 9.5.15.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of user agreements</b>						
userId	Mand.	String		Document.documentID	Examples: “Basic Services”	The user agreement ID identifies the specific user agreement for which the status is to be recorded.
uaAcceptanceStatus						
	Mand.	String	12	UserAgreementConsent.AcceptanceStatus (the model deviates slightly from	“Accepted”, “Not Accepted”	The acceptance status of the user agreement for the specific user.

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
				what possible values are returned in this IF. Only relevant states are “accepted” and “not accepted”. Other states are derived to the former two.)		
uaVersion	Mand.	int	10	Document.VersionID	Examples: 1, 12356, 1234567890	The user agreement version that the customer signed or revoked.
uaAcceptanceLocale	Mand.	String	5	UserAgreementConsent. uaAcceptanceLocale	Examples: de_DE, de_AT, fr_CH, de_CH, en	Locale (either a language or a language in combination with a country). Specifies the language of the signed user agreement.
<b>Metadata</b>						
SystemId	Man d.	String	20	Used as input parameter for AF_SetUserAgreementStateOfConsent	Examples: “My Mercedes”, “MBC POS”, “CPD”	The name of the system the action was originally triggered in (not necessarily the one which sent the message).
Orgid (RetailerId)	Opt.	String	20	Used as input parameter for AF_SetUserAgreementStateOfConsent (Market.GSSNOutletOutletID)	Example: “TR0364178”, “GS0008205”	ID of the retailer who triggered the action (if applicable). The organization ID of the retailer, which is recorded for safekeeping purposes to know where the consent was given.
UserId	Opt.	String	50	Used as input parameter for AF_SetUserAgreementStateOfConsent	Examples: „D0X00107“, „mmeier“, „vorna-me.nachname“	User ID of the user who originally triggered the action (if applicable). The ID of the retailer user, if the call came from POS. Else it's the user ID from the customer if the call came from MyMercedes.

Table 308: AF\_SetUserAgreementConsent input

### 9.5.15.3 Output

None.

### 9.5.15.4 Exceptions

None.

## 9.6 Batches

Batchname	Called Application Function	Description
Batch “Send new legal documents to Customer”	AF_SendNewLegalDocumentsBatch	Execution time: night time Execution frequency: daily
Batch “Select Customer, who has to be informed about new legal documents”	AF_SelectNewLegalDocumentsBatch	Execution time: night time Execution frequency: daily after the Batch “Send new legal documents to Customer” has ended.

## 9.7 Error Messages

None.



---

# **10 Component „Availabilities“**

## **10.1 Dialogs**

None.

## **10.2 External View - Offered Interfaces**

None.

## **10.3 External View - Consumed Interfaces**

None.

## **10.4 Internal View – Offered Interfaces**

### **10.4.1 IIF\_DeleteAvailabilitiesForUserAndVehicle**

Internally calls AF\_DeleteAvailabilitiesForUserAndVehicle (→ section 10.5.1).

### **10.4.2 IIF\_GetAvailabilitiesForUser**

Internally calls AF\_GetAvailabilitiesForUser (→ section 10.5.2).

### **10.4.3 IIF\_UpdateAvailabilitiesForUserAndVehicle**

Internally calls AF\_UpdateAvailabilitiesForUserAndVehicle (→ section 10.5.3).

### **10.4.4 IIF\_UpdateAvailabilitiesForBusinessPartnerAndVehicle**

Internally the AF\_UpdateAvailabilitiesForBusinessPartnerAndVehicle is called.

## **10.5 Implementation**

### **10.5.1 AF\_DeleteAvailabilitiesForUserAndVehicle**

This application functions deletes the service availabilities, which are determined through the optional input parameters.

#### **10.5.1.1 Sequence Description**

Delete all availabilities that reference the given input parameters (ServiceAvailability.user and/or ServiceAvailability.vehicle).

#### **10.5.1.2 Input**

Name	Type / Length / BOM	Description
UserId	String	optional
FIN	String	optional

Table 309: AF\_DeleteAvailabilitiesForUserAndVehicle input

### 10.5.1.3 Output

Name	Type / Length / BOM	Description
Availabilities	List	

Table 310: AF\_DeleteAvailabilitiesForUserAndVehicle output

### 10.5.1.4 Exceptions

None.

## 10.5.2 AF\_GetAvailabilitiesForUser

The application function returns the service availability entities for given user.

### 10.5.2.1 Sequence Description

Return all service availabilities referencing the userId given as input parameter (ServiceAvailability.user).

### 10.5.2.2 Input

Name	Type / Length / BOM	Description
UserId	String	optional

Table 311: AF\_GetAvailabilitiesForUser input

### 10.5.2.3 Output

Name	Type / Length / BOM	Description
Availabilities	List	

Table 312: AF\_GetAvailabilitiesForUser output

### 10.5.2.4 Exceptions

None.

## 10.5.3 AF\_UpdateAvailabilitiesForUserAndVehicle

The application function updates the service availability entities for given user, vehicle and list of services.

### 10.5.3.1 Sequence Description

#### **Step 1: Delete all availabilities for previous vehicles owner**

Delete all availabilities, which are referencing the given VehicleProxy, but not the given UserProxy.

#### **Step 2: Delete obsolete availabilities**

Delete all availabilities, which are referencing the given VehicleProxy and UserProxy, but not the given service ids.

#### **Step 3: Create new availabilities**

For all of the services from the input list, which are not referenced in the existing availabilities, create a new service availability based on the given list of services.

- Connect the service availabilities to the given UserProxy

- Connect the service availabilities to the given VehicleProxy

### 10.5.3.2 Input

Name	Type / Length / BOM	Description
UserProxy	UserProxy	The user proxy object of the user whose service availabilities will be updated
VehicleProxy	VehicleProxy	The vehicle proxy assigned to the given UserProxy and used to link availability entities to VehicleProxy
Services	List of Services	List of services available for user and his vehicle

Table 313: AF\_UpdateAvailabilitiesForUserAndVehicle input

### 10.5.3.3 Output

None.

### 10.5.3.4 Exceptions

None.

## 10.5.4 AF\_UpdateAvailabilitiesForBusinessPartnerAndVehicle

### 10.5.4.1 General Description

The application function updates the B2B service availabilities for the given business partner and vehicle.

#### Sequence Description

##### Step 1: Retrieve available services

Call the internal interface **Error! Reference source not found.** with the given VehicleConfiguration, the business area “B2B”, an empty address country and an empty consumer country as input parameters to retrieve the available B2B services.

##### Step 2: Delete availabilities for previous business partners

Delete all B2B service availabilities which are referencing the given vehicle, but not the given business partner, i.e. where the Availability.BusinessPartner does not equal the BusinessPartner from the input.

##### Step 3: Delete obsolete availabilities

Delete all B2B service availabilities which are referencing the given vehicle and business partner, but none of the B2B services from step 1.

##### Step 4: Create missing availabilities

For each B2B service from step 1 which is not referenced by any of the existing B2B service availabilities, create a new B2B service availability:

- Connect the ServiceAvailabilityB2B to the Vehicle
- Connect the ServiceAvailabilityB2B to the BusinessPartner
- Connect the ServiceAvailabilityB2B to the B2B service

#### Input

Name	Type / Length / BOM	Description
BusinessPartner	BusinessPartnerProxy	The proxy object for the business partner for whom to update the availabilities.

---

Vehicle	VehicleProxy	The proxy object for the vehicle for which to update the availabilities.
VehicleConfiguration	VehicleConfiguration	The configuration of the vehicle for which to update the availabilities.

Table 314: AF\_UpdateAvailabilitiesForBusinessPartnerAndVehicle input

### Output

None.

### Exceptions

None.

## 10.5.5 AF\_DeleteAvailabilitiesForBusinessPartnerAndVehicle

### 10.5.5.1 General Description

This application functions deletes the B2B service availabilities for the given business partner and vehicle.

### 10.5.5.2 Sequence Description

For the given business partner and vehicle delete the corresponding B2B service availabilities, i.e. all B2B service availabilities where the ServiceAvailabilityB2B.BusinessPartner.BusinessPartnerId matches the BusinessPartnerId from the input and where the ServiceAvailabilityB2B.Vehicle.Fin matches the Fin from the input.

### 10.5.5.3 Input

Name	Type / Length / BOM	Description
BusinessPartnerId	String / 20 /BusinessPartnerProxy.BusinessPartnerId	The id of the business partner for whom to update the availabilities.
Fin	String / 17 / VehicleProxy.Fin	The European identifier of the vehicle for which to update the availabilities.

Table 315: AF\_DeleteAvailabilitiesForBusinessPartnerAndVehicle input

### 10.5.5.4 Output

None.

### 10.5.5.5 Exceptions

None.

## 10.6 Batches

None.

## 10.7 Error Messages

None.

---

# **11 Component “Context Proxy”**

## **11.1 Dialogs**

None.

## **11.2 External View - Offered Interfaces**

None.

## **11.3 External View - Consumed Interfaces**

None.

## **11.4 Internal View – Offered Interfaces**

### **11.4.1 IIF\_GetVehicleProxiesAndAssociatedUserProxy**

This internal interface is implemented by the application function AF\_GetVehicleProxiesAndAssociatedUserForModelSeries.

### **11.4.2 IIF\_AssignUserProxyToVehicleProxy**

Internally, the AF\_AssignUserProxyToVehicleProxy (→ section 11.5.3) is called.

### **11.4.3 IIF\_DeleteUserProxy**

Internally, the AF\_DeleteUserProxy is called.

### **11.4.4 IIF\_DetachUserProxyFromVehicleProxy**

Internally, the AF\_DetachUserProxyFromVehicleProxy (→ section 11.5.5) is called.

### **11.4.5 IIF\_AssignBusinessPartnerProxyToVehicleProxy**

Internally, the **Error!** Reference source not found. is called.

### **11.4.6 IIF\_DetachBusinessPartnerProxyFromVehicleProxy**

Internally, the AF\_DetachBusinessPartnerProxyFromVehicleProxy is called.

### **11.4.7 IIF\_GetOrCreateOrUpdateUserProxy**

Internally, the AF\_GetOrCreateOrUpdateUserProxy (→ section 11.5.7) is called.

### **11.4.8 IIF\_GetVehicleProxy**

Internally, the AF\_GetVehicleProxy (→ section 11.5.8) is called.

---

## **11.4.9 IIF\_GetVehicleProxiesForUser**

Internally, the AF\_GetVehicleProxiesForUser (→ section 11.5.9) is called.

# **11.5 Implementation**

## **11.5.1 AF\_GetOrCreateBusinessPartnerProxy**

### **11.5.1.1 General Description**

This application function retrieves the proxy object for the given business partner. The proxy object will be created if it does not exist yet.

### **11.5.1.2 Sequence Description**

For the given BusinessPartnerId determine the corresponding BusinessPartnerProxy.

If there is no BusinessPartnerProxy that matches the given BusinessPartnerId, create a new BusinessPartnerProxy and set the BusinessPartnerProxy.BusinessPartnerId to the BusinessPartnerId from the input.

Return the BusinessPartnerProxy as output parameter.

### **11.5.1.3 Input**

Name	Type / Length / BOM	Description
BusinessPartnerId	String / 20 / BusinessPartnerProxy.BusinessPartnerId	The ID of the business partner, e.g. "MBC.DFS".

Table 316: AF\_GetOrCreateBusinessPartnerProxy input

### **11.5.1.4 Output**

Name	Type / Length / BOM	Description
BusinessPartner	BusinessPartnerProxy	The proxy object for the business partner.

Table 317: AF\_GetOrCreateBusinessPartnerProxy output

### **11.5.1.5 Exceptions**

None.

## **11.5.2 AF\_GetOrCreateVehicleProxy**

### **11.5.2.1 General Description**

This application function retrieves the proxy object for the given vehicle. The proxy object will be created if it does not exist yet.

### **11.5.2.2 Sequence Description**

For the given Fin determine the corresponding VehicleProxy.

If there is no VehicleProxy that matches the given Fin, create a new VehicleProxy and set the VehicleProxy.Fin to the Fin from the input.

---

Return the VehicleProxy as output parameter.

### 11.5.2.3 Input

Name	Type / Length / BOM	Description
Fin	String / 17 / VehicleProxy.Fin	The European vehicle identification number ("Fahrzeug-Identifizierungsnummer").

Table 318: AF\_GetOrCreateBusinessPartnerProxy input

### 11.5.2.4 Output

Name	Type / Length / BOM	Description
Vehicle	VehicleProxy	The proxy object for the vehicle.

Table 319: AF\_GetOrCreateBusinessPartnerProxy output

### 11.5.2.5 Exceptions

None.

## 11.5.3 AF\_AssignUserProxyToVehicleProxy

This application function assigns the given user to the given vehicle. If the vehicle is still assigned to another user, the user will be separated from the vehicle first. The proxy objects for the user and the vehicle will be created if they do not exist yet.

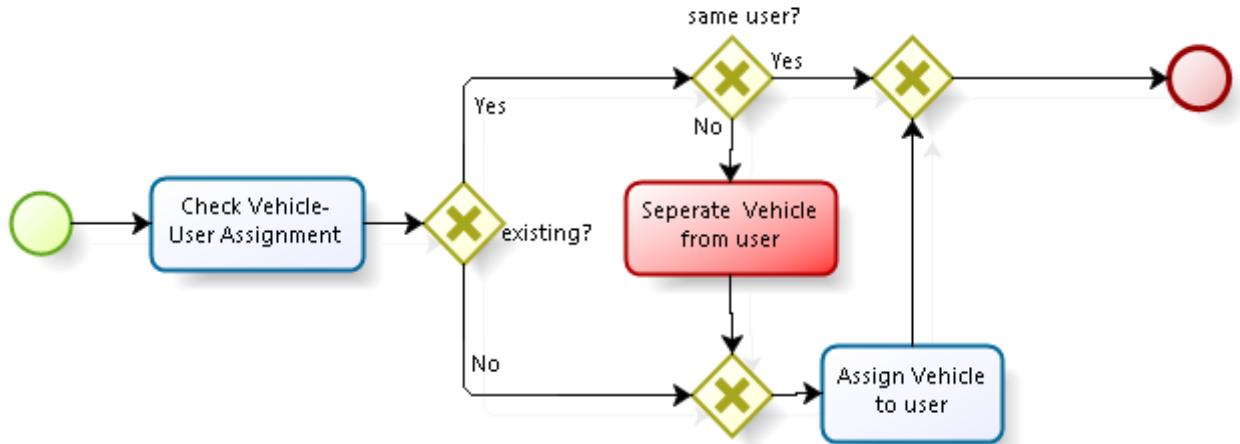
### 11.5.3.1 Sequence Description

Call the AF\_GetOrCreateOrUpdateUserProxy with the given UserId and AddressCountry as input parameters to retrieve or create the UserProxy.

Call the application function AF\_GetOrCreateVehicleProxy with the given Fin to retrieve or create the VehicleProxy.

Check the assignment of the vehicle to the user (see illustration below):

- a. If the vehicle is already assigned to the user from the input then skip this step.
- b. If the vehicle is still assigned to another user than the user from the input then:
  - Log a warning
  - Call AF\_DetachUserProxyFromVehicleProxy with the "old" user and FIN.
  - Connect the UserProxy for the user from the input to the VehicleProxy
- c. If the vehicle is not yet assigned to any user, connect the UserProxy for the user from the input with the VehicleProxy.



Return the determined UserProxy and VehicleProxy.

### 11.5.3.2 Input

Parameter Name	Type / Length / BOM	Possible Values	Description
UserId	String / 50 / UserProxy.UserId		The user id of the customer.
AddressCountry	String / 2	"DE", "GB"	The ISO code for the address country of the customer.
FIN	String / 17 / VehicleProxy.FIN		The FIN of the vehicle.

Table 320: AF\_AssignUserProxyToVehicleProxy input

### 11.5.3.3 Output

Name	Type / Length / BOM	Description
UserProxy	UserProxy	The proxy object for the customer.
VehicleProxy	VehicleProxy	The proxy object for the vehicle.

Table 321: AF\_AssignUserProxyToVehicleProxy output

### 11.5.3.4 Exceptions

None.

## 11.5.4 AF\_DeleteUserProxy

### 11.5.4.1 General Description

This application function deletes UserProxy of the given user.

### 11.5.4.2 Sequence Description

For the given userID, do following:

Step1: determine the corresponding UserProxy.

Step 2: Determine all VehicleProxies assigned to the given UserId.  
Call AF\_GetVehicleProxiesForUser and pass in the given UserId.

Step 3: Detach UserProxy from VehicleProxy

---

For each VehicleProxy determined in step 2 do following: call AF\_DetachUserProxyFromVehicleProxy and pass in the VehicleProxy and UserProxy (from step 1).

#### Step 4: Delete UserProxy.

##### **11.5.4.3 Input**

Name	Type / Length / BOM	Description
UserId (Mand.)	String / 50 / User.UserId	The user id of the customer.

Table 322: AF\_DeleteUserProxy input

##### **11.5.4.4 Output**

None.

##### **11.5.4.5 Exceptions**

None.

#### **11.5.5 AF\_DetachUserProxyFromVehicleProxy**

This application function detaches the given user from the given vehicle.

##### **11.5.5.1 Sequence Description**

For the given FIN, determine the corresponding VehicleProxy.

Check the assignment between the user and the vehicle and detach the user from the vehicle:

- If the vehicle is not assigned to a user or if the vehicle is assigned to another user than the user from the input then skip this step.
- If the vehicle is assigned to the user from the input, i.e. if the VehicleProxy.UserProxy.UserId matches the UserId from the input, then disconnect the UserProxy from the VehicleProxy.

##### **11.5.5.2 Input**

Parameter Name	Type / Length / BOM	Possible Values	Description
UserId	String / 50 / UserProxy.UserId		The user id of the customer.
FIN	String / 17 / VehicleProxy.FIN		The FIN of the vehicle.

Table 323: AF\_DetachUserProxyFromVehicleProxy input

##### **11.5.5.3 Output**

None.

##### **11.5.5.4 Exceptions**

Message Id	Fault Message	Error reason
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
ACCDAS_004	An unexpected error occurred while connecting to the system CPD.	if any unexpected, technical errors occur when connecting to CPD.

---

Table 324: AF\_DetachUserProxyFromVehicleProxy exceptions

## 11.5.6 AF\_DetachBusinessPartnerProxyFromVehicleProxy

### 11.5.6.1 General Description

This application function detaches the given business partner from the given vehicle.

### 11.5.6.2 Sequence Description

For the given Fin, determine the corresponding VehicleProxy.

Check the assignment between the business partner and the vehicle and detach the business partner from the vehicle:

- If the vehicle is not assigned to a business partner or if the vehicle is assigned to another business partner than the business partner from the input then skip this step.
- If the vehicle is assigned to the business partner from the input, i.e. if the VehicleProxy.BusinessPartnerProxy.BusinessPartnerId matches the BusinessPartnerId from the input, then disconnect the BusinessPartnerProxy from the VehicleProxy.

### 11.5.6.3 Input

Parameter Name	Type / Length / BOM	Possible Values	Description
BusinessPartnerId	String / 20 / BusinessPartnerProxy.BusinessPartnerId		The ID of the business partner.
Fin	String / 17 / VehicleProxy.FIN		The European vechicle identification number ("Fahrzeug-Identifizierungsnummer").

Table 325: AF\_DetachBusinessPartnerProxyFromVehicleProxy input

### 11.5.6.4 Output

None.

### 11.5.6.5 Exceptions

None.

## 11.5.7 AF\_GetOrCreateOrUpdateUserProxy

This application function retrieves the proxy object for the given user. The proxy object will be created if it does not exist yet or updated in case the address country has changed.

### 11.5.7.1 Sequence Description

For the given UserId determine the corresponding UserProxy.

If there is a UserProxy that matches the given UserId and if the input parameter addressCountry is given, set the UserProxy.CustAddressCountry to the AddressCountry from the input.

If there is no UserProxy that matches the given UserId, create a new UserProxy:

- If the AddressCountry is given as input parameter, create a new UserProxy with UserProxy.UserId = UserId and UserProxy.CustAddressCountry = AddressCountry
- If the AddressCountry is not given as input parameter, call the internal interface IIF\_GetUserProfile with the given UserId as input parameter to retrieve the customer data and create a new UserProxy with UserProxy.UserId = UserId and UserProxy.CustAddressCountry = User.CustAddressCountry

Return the UserProxy as output parameter.

#### 11.5.7.2 Input

Name	Type / Length / BOM	Description
UserId (Mand.)	String / 50 / User.UserId	The user id of the customer.
AddressCountry (Opt.)	String / 2 / UserProxy.CustAddressCountry	The address country of the customer.

Table 326: AF\_GetOrCreateOrUpdateUserProxy input

#### 11.5.7.3 Output

Name	Type / Length / BOM	Description
UserProxy	UserProxy	The proxy object for the customer.

Table 327: AF\_GetOrCreateOrUpdateUserProxy output

#### 11.5.7.4 Exceptions

Message Id	Fault Message	Error reason
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
ACCDAS_004	An unexpected error occurred while connecting to the system CPD.	if any unexpected, technical errors occur when connecting to CPD.

Table 328: AF\_GetOrCreateOrUpdateUserProxy exceptions

### 11.5.8 AF\_GetVehicleProxy

This application function retrieves the proxy object for the given vehicle.

#### 11.5.8.1 Sequence Description

For the given fin determine the corresponding VehicleProxy. Return the VehicleProxy as output parameter.

#### 11.5.8.2 Input

Parameter Name	Type / Length / BOM	Possible Values	Description
FIN	String / 17 / VehicleProxy.FIN		The FIN of the vehicle.

Table 329: AF\_GetVehicleProxy input

#### 11.5.8.3 Output

Parameter Name	Type / Length / BOM	Possible Values	Description
VehicleProxy	VehicleProxy		The proxy object for the vehicle.

Table 330: AF\_GetVehicleProxy output

---

#### **11.5.8.4 Exceptions**

None.

### **11.5.9 AF\_GetVehicleProxiesForUser**

This application function returns the proxy objects for all vehicles that are assigned to the given user.

#### **11.5.9.1 Sequence Description**

For the given userId determine the corresponding VehicleProxies. Return the VehicleProxies as output parameter.

#### **11.5.9.2 Input**

Parameter Name	Type / Length / BOM	Possible Values	Description
UserId	String /50 / User-Proxy.UserId		The user id of the customer.

Table 331: AF\_GetVehicleProxiesForUser input

#### **11.5.9.3 Output**

Parameter Name	Type / Length / BOM	Possible Values	Description
<b>List of VehicleProxies</b>			
VehicleProxy	VehicleProxy.FIN		The proxy object for the vehicle.

Table 332: AF\_GetVehicleProxiesForUser output

#### **11.5.9.4 Exceptions**

None.

### **11.5.10 AF\_GetVehicleProxiesAndAssociatedUserProxy**

This application function returns all vehicle proxies of the given model series and the associated user.

#### **11.5.10.1 Sequence Description**

Select all vehicle proxies for which the attribute of the given model series and return them.

#### **11.5.10.2 Input**

None.

#### **11.5.10.3 Output**

Name	Type / Length / BOM	Description
List of vehicle proxies and associated user proxies		
VehicleProxy	ContextProxy::VehicleProxy	A vehicle proxy
UserProxy	ContextProxy::UserProxy	User Proxy associated to the vehicle proxy

Table 333: AF\_GetVehicleProxiesAndAssociatedUserForModelSeries output

---

#### **11.5.10.4 Exceptions**

None.

### **11.6 Batches**

None.

### **11.7 Error Messages**

<b>Message Id</b>	<b>Fault Message</b>	<b>Error reason</b>
ACCDAS_002	The requested customer could not be found in the customer directory.	if the given user could not be resolved in the CPD.
ACCDAS_004	An unexpected error occurred while connecting to the system CPD.	if any unexpected, technical errors occur when connecting to CPD.
ACCDAS_005	Action is not possible, because the customer is notified in an unsupported country.	Action is not possible, because the customer is notified in an unsupported country.

Table 334: Component "Context Proxy" – error messages

# 12 Component “AccountDataSupport”

## 12.1 Dialogs

None.

## 12.2 External View - Offered Interfaces

### 12.2.1 IF\_SOE\_GetProfileFieldsUsedByCountry

Communication type: Synchronously

- This interface returns a list of fields which are needed in a certain country for recording customer data and sign a user agreement. The returned list covers account related profile data fields only! If no data fields can be found for the requested country, at least the field “ADDRESS\_COUNTRY”, “FIRST\_NAME”, “LAST\_NAME\_1” will be returned.

Note:

- Additionally to the profile data fields, this interface returns structural information on the data fields. For further information see **Configuration of Profile Data Fields** (→ see chapter 2.3.19).

Internally the AF\_GetProfileFieldsUsedByCountry (→ see chapter 12.5.3) is called to retrieve the list of fields.

#### 12.2.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
CountryCode	Mand.	String	2	Used as input parameter for AF_GetProfileFieldsUsedByCountry	Examples: “DE”, “CH”, “UK”, ...	The ISO country code of the country to get the required fields for.
Locale	Mand.	String	5	Used as input parameter for AF_GetProfileFieldsUsedByCountry	Examples: “de_DE”, “de_AT”, “fr_CH”, “de_CH”, ...	Locale (language in combination with a country). This field specifies in which locale the description fields of the returned enumeration values shall be delivered.

Table 335IF\_SOE\_GetProfileFieldsUsedByCountry Input

#### 12.2.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>StructureItems (0..*)</b>						
structureItem Type	Mand.	Enum-erati	-	As returned by the underlying AF.	{GROUP, DATAFIELD}	Determines if structureItemID references

Parameter Name	Mand./ Opt.	For- mat	Length	Data Model	Possible Values	Annotation
		on				to a data field or to a group.
structureItemID	Mand.	String	-	As returned by the underlying AF.	-	The id of a group field or a data field.
childrenIDs	Opt.	List of Enumeration	-	As returned by the underlying AF.	{see list in chapter 2.3.18}	The ID of the dependent fields. For the available values please refer to See <b>Handling of Profile Data within SOE</b> (→ see chapter 2.3.18).
<b>Fields (0..*)</b>						
FieldId	Mand.	Enu-merati- on	-	ProfileDataFieldMetadata.FieldId	{...}	The ID of the field. For the available values please refer to See <b>Handling of Profile Data within SOE</b> (→ see chapter 2.3.18).
Sequence	Mand.	Inte- ger	-	ProfileDataFieldMetadata.Sequence	Examples: 1, 2, 3...	The sorting sequence of the field.
FieldUsage	Mand.	Enu-merati- on	-	ProfileDataFieldMetadata.FieldUsage	{INVISIBLE, OPTIONAL, MANDATORY, READONLY}	INVISIBLE: Field not needed for the requested country OPTIONAL: Field needed, but does not have to be filled MANDATORY: Field needed and must be provided READONLY: Field can only be read and not set.
MatchSelectableValueByKey	Opt.	Bool- ean	-	ProfileDataFieldMetadata.MatchSelectableValueByKey	{true, false}	Is filled if this field is an enumerated field (=offers a list of selectable values). It states whether the consuming system must use the code (true) or the description (false) of the enumerated value when mapping the data to the consumer object for CPD.
defaultSelectableValue	Opt.	String	5	ProfileDataFieldMetadata.DefaultSelectableValue	Examples: "de_DE", "DE", ...	Is filled if this field is an enumerated field (=offers a list of selectable values). It states which of the selectable values shall be used as default (refers to the key of the respective element). If empty, there is no default.
<b>• FieldValidation (0..1)</b>						Is provided if the field needs special validation. Usually only for text fields.
MinLength	Mand.	Inte- ger	-	ProfileDataFieldMetadata.FieldValidation.MinLength	Examples: 1, 2, 3...	Minimum length of the provided value.
MaxLength	Mand.	Inte- ger	-	ProfileDataFieldMetadata.FieldValidation.MaxLength	Examples: 1, 2, 3...	Maximum length of the provided value. Can be used to limit the maximum input length of a text box.
RegularExpression	Opt.	String	100	ProfileDataFieldMetadata.FieldValidation.RegularExpression	Example: "\b[A-Z0-9._%-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b" for an email address	Optionally provides a regular expression to validate the entered text against. The regular expression is supported through the standard java library for regular expressions (see java.util.regex package).
<b>• SelectableValue (0..*)</b>						Is provided if the field is a

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
						selectable values field.
Key	Mand.	String	5	ProfileDataFieldMetadata.SelectableValue.Key	Example: "DE"	Code of the selectable value.
Description	Mand.	String	100	ProfileDataFieldMetadata.SelectableValue.Description (in the requested locale)	Example: "Germany"	Description of the selectable value.

Table 336 IF\_SOE\_GetProfileFieldsUsedByCountry Output

### 12.2.1.3 Exceptions

None.

## 12.3 External View - Consumed Interfaces

### 12.3.1 IF\_CPD\_getUserProfile

**Communication type:** Synchronously

This interface returns the customer profile data of a specific customer.

#### 12.3.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
UserId	Mand.	String		As requested by the calling AF.		The ID of the customer which shall be retrieved.

Table 337: IF\_CPD\_getUserProfile Input

#### 12.3.1.2 Output

Note: The incoming customer data fields and the mapping of this interface are described centrally in **Handling of Profile Data within SOE** (→ see chapter 2.3.18).

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Customer Data Fields</b>						
See <b>Handling of Profile Data within SOE</b> (→ see chapter 2.3.18).						

Table 338: IF\_CPD\_getUserProfile Output

#### 12.3.1.3 Exceptions

Error code 002 (RI\_CODE\_USER\_NOT\_FOUND): Customer does not exist.

Error code 999 (RI\_CODE\_UNEXPECTED\_ERROR): Es ist ein unerwarteter Fehler aufgetreten.

## 12.4 Internal View - Offered Interfaces

### 12.4.1 IIF\_GetUserProfile

This interface returns the customer profile data of a specific customer.

---

Internally AF\_GetUserProfile (→ section 12.5.2) is called.

### 12.4.2 IIF\_GetProfileDataFieldsAndGroups

This internal interface gives information on the profile data fields and groups. In detail, the structure and relationships between the data fields and groups will be returned. Internally AF\_GetProfileDataFieldsAndGroups (→ section 12.5.1) will be called.

## 12.5 Implementation

### 12.5.1 AF\_GetProfileDataFieldsAndGroups

This internal interface gives information on the profile data fields and groups. In detail, the structure and relationships between the data fields and groups will be returned. Note: this AF returns all fields that are either mandatory or optional in ALL countries.

#### 12.5.1.1 Sequence Description

##### **Step 1: Accumulate all groups**

For each instance of <ProfileDataFieldGroup> return <ProfileDataFieldGroup>.<groupID> with <ProfileDataFieldGroup>.<children>.<fieldID> for each child as <childrenIDs>.

##### **Step 2: Accumulate all data fields**

For each instance of <ProfileDataFieldItem> return <ProfileDataFieldItem>.<fieldID> with <ProfileDataFieldGroup>.<hasDependentMandatoryFields>.<fieldID> for each dependent mandatory or optional field as <childrenIDs>.

For each instance of <ProfileDataFieldItem>, also determine the parameters <fieldOwnerType> and <marketSpecific>:

- Therefore find the instance of <ProfileDataFieldIdEnum> where <fieldID> is identical for both instances and return <ProfileDataFieldIdEnum>.<fieldOwnerType>.
- Also examine all instances of <ProfileDataFieldMetaData> where <fieldID> is identical with <fieldID> from instances of <ProfileDataFieldItem>. If <ProfileDataFieldMetaData>.<fieldUsage> is optional for all <countryCode>, set the parameter <marketSpecific> to false, otherwise set <marketSpecific> to true.

#### 12.5.1.2 Input

None.

#### 12.5.1.3 Output

Name	Type / Length / BOM	Description
<b>List of groups (each of the following entries exist per group)</b>		
groupID	ProfileDataFieldGroup.groupId	The ID refers to a group.
childrenIDs	List of children.fieldID	Containing the IDs of the children.

List of data fields (each of the following entries exist per data field)		
datafieldID	ProfileDataFieldItem.fieldID	Depending on the relationship type, the ID refers to a data field or a group.
fieldOwnerType	ProfileDataFieldIdEnum.fieldOwnerType	{VEHICLE, ACCOUNT}
childrenIDs	ProfileDataFieldItem.hasDependentMandatoryFields	Containing the IDs of the dependent mandatory fields.
marketSpecific	Boolean	Determines if a data field is market specific or available globally.  True: data field is market specific False: data field is available globally

Table 339: Output parameters

#### 12.5.1.4 Exceptions

None.

### 12.5.2 AF\_GetUserProfile

This AF returns the customer profile data of a specific customer.

#### 12.5.2.1 Sequence description

Call IF\_CPD\_getUserProfile (→ see chapter 12.3.1) to retrieve the necessary data. Using the retrieved data create the UserProxy.

If IF\_CPD\_getUserProfile aborts with an error indicating that the customer could not be found, abort with error message ACCDAS\_002.

If any unexpected, technical errors occur when connecting to CPD, abort with error message ACCDAS\_004.

#### 12.5.2.2 Input

Name	Type / Length / BOM	Description
UserId	String	User ID of the customer whose profile shall be retrieved.

Table 340: AF\_GetUserProfile input

#### 12.5.2.3 Output

Name	Type / Length / BOM	Description
CustomerAttributes	List of CustomerAttributes	The customer with its data.
UserProxy	UserProxy	The proxy object for the customer.

Table 341: AF\_GetUserProfile output

#### 12.5.2.4 Exceptions

ACCDAS\_002 if the given customer could not be resolved.

ACCDAS\_004 if any unexpected, technical errors occur when connecting to CPD.

---

## **12.5.3 AF\_GetProfileFieldsUsedByCountry**

This AF returns the customer profile data fields which are used in a specific country to create a basic user profile for a customer. The returned list covers account related profile data fields only! For every field detailed information is provided.

### **Note:**

- The address country of the customer will be set to mandatory for all countries independent of the attribute set for this field in the object model. This is to ensure a correct behavior of the front end system regarding the handling of the address country.

### **12.5.3.1 Sequence description**

#### ***Step 1: Check MBconnect support of country:***

Call IIF\_CheckCountrySupportedByMbconnect with the committed country code and remember the result.

#### ***Step 2: Retrieve Profile Data Field Groups:***

Return all entries of the entity ProfileDataFieldGroup with its groupID and its child data fields. Add this structure to the list “StructureItems” with structureItemType = GROUP.

#### ***Step 3: Retrieve Dependent Data Fields:***

Return all entries of the entity ProfileDataFieldItem with its fieldID and its dependent data fields. Add this structure to the list “StructureItems” with structureItemType = DATAFIELD.

#### ***Step 4: Retrieve Profile Data Fields:***

Add all instances of ProfileDataFieldMetadata where ProfileDataFieldMetadata match the requested country (attribute ProfileDataFieldMetadata.CountryCode must match the requested country) and are account related (attribute ProfileDataFieldIDEnum.fieldOwnerType = ACCOUNT) to the list “Fields”.

If no fields are available for the requested country, return the information about the following fields:

- “EMAIL”:
  - ProfileDataFieldMetadata.fieldUsage = “OPTIONAL”
  - ProfileDataFieldMetadata.sequence = 1
- “FIRST\_NAME”:
  - ProfileDataFieldMetadata.fieldUsage = “MANDATORY”
  - ProfileDataFieldMetadata.sequence = 2
- “LAST\_NAME\_1”:
  - ProfileDataFieldMetadata.fieldUsage = “MANDATORY”
  - ProfileDataFieldMetadata.sequence = 3

- 
- “BIRTHDAY”:
    - ProfileDataFieldMetadata.fieldUsage = “MANDATORY”
    - ProfileDataFieldMetadata.sequence = 4
    -
  - “ADDRESS\_COUNTRY”:
    - ProfileDataFieldMetadata.fieldUsage = “MANDATORY”
    - ProfileDataFieldMetadata.sequence = 5
    - ProfileDataFieldMetadata.matchSelectableValueByKey = true
    - ProfileDataFieldMetadata.selectableValues = <retrieve the distinct list of countries based on the reunion of all preconfigured countries stored inside the SelectableValue entity>

### **12.5.3.2 Input**

See external interface “IF\_SOE\_GetProfileFieldsUsedByCountry”.

### **12.5.3.3 Output**

See external interface “IF\_SOE\_GetProfileFieldsUsedByCountry”.

### **12.5.3.4 Exceptions**

See external interface “IF\_SOE\_GetProfileFieldsUsedByCountry”.

## **12.5.4 AF\_InformOfCountryFieldsUpdates**

This AF sends the notification that the country fields have been updated for all supported countries.

### **12.5.4.1 Sequence Description**

The application configuration PROP\_PDF\_CONFIG is parsed to retrieve the list of all supported countries. The internal interface IIF\_InformOfMasterDataChange is called with the following parameters:

- <MasterDataResourceTypeEnum> = ACCOUNTDATASUPPORT
- <dataType> = COUNTRYFIELDS
- <subSet> = List of <Country code>

### **12.5.4.2 Input**

None.

### **12.5.4.3 Output**

None.

### **12.5.4.4 Exceptions**

None.

## **12.6 Batches**

None.

---

## 12.7 Error Messages

Message Id	Fault Title	Fault Message
ACCDAS_002	Customer not found	The requested customer could not be found in the customer directory.
ACCDAS_004	Error connecting to system CPD	An unexpected error occurred while connecting to the system CPD.

Table 342: Messages of Component “AccountDataSupport“

# 13 Component “MBconnect Countries”

The component handles the maintenance of the MBconnect countries and all related internal as well as external interfaces.

## 13.1 Dialogs

### 13.1.1 DLG\_CountriesOverview

The dialog provides an overview of the countries in SOE. In this dialog the user can create and maintain the countries that shall be supported by MBconnect.

DLG_CountriesOverview					
<a href="#">Add new country</a>					
Items per page <input type="button" value="10"/> Page <a href="#">1</a> <a href="#">2</a> <a href="#">3</a>					
ISO Code	Mbconnect country	GSSN ID	Enabled	Opened Sessions	Action
Es	Spain	GS12345677	Yes		<a href="#">Edit</a>
De	Germany	GS12345678	Yes		<a href="#">View</a>
It	Italy	GS12345679	Yes		<a href="#">Edit</a>

Figure 70: DLG\_CountriesOverview

#### 13.1.1.1 Buttons and functions

Remark: Please note that a delete button and function is explicitly not provided. The deletion of a specific country cannot be supported without checking all user agreements.

Linked Label / Button Labelling	Type	Action Description
<init>		Load all available countries and list them in the order of their ISO codes together with the permitted actions for each loaded country by calling the loading algorithm described in Chapter “Loading of master data elements”. (see details in subchapter “Dialog Elements States”)
“Add new Country”	Button	Navigates to the dialog DLG_CountryDetail (→ see chapter 13.1.2) to add a new country. The dialog is opened in new mode.

“Edit”	Button	Navigates to the dialog DLG_CountryDetail (→ see chapter 13.1.2) to edit the country. The dialog is opened in new mode.
“View”	Button	Navigates to the dialog DLG_CountryDetail (→ see chapter 13.1.2) to view the country. The dialog is opened in edit mode.

Table 343: Buttons and functions

### 13.1.1.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Column “ISO Code”	Label	The ISO code of the country.	<MbcCountry>.<countryCode>
Column “Mbconnect country”	Label	The internationalized name of the country.	<MbcCountry>.<name>
Column “GSSN ID”	Label	Displays the outletOutletId from DRD/GEMS/GSSN	MbcCountry.outletOutletId
Column “Enabled”	Label	The state of the MBconnect support. It determines if the country is available for the usage of MBconnect services or not.	If <MbcCountry>.<mbConnectEnabled> is true, return “Yes”, otherwise return “No”.
Table Column “Opened Sessions”	Label/Icon	Indicates whether the entity Document is edited inside a change session.	-

Table 344: Form fields and front-end data objects

### 13.1.1.3 Dialogue field validation

None.

### 13.1.1.4 Configurability (incl. setting for roles)

None.

### 13.1.1.5 Dialog Elements State

See states of the common elements displayed by the “Overview”dialogs.

## 13.1.2 DLG\_CountryDetail

This dialog allows the user to view or edit the country information, as well as create a new country entry. Once a country is created, it cannot be deleted anymore. In order to prevent fault user inputs, the country has to be enabled before it can be used for MBconnect.

### ***Enabling country for MBconnect:***

Once a country is enabled, it cannot be disabled anymore. Only for MBconnect enabled countries user agreements can be maintained.

ISO Code\*: De

GSSN ID: GS12345678

Enabled: Yes

Country Name:

Language	Translation
<b>Deutsch (Deutschland)</b>	Deutschland
Deutsch (Österreich)	
Deutsch (Schweiz)	
<b>English (United Kingdom)</b>	Germany
English (Switzerland)	
<b>French (France)</b>	Allemagne
French (Belgium)	
French (Switzerland)	
<b>Italian (Italy)</b>	Germania
Italian (Switzerland)	

Figure 71: DLG\_CountryDetail

### 13.1.2.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		<u>Edit and Read-Only mode:</u> Load all country associated information. If the country is supported by MBconnect (<MbcCountry>.<mbConnectEnabled> == true), then the button "Enable" is deactivated as well as the checkbox "Selectable for user". Use the loading algorithm described in Chapter "Loading of master data elements"
"Enable"	Button	Enables the country for MBconnect. This freezes MBconnect country state and prevents it from being altered. The button itself is then disabled. The column "Translation" thus can still be altered.
"Save"	Button	Saves the changes made for the specific country object. Bring the currently maintained Country in the change session of the user. ( see general algorithm Apply ChangeSessionOnElement)
"Cancel"	Button	Discards all changes and return to the calling dialog.

Table 345: Buttons and functions

### 13.1.2.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
ISO Code	Textbox	The ISO code is the primary key for the country object. It references to ISO-3166 Alpha 2.	<MbcCountry>.<countryCode>
GSSN ID	Textbox	Displays the outletOutletId from DRD/GEMS/GSSN	<MbcCountry>.<outletOutletId>
Enabled	Label	Describes whether the country is supported by MBconnect or not.  Default: No / False	<MbcCountry>.<mbConnectEnabled>
Table column "Language"	Label	Locale of the maintained translation.	-
Table column "Translation"	Textbox	The name of the country (localized).	<MbcCountry>.<name>
ReadOnlyInfo	Label	Presents this information: "The dialog is opened in read-only mode."	-

Table 346: Form fields and front-end data objects

### 13.1.2.3 Dialogue field validation

The ISO code of the country is mandatory.

### 13.1.2.4 Configurability (incl setting for roles)

None.

### 13.1.2.5 Dialog Elements State

Linked Label	Type	State Description
ISO Code	Textbox	Can only be edited when the country is created but not saved yet. Otherwise the field is read only.
outletOutletId	Textbox	Enabled: if dialog is in edit mode. Disabled: if dialog is in read-only mode.
"Enable"	Button	Enabled: as long as the Mbccountry.mbConnectEnabled is false. Disabled: when Mbccountry.mbConnectEnabled is true.
Table column "Translation"	Textbox	Enabled: if dialog is in edit mode. Disabled: if dialog is in read-only mode.
ReadOnlyInfo	Label	Visible: if dialog is in read-only mode Invisible: if dialog is in edit mode

Table 347: Dialog element states

## 13.2 External View - Offered Interfaces

### 13.2.1 IF\_SOE\_GetSupportedMBconnectCountries

The interface returns the list of countries supported by MBconnect. The list includes the ISO code of the country as well as the internationalized country name. Internally AF\_GetMBconnectCountries is called.

### 13.2.1.1 Input

Parameter Name	Mand./ Opt.	For- mat	Length	Data Model	Possible Values	Annotation
locale	Mand.	String	5	Used as input parameter for AF_GetSupportedMBconnectCountries	Example: "de_De"	The locale is used to retrieve the internationalized country name.

Table 348: External interface input

### 13.2.1.2 Output

Parameter Name	Mand./ Opt.	For- mat	Length	Data Model	Possible Values	Annotation
<b>List of Countries</b>						
countryCode	Mand.	String	2	Country.countryCode	Example: "De"	-
countryName	Mand.	String	.	Country.name	Germany	Internationalized country name.

Table 349: External interface output

### 13.2.1.3 Exceptions

No exceptions

## 13.3 External View - Consumed Interfaces

None.

## 13.4 Internal View - Offered Interfaces

### 13.4.1 IIF\_GetCountries

Internally calls AF\_GetCountries (→ see chapter 13.5.1) to retrieve all existing MbcCountry entities.

### 13.4.2 IIF\_UpdateCountries

Internally calls AF\_UpdateCountries (→ see chapter 13.5.2) to update all MbcCountry entities.

### 13.4.3 IIF\_IsMbconnectCountry

This interface checks if the given country is supported by MBconnect.

Calls AF\_IsMbconnectCountry (→ section 13.5.3) to check if the given country is supported by MBconnect.

### 13.4.4 IIF\_GetMBconnectCountries

This interface returns all instances of <MBcCountry> that is supported by MBconnect.

---

Calls AF\_GetMBconnectCountries (→ section 13.5.4) in order to return the MBconnect supported countries.

### **13.4.5 IIF\_GetOutletOutletIdsForAllMbcCountries**

Implemented by AF\_GetOutletOutletIdsForAllMbcCountries.

## **13.5 Implementation**

### **13.5.1 AF\_GetCountries**

#### **13.5.1.1 General Description**

This application function provides all existing MbcCountry entities.

#### **13.5.1.2 Sequence Description**

Determine all instances of entity <MbcCountry> that are available in SOE and return the data.

#### **13.5.1.3 Input**

None.

#### **13.5.1.4 Output**

Parameter Name	Type / Length / BOM	Description
<b>List&lt;MbcCountry&gt;: List of all MbcCountry entities</b>		
countryCode	MbcCountry.countryCode	The CountryCode of a Country.
mbConnectEnabled	MbcCountry.mbConnectEnabled	Flag that indicates if the country is supported by MBconnect.
outletOutletId	MbcCountry.outletOutletId	The outletOutletId of the country
- Inner List of <String(50)>		
Name	MbcCountry.name	Translation of a country's name (i18n).

Table 350: AF\_GetCountries output

#### **13.5.1.5 Exceptions**

None.

### **13.5.2 AF\_UpdateCountries**

#### **13.5.2.1 General Description**

This application function updates the Country entities.

#### **13.5.2.2 Sequence Description**

Compare the <MbcCountry> given as input parameter with the existing <MbcCountry> in SOE retrieved by calling AF\_GetCountries.

#### **If the updateMode = ADD**

Add all instances of <MbcCountry> which are given as input parameter and not parts of the existing <MbcCountry>.

---

#### If the updateMode = UPDATE

Update all instances of <MbcCountry> that are available in SOE which are not completely equal to the <MbcCountry> given as input parameter.

#### If the updateMode = DELETE

Delete all instances of <MbcCountry> that are available in SOE which are not parts of the <MbcCountry> given as input parameter.

#### 13.5.2.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the country entities.
<b>List&lt;MbcCountry&gt;: List of either all MbcCountry entities or specific MbcCountry entities – if given</b>		
countryCode	MbcCountry.countryCode	The CountryCode of a Country.
mbConnectEnabled	MbcCountry.mbConnectEnabled	Flag that indicates if the country is supported by MBconnect.
outletOutletId	MbcCountry.outletOutletId	The outletOutletId of the country
- Inner List of <String(50)>		
Name	MbcCountry.name	The internationalized name of a Country.

Table 351: AF\_UpdateCountries input

#### 13.5.2.4 Output

None.

#### 13.5.2.5 Exceptions

None.

### 13.5.3 AF\_IsMbconnectCountry

This AF checks if the given country is supported by MBconnect.

#### 13.5.3.1 Sequence description

If a <MbcCountry> with the given <countryCode> exists and <mbConnectEnabled> is true, return true as well. Otherwise, return false.

If the country code does not exist at all, log a warning ("The country with the country code <countryCode> does not exist."), return false and continue without exception.

#### 13.5.3.2 Input

Parameter Name	Type / Length / BOM	Description
Country	String	The ISO code of the country to check.

Table 352: AF\_IsMCconnectCountry input

#### 13.5.3.3 Output

Parameter Name	Type / Length / BOM	Annotation
isMBconnectCountry	Boolean	True: The country is supported by MBconnect. False: It is not.

Table 353: AF\_IsMCconnectCountry output

---

### **13.5.3.4 Exceptions**

None.

## **13.5.4 AF\_GetMBconnectCountries**

This AF returns all countries that are supported by MBconnect.

### **13.5.4.1 Sequence description**

From all <MBcCountry> instances, return a list with the countries where <mbConnectEnabled> is true.

The returned list considers the context from which this AF is called see loading algorithm described in Chapter “Loading of master data elements”.

### **13.5.4.2 Input**

None.

### **13.5.4.3 Output**

Parameter Name	Type / Length / BOM	Annotation
MBconnectEnabledCountries	List of MbcCountry	A list of the countries that are supported by MBconnect.

Table 354: AF\_GetMBconnectCountries output

### **13.5.4.4 Exceptions**

None.

## **13.5.5 AF\_GetOutletOutletIdsForAllMbcCountries**

### **13.5.5.1 General Description**

Returns the DRD/GSSN/GEMS OutletOutletIds for all MBconnect countries currently available in SOE. Also includes OutletOutletIds from countries which are not enabled yet.

### **13.5.5.2 Sequence Description**

- Determine and return all available instances of MbcCountry.outletOutletId.
- Eliminate duplicates.
- Ignore countries without a given outletOutletId.

### **13.5.5.3 Input**

No Input

### **13.5.5.4 Output**

Parameter Name	Type / Length / BOM	Annotation
outletOutletIds	List<String>	The list of DRD/GSSN/GEMS OutletOutletIds Possible values: GS12345678, GS23456789

Table 355: AF\_GetOutletOutletIdsForAllMbcCountries output

---

### **13.5.5.5 Exceptions**

None.

## **13.6 Batches**

None.

## **13.7 Error Messages**

<b>Message Id</b>	<b>Fault Title</b>	<b>Fault Message</b>
MBCSESSION_001	Service <Service> is locked by another user	The service <service> you are trying to modify is already being edited by user <user>. Changes are not possible and will be discarded automatically.

Table 356: Table of error messages

---

## **14 Component “Authorization Support”**

The component “AuthorizationSupport” is responsible for managing the authorization related data in SOE. For a further description of the roles and rights see MBconnect role concept (→ see chapter **2.3.20**).

When a service is edited inside a change session all associated authorizations (ResourceAuthorization entities) will be automatically brought inside the change session.

### **14.1 Dialogs**

None.

### **14.2 External View - Offered Interfaces**

None.

### **14.3 External View - Consumed Interfaces**

None.

## **14.4 Internal View - Offered Interfaces**

### **14.4.1 IIF\_GetServiceAuthorizations**

Internally calls AF\_GetServiceAuthorizations (→ see chapter 14.5.1) to retrieve all existing ServiceAuthorization entities.

### **14.4.2 IIF\_UpdateServiceAuthorizations**

Internally calls AF\_UpdateServiceAuthorizations (→ see chapter 14.5.2) to update all ServiceAuthorization entities.

### **14.4.3 IIF\_GetRightsForResource**

The IIF returns the rights for each role of the given resource.

Internally AF\_GetRightsForResource (→ section 14.5.3) is called.

### **14.4.4 IIF\_SetRightsForResource**

The IIF sets the rights for each role of the given resource.

Internally AF\_SetRightsForResource (→ section 14.5.4) is called.

### **14.4.5 IIF\_GetRightsByServiceMasterAndRole**

Internally calls AF\_GetRightsByServiceMasterAndRole (→ see chapter 14.5.5) to retrieve the right to a given servicemaster and given roles.

---

## 14.5 Implementation

### 14.5.1 AF.GetServiceAuthorizations

#### 14.5.1.1 General Description

This application function provides all existing ServiceAuthorization entities.

#### 14.5.1.2 Sequence Description

Determine all instances of entity <ServiceAuthorization> that are available in SOE and return the data.

#### 14.5.1.3 Input

None.

#### 14.5.1.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;ServiceAuthorization&gt;: List of all ServiceAuthorization entities</b>		
Right	AccessRight.Right	The Right on a Resource..
ResourceName	ResourceAuthorization.resourceName	The name of the resource.
EntitlementName	EntitlementRole.entitlementName	Name of an entitlement role.

Table 357: AF\_GetServiceAuthorizations output

#### 14.5.1.5 Exceptions

None.

### 14.5.2 AF\_UpdateServiceAuthorizations

#### 14.5.2.1 General Description

This application function updates the ServiceAuthorization entities.

#### 14.5.2.2 Sequence Description

Compare the <ServiceAuthorization> given as input parameter with the existing <ServiceAuthorization> in SOE retrieved by calling AF\_GetServiceAuthorizations.

##### If the updateMode = ADD

Add all instances of <ServiceAuthorization> which are given as input parameter and not parts of the existing <ServiceAuthorization>.

##### If the updateMode = UPDATE

Update all instances of <ServiceAuthorization> that are available in SOE which are not completely equal to the <ServiceAuthorization> given as input parameter.

##### If the updateMode = DELETE

Delete all instances of <ServiceAuthorization> that are available in SOE which are not parts of the <ServiceAuthorization> given as input parameter.

### 14.5.2.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the resource authorization entities.
<b>List&lt;ServiceAuthorization&gt;: List of either all ServiceAuthorization entities or specific ServiceAuthorization entities – if given</b>		
Right	AccessRight.Right	The Right on a Resource..
ResourceName	ResourceAuthorization.resourceName	The name of the resource.
EntitlementName	EntitlementRole.entitlementName	Name of an entitlement role.

Table 358: AF\_UpdateServiceAuthorizations input

### 14.5.2.4 Output

None.

### 14.5.2.5 Exceptions

None.

## 14.5.3 AF\_GetRightsForResource

This application function is called in order to read the authorization related data for the given resource.

### 14.5.3.1 Sequence Description

For the pair of <ResourceTypeEnum> and <ResourceName>, return a list of roles with the associated right for each role.

The returned list considers the context from which this AF is called see loading algorithm described in Chapter “Loading of master data elements”.

### 14.5.3.2 Input

Name	Type / Length / BOM	Description
ResourceTypeEnum	ResourceTypeEnum	The resource type of the changed master data. Only the value “SERVICE” is possibly for now.
resourceName	String	The name of the requested recourse type. In case of the resource type “SERVICE”, a resourceName is the same than a service ID of a service.

Table 359: AF\_GetRightsForResource input

### 14.5.3.3 Output

Name	Type / Length / BOM	Description
<b>List of roles and rights (the following entries exist for each role that is associated with the given resource)</b>		
Role	String	The role that the following rights applies to.
Right	String	The right that is associated with the role.

Table 360: AF\_GetRightsForResource output

### 14.5.3.4 Exceptions

None.

## 14.5.4 AF\_SetRightsForResource

This application function is called in order to set the authorization related data for a given resource inside a change session.

### 14.5.4.1 Sequence Description

#### Verify if ResourceAuthorization is already inside a change session

- Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter 17.4.1). If the error SESSION\_001 is thrown executions stops here.
- Verify if ResourceAuthorization is *already inside a change session of another user*: if yes, throw error AUTHSESSION\_001. Execution stops here.

Retrieve the change session of the logged on user by calling IIF\_GetChangeSessionForUser (see chapter 17.4.1). If the error SESSION\_001 is thrown executions stops here.

For the pair of <ResourceTypeEnum> and <ResourceName>, update or insert the given role with its associated right. (update the associations from a ResourceAuthorization element to the EntitlementRole and AccessRight elements).

Bring the ResourceAuthorization in the change session of the user ( see general algorithm **ApplyChangeSessionOnElement**).

If the given <ResourceTypeEnum>, <ResourceName>, <Role> or <Right> is invalid, quit the function with the exception AUTSUP\_001.

### 14.5.4.2 Input

Name	Type / Length / BOM	Description
ResourceTypeEnum	ResourceTypeEnum	The resource type of the changed master data. Only the value "SERVICEMASTER" is possibly for now.
resourceName	String	The name of the requested recourse type. In case of the resource type "SERVICEMASTER", a resourceName is the same than a service ID of a service.
<b>List of roles with rights (each of the following entries exist per role)</b>		
role	String	A valid MBconnect role.
right	String	A valid MBconnect right.

Table 361: AF\_SetRightsForResource input

### 14.5.4.3 Output

None.

### 14.5.4.4 Exceptions

Code	Message
AUTSUP_001	The given resource, role or right does not exist.

Table 362: AF\_SetRightsForResource exceptions

If the user doesn't have an opened change session, then the error SESSION\_001 is thrown.

---

## 14.5.5 AF\_GetRightsByServiceMasterAndRole

### 14.5.5.1 General Description

This application function returns the right for the given servicemaster and the given roles.

For a detailed description of the authorization roles and rights for servicemasters, see MBconnect role concept (→ see chapter 2.3.20).

### 14.5.5.2 Sequence Description

Go through <ResourceAuthorization> and select all entries where <ResourceTypeEnum> match “SERVICEMASTER”, resourceName match <Service-Master>.<ServiceMasterID>. For that list, select all entries where entitlementRole match one of the given roles.

In case only one role is given, return the right associated to that role. In case a list of roles is given, return the highest right.

Note: The order of the rights is described in see chapter MBconnect role concept (→ see chapter 2.3.20).

### 14.5.5.3 Input

Parameter Name	Type / Length / BOM	Description
serviceMasterId	ServiceMas-ter.serviceMasterId	The ID of a ServiceMaster.
• List<Role> (1..*)		
Role	ExternalEntitlementRolesEnum	The role for which SOE has to return the associated right.

Table 363: AF\_GetRightsByServiceMasterAndRole input

### 14.5.5.4 Output

Parameter Name	Type / Length / BOM	Description
serviceMasterId	ServiceMas-ter.serviceMasterId	The ID of a ServiceMaster.
right	AccessRight.Authoriz-ationAccessEnum	Determines the right the given roles have for the given servicemaster. The highest right of each role will be returned for the list of roles.

Table 364: AF\_GetRightsByServiceMasterAndRole output

### 14.5.5.5 Exceptions

None.

## 14.6 Batches

None.

## 14.7 Error Messages

Code	Message
AUTSUP_001	The given resource, role or right does not exist.

---

Code	Message
AUTHSESSION_001	The ResourceAuthorization < ResourceAuthorization > you are trying to modify is already being edited by user <user>. Changes are not possible and will be discarded automatically.

Table 365: Component “Authorization Support” error messages

---

# 15 Component “Users and Organizations”

## 15.1 Dialogs

None.

## 15.2 External View - Offered Interfaces

None.

## 15.3 External View - Consumed Interfaces

### 15.3.1 IF\_DRD\_GetOrganization

**Communication type:** Synchronously

This interface calls DRD to retrieve the organizations that exist beneath the given org unit.

#### 15.3.1.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Org Unit ID	Mand. .	String	-	OrgUnit	-	The org unit ID for which the child org units shall be returned.

#### 15.3.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Org Unit ID	Mand. .	String	-	OrgUnit.GSSNOutLetOutletID	-	The ID of the org unit.
VBET	Opt	String	-	OrgUnit.VBET		
Country	Opt.	String	-	OrgUnitCountry		
dcxlapSTARCONOrg Category	Mand. .	String	-	OrgUnit.OrgType	-	The category of the org units. e.g. "MPC", "Retailer", ...
Org Name	Mand. .	String	-	OrgUnit.Name	-	The name of the org unit.
...						

See Figure 72 for all fields provided by the DRD.

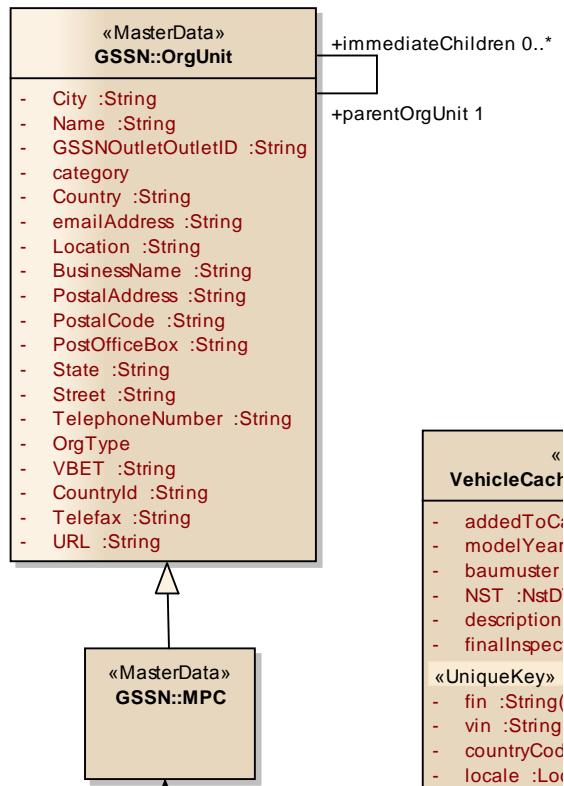


Figure 72 Attributes and Relationships provided by the Dealer Runtime Directory (DRD) for a OrgUnit

### 15.3.1.3 Exceptions

None.

## 15.4 Internal View - Offered Interfaces

### 15.4.1 IIF\_GetMPCByCountry

This interface determines a list of GEMS/GSSN outlet-outlet IDs of MPCs that correspond to a given Country (VBET).

Internally calls AF\_GetMpcByCountry (→ section 15.5.1) to retrieve the markets.

### 15.4.2 IIF\_GetMarkets

This internal interface retrieves the markets from the DRD based on the list of outletOutlets which are retrieved from the MBconnect Countries and returns them as a list.

Internally calls AF\_GetMarkets (→ section 15.5.2) to retrieve the markets.

## 15.5 Implementation

### 15.5.1 AF\_GetMpcByCountry

This application function determines a list of GEMS/GSSN outlet-outlet IDs of MPCs that correspond to a given Country (VBET).

### **15.5.1.1 Sequence Description**

The application function calls the third party system DRD in order to determine a list of GEMS/GSSN outlet-outlet IDs of MPCs that correspond to the given Country (VBET).

This is done by calling UAODRD\_001 - Problem while accessing the DRD. This is a general problem whenever the DRD cannot be reached.

AF\_GetMarkets and then search that list for the given marketNumber. (Because the MPCs are cached in SOE, this is the most efficient way.)

### **15.5.1.2 Input**

Parameter Name	Type / Length / BOM	Description
marketNumber	String (3)	The market number (VBET) of the sales unit. A matching between market number and market can be found in the application configuration.

Table 366: AF\_GetMpcByCountry input

### **15.5.1.3 Output**

Parameter Name	Type / Length / BOM	Annotation
outletOutletIDList	List of String(9)	A list of the GEMS/GSSN outlet-outlet IDs of the MPCs of the calling market.

Table 367: AF\_GetMpcByCountry input

### **15.5.1.4 Exceptions**

UAODRD\_001 - Problem while accessing the DRD. This is a general problem whenever the DRD cannot be reached.

## **15.5.2 AF\_GetMarkets**

This AF retrieves the markets from the DRD based on the list of outletOutletIds which are retrieved from the MBconnect Countries component and returns them as a list. The retrieved markets are cached by this AF for 24h. Any successive calls made to this AF within the time period of the cache, will receive the cached markets.

### **15.5.2.1 Sequence Description**

#### **Step 1: Retrieve outletOutletIds**

Call IIF\_GetOutletOutletIdsForAllMbcCountries to get the list of all outletOutletIds needed by SOE.

#### **Step 2: Retrieve Markets**

Call the LDAP server with the list of outletOutletIds to retrieve the organizations from the D-Tree.

Add all organizations to the list of markets to be returned. Cache the found markets.

### **15.5.2.2 Input**

None.

---

### 15.5.2.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of Organizations						
Org Unit ID	Mand. ..	String	-	OrgUnit	-	The ID of the org unit.
Org Name	Mand. ..	String	-	OrgUnit.Name	-	The name of the org unit.

### 15.5.2.4 Exceptions

- UAODRD\_001 - Problem while accessing the DRD. This is a general problem whenever the DRD cannot be reached.
- SERMAN\_009 - No continents are configured for the system. As a result, no markets (MPCs) can be determined for the continents. The system's configuration needs to be checked and fixed.

## 15.6 Batches

None.

## 15.7 Error Messages

None.

# 16 Component “Generic Master Data”

## 16.1 Dialogs

### 16.1.1 DLG\_GenericMasterDataOverview

#### 16.1.1.1 General Description

This dialogue is the entry point for maintaining the generic master data. The dialogue displays a list of keys with the optional key description and the uploaded data. The dialogue allows the creation, modification or deletion of each generic master data.

Items per page 10 Page 1   2   3				
	Key	Description	Opened Sessions	Action
<input type="checkbox"/>	otaRegions	This Key is necessary for the Service Mapupdate		<button>Edit</button>
<input type="checkbox"/>	phoneList	This Key is necessary for the Service FBS-Mobil		<button>View</button>
<button>Delete</button>				Items per page 10 Page 1   2   3
<button>Add new GenericMasterData</button>				

Figure 73: DLG\_GenericMasterDataOverview

#### 16.1.1.2 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		Load all available GenericMasterData and sort them by <GenericMasterData.Key> together with the permitted actions for each loaded GenericMasterData by calling the loading algorithm described in Chapter “ <b>Loading of master data elements</b> ”.
“Add new GenericMasterData”	Button	Navigates to the dialog DLG_GenericMasterDataDetail (see chapter 16.1.2) to add new Generic Master Data
“Edit” (1..n)	Button	Navigates to the dialog DLG_GenericMasterDataDetail (see chapter 16.1.2) to edit the selected Generic Master Data. The dialog is opened in edit mode.
“View” (1..n)	Button	Navigates to the dialog DLG_GenericMasterDataDetail (see chapter 16.1.2) to visualize the selected Generic Master Data. The dialog is opened in read-only mode.
“Delete”	Button	After displaying a confirmation popup delete the selected Generic Master Data.

Table 368: Buttons and functions (DLG\_GenericMasterDataOverview)

---

### 16.1.1.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Key	Label	The list of available PropertKeys.	GenericMasterData.Key
Description	Label	The list of available descriptions of GenericMasterData.	GenericMasterData.description
Table Column “Opened Sessions”	Label/Icon	Indicates whether the entity Document is edited inside a change session.	

Table 369: Form fields and front-end data objects (DLG\_GenericMasterDataOverview)

### 16.1.1.4 Dialogue field validation

None.

### 16.1.1.5 Configurability (incl setting for roles)

None.

### 16.1.1.6 Dialog Elements State

Linked Label	Type	State Description
Delete	Button	Enabled: If there is a open change session related to the current user Disabled: otherwise

Table 370: Dialog Elements State (DLG\_GenericMasterDataOverview)

## 16.1.2 DLG\_GenericMasterDataDetail

### 16.1.2.1 General Description

This dialog provides a detailed view of the GenericMasterData. A particular GenericMasterData is referenced via the Key.

A GenericMasterData File can be uploaded and thereby assigned to a Key and an optional description.

The File of a certain GenericMasterData can be re-uploaded in case of errors or updates that need to be fixed.

The screenshot shows a dialog box titled 'DLG\_GenericMasterDataDetail'. It has a 'Key\*' field containing 'otaRegions', a 'Description' field with the note 'This Key is necessary for the Service Mapupdate.', and a 'File Upload' section. In the 'File Upload' section, 'File Uploaded\*' is set to 'None', and there is a file input field showing 'C:\otaregions.xml', a 'Browse...' button, and an 'Upload' button. At the bottom are 'Save' and 'Cancel' buttons.

Figure 74: DLG\_GenericMasterDataDetail

#### 16.1.2.2 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		Edit mode: Load the generic master data for the selected Key.
"Browse..."	Button	Opens the native file chooser dialog to select the file for upload.
"Upload"	Button	Uploads the selected file to the server.
"Save"	Button	Save the generic master data (see AF_SaveGenericMasterData) and switch to DLG_GenericMasterDataOverview (see chapter 16.1.1).  Bring the currently maintained GenericMasterData in the change session of the user. (see general algorithmApplyChangeSessionOnElement)
"Cancel"	Button	Discard the changes and switch to DLG_GenericMasterDataOverview (see chapter 16.1.1).

Table 371: Buttons and functions (DLG\_GenericMasterDataDetail)

#### 16.1.2.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Key	Label	The Key which identifies the uploaded file.	GenericMasterData.Key
Description	Label	The description of a GenericMasterData.	GenericMasterData.description
File Uploaded	Label	Shows a link named as the propertKey to the uploaded file, if available. The file can be downloaded by the user if clicked on.	GenericMasterData.data
ReadOnlyInfo	Label	Presents this information: "The dialog is opened in read-only mode."	-

Table 372: Form fields and front-end data objects (DLG\_GenericMasterDataDetail)

#### 16.1.2.4 Dialogue field validation

Linked Field	Validation	Errormessage
Key	The Key cannot be left blank.	GMDM_001
File Uploaded	A file must be uploaded.	GMDM_002

Table 373: Dialogue field validations (DLG\_GenericMasterDataDetail)

#### 16.1.2.5 Configurability (incl setting for roles)

None.

#### 16.1.2.6 Dialog Elements State

Linked Label	Type	State Description
ReadOnlyInfo	Label	Visible: if dialog is in read-only mode Invisible: if dialog is in edit mode
<all buttons, textboxes, checkboxes>	Button/Checkboxes/Textboxes	Enabled: if dialog is in edit mode Disabled: if dialog is in read-only mode

Table 374: Dialog Elements State (DLG\_GenericMasterDataDetail)

### 16.2 External View - Offered Interfaces

#### 16.2.1 IF\_SOE\_GetGenericMasterDataForKey

##### 16.2.1.1 General Description

Communication type: <Asynchronously>

This internal interface provides the GenericMasterData for a given Key.

Internally, AF\_GetGenericMasterDataForKey (→ see chapter 16.5.1) is called.

##### 16.2.1.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Description
Key	Mand.	String		GenericMasterData.Key		The Key for GenericMasterData assigned to a service.

Table 375: IF\_SOE\_GetGenericMasterDataForKey input

##### 16.2.1.3 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Description
Data	Mand.	Binary		GenericMasterData.data		Binary attribute storing the data that is used as generic master data for the service.

Table 376: IF\_SOE\_GetGenericMasterDataForKey output

##### 16.2.1.4 Exceptions

Code	Message
GMDM_003	The requested Key <Key> is not known to SOE.

Table 377: IF\_SOE\_GetGenericMasterDataForKey exceptions

---

## **16.3 External View - Consumed Interfaces**

None.

## **16.4 Internal View - Offered Interfaces**

### **16.4.1 IIF\_GetGenericMasterData**

Internally calls AF\_GetGenericMasterDataForKey (→ see chapter 16.5.1) to retrieve all existing GenericMasterData entities.

## **16.5 Implementation**

### **16.5.1 AF\_GetGenericMasterDataForKey**

#### **16.5.1.1 General Description**

This application function provides the GenericMasterData for a given Key.

#### **16.5.1.2 Sequence Description**

For given <GenericMasterData.Key> the <GenericMasterData.data> is returned.

If such a Key does not exist for the given service, a functional error GMDM\_003 is returned.

#### **16.5.1.3 Input**

See external interface IF\_SOE\_GetGenericMasterDataForKey (→ see chapter 16.2.1).

#### **16.5.1.4 Output**

See external interface IF\_SOE\_GetGenericMasterDataForKey (→ see chapter 16.2.1).

#### **16.5.1.5 Exceptions**

See external interface IF\_SOE\_GetGenericMasterDataForKey (→ see chapter 16.2.1).

### **16.5.2 AF\_SaveGenericMasterData**

#### **16.5.2.1 General Description**

This AF saves a GenericMasterData for an already maintained or new Key.

#### **16.5.2.2 Sequence Description**

If the given Key exists, then apply the changes done to the <GenericMasterData> of the given Key.

Otherwise create a new instance of <GenericMasterData>, with <GenericMasterData.Key> as given Key, with <GenericMasterData.data> as given data and with <GenericMasterData.description> as given description if available.

#### **16.5.2.3 Input**

Name	Type / Length / BOM	Description
Key	GenericMasterData.Key	The Key for GenericMasterData assigned to a service.
Description	GenericMasterData.description	The description of a GenericMasterData.

Data	GenericMasterData.data	The data to be added/changed.
------	------------------------	-------------------------------

Table 378: AF\_SaveGenericMasterData input

#### 16.5.2.4 Output

None.

#### 16.5.2.5 Exceptions

None.

### 16.5.3 AF\_GetGenericMasterData

#### 16.5.3.1 General Description

This application function provides all existing GenericMasterData entities.

#### 16.5.3.2 Sequence Description

Determine all instances of entity <GenericMasterData> that are available in SOE and return the data.

#### 16.5.3.3 Input

None.

#### 16.5.3.4 Output

Parameter Name	Type / Length / BOM	Description
<b>List&lt;GenericMasterData&gt;: List of all GenericMasterData entities</b>		
Key	GenericMasterData.Key	The Key for GenericMasterData assigned to a service.
Description	GenericMasterData.description	The Description of a GenericMasterData
Data	GenericMasterData.data	Binary attribute storing the data that is used as generic master data for the service.

Table 379: AF\_GetGenericMasterData output

#### 16.5.3.5 Exceptions

None.

### 16.5.4 AF\_UpdateGenericMasterData

#### 16.5.4.1 General Description

This application function updates the GenericMasterData entities.

#### 16.5.4.2 Sequence Description

Compare the <GenericMasterData> given as input parameter with the existing <GenericMasterData> in SOE retrieved by calling AF\_GetGenericMasterData.

#### If the updateMode = ADD

Add all instances of <GenericMasterData> which are given as input parameter and not parts of the existing <GenericMasterData>.

#### If the updateMode = UPDATE

---

Update all instances of <GenericMasterData> that are available in SOE which are not completely equal to the <GenericMasterData> given as input parameter.

**If the updateMode = DELETE**

Delete all instances of <GenericMasterData> that are available in SOE which are not parts of the <GenericMasterData> given as input parameter.

#### 16.5.4.3 Input

Parameter Name	Type / Length / BOM	Description
updateMode	String	Determines the mode how to update the service entities.
<b>List&lt;GenericMasterData&gt;: List of all GenericMasterData entities</b>		
Key	GenericMasterData.Key	The Key for GenericMasterData assigned to a service.
Description	GenericMasterData.description	The Description of a GenericMasterData
Data	GenericMasterData.data	Binary attribute storing the data that is used as generic master data for the service.

Table 380: AF\_UpdateGenericMasterData input

#### 16.5.4.4 Output

None.

#### 16.5.4.5 Exceptions

None.

### 16.6 Batches

None.

### 16.7 Error Messages

Message Id	Fault Title	Fault Message
GMDM_001	Mandatory field missing	The field <field> cannot be left empty.
GMDM_002	File must be uploaded	A File must be present to save a GenericMasterData.
GMDM_003	Invalid Key	The requested Key <Key> is not known to SOE.

Table 381: Messages of the component Generic Master Data Management

# 17 Component “Master Data Change Management”

## 17.1 Dialogs

The change session is accessible directly from DLG\_Main. The dialogs DLG\_ChangeSessionMenu (see chapter 17.1.1), DLG\_ManageChangeSession (see chapter 17.1.2) and DLG\_TakeOverChangeSession (see chapter 17.1.3) describe how to work with a change session inside the UI.

### 17.1.1 DLG\_ChangeSessionMenu

*Note: This dialog is embedded inside the DLG\_Main as a button bar on the top left corner. It is not a standalone dialog.*

The button bar has two states depending on the status of the change session.

- If the change session is not opened than the menu bar offers the possibility to open the change session.

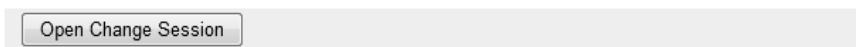


Figure 75: DLG\_ChangeSessionMenu (no opened change session)

- If the change session is opened than the menu offers the possibility to manage the own change session or to display and to take over the active change session of other users.



Figure 76: DLG\_ChangeSessionMenu (change session opened)

#### 17.1.1.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		For dialogue initialization the following AF(s) are called: AF_IsChangeSessionOpenedForUser (see chapter 11.4.1)
Open Change Session	Button	Will open a change session for the logged on user. Calls AF_OpenNewChangeSession (see chapter 17.5.3).
Manage Change Session	Button	Displays DLG_ManageChangeSession
Take over Change Sessions	Button	Displays DLG_TakeOverChangeSession

Table 382: Buttons and functions (DLG\_ChangeSessionMenu)

#### 17.1.1.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object Attribute (AS08 reference)
Change Session opened	Label	If the label is visible it indicates that the logged on user has opened the change session	-

Table 383: Form fields and front-end data objects (DLG\_ChangeSessionMenu)

### 17.1.1.3 Dialogue field validation

None.

### 17.1.1.4 Configurability (incl setting for roles)

None.

### 17.1.1.5 Dialog Elements State

Linked Label	Type	State Description
Open Change Session	Button	Will only be shown if the user has NOT opened the change session.
Manage my Change Session	Button	Will only be shown if the user has opened the change session.
Display others' Change Sessions	Button	Will only be shown if the user has opened the change session.
Change Session opened	Label	Will only be shown if the user has opened the change session.

Table 384: Dialog Elements State (DLG\_ChangeSessionMenu)

### 17.1.2 DLG\_ManageChangeSession

This dialog bundles all elements that have been modified inside the change session. It displays a list of the modified elements identified by the “element type”, “element name”, and other three attributes derived from the business key of the master data elements.

Additionally from this dialog the user can:

- Delete all modifications done during the change session and delete the change session as well
- Select particular elements that have been modified and discard modifications
- Release all modifications done during the change session and close the change session

This dialog opens as a pop-up dialog.

Manage Change Session

The following elements have been modified inside this change session.

Items per page 10 Page [1](#) [2](#) [3](#)

Type	Element Name	Attribute#1	Attribute#2	Attribute#3	Remove
Template	DEFAULT_TEMPLATE				<a href="#">Remove</a>
Template	AGB_TEMPLATE				<a href="#">Remove</a>
Document Block	UA REMOTE MAIN BLOCK	All	German(Germany)		<a href="#">Remove</a>
Document Block	UA REMOTE MAIN BLOCK	All	English(UK)		<a href="#">Remove</a>
User Agreement	Remote Services			PDF	<a href="#">Remove</a>

Items per page 10 Page [1](#) [2](#) [3](#)

[Delete Change Session](#) [Cancel](#) [Release All](#)

Figure 77: DLG\_ManageChangeSession

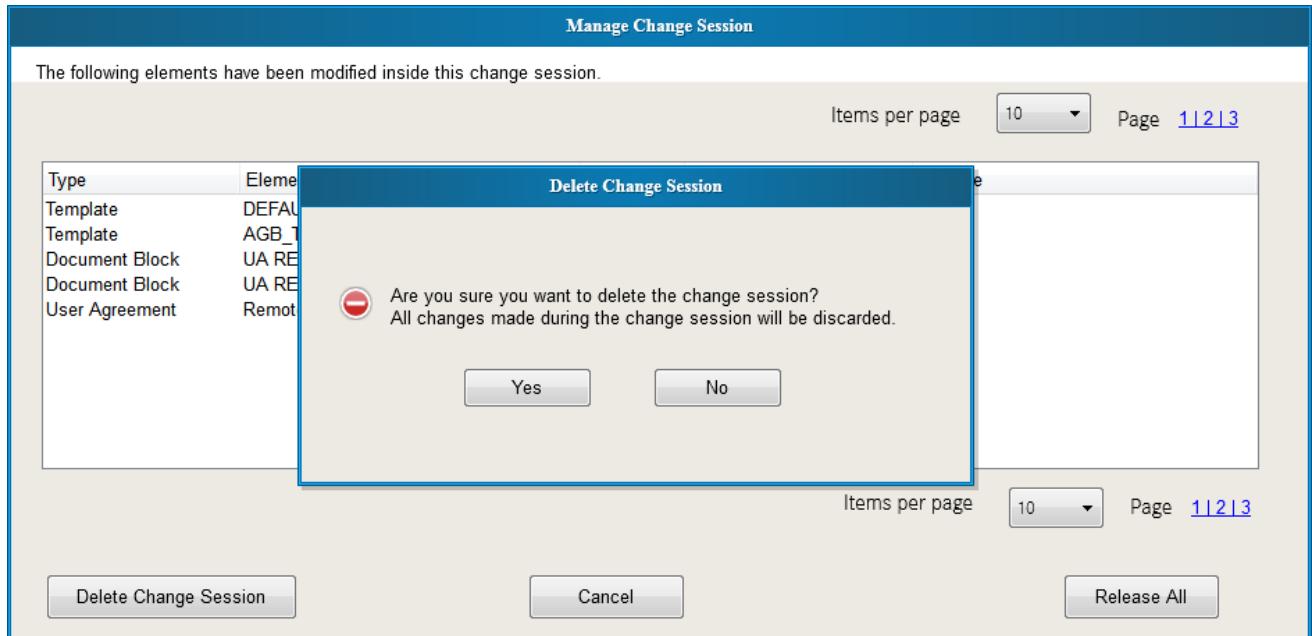


Figure 78: DLG\_ManageChangeSession – Delete Change Session PopUp

#### 17.1.2.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>	-	Load all elements modified inside the change session. Call AF_RetrieveModifiedElements (see chapter 17.5.6). Order lexicographically by "Type" (first criteria) and "Element Name" (second criteria)
Delete Change Session	Button	Displays a popup dialog (see Figure 78: DLG_ManageChangeSession – Delete Change Session PopUp , chapter 17.1.2). Confirming the popup dialog with: “YES”: if the user has already opened a change session, the error SESSION_002 is returned - AF_DeleteChangeSession is called (see chapter -). - “NO” : returns to DLG_ManageChangeSession (see chapter 17.1.2)
Release All	Button	All modifications done during the change session will be released for productive/operational usage. The change session is closed. Calls AF_ReleaseChangeSession (see chapter 17.5.7 )
Cancel	Button	Disposes dialog.
Delete modified element (1..n)	Button	Calls AF_DiscardModifications (see chapter 17.5.8) and refreshes dialog.

Table 385: Buttons and functions (DLG\_ManageChangeSession)

### 17.1.2.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
<b>Modified Elements Table (1..n Entries)</b>			
Type	String	Name of the modified element	-
Element Name	String	Name of the modified element (optional, applies only for elements that are referenced )	-
Attribute#1	String	The value of the master data attribute – usually is this identifier part of the primary key. This attribute is used for additional identification of the modified element.	-
Attribute#2	String	The value of a master data attribute – usually is this identifier part of the primary key. This attribute is used for additional identification of the modified element.	-
Attribute#3	String	The value of a master data attribute – usually is this identifier part of the primary key. This attribute is used for additional identification of the modified element.	-

Table 386: Form fields and front-end data objects (DLG\_ManageChangeSession)

### 17.1.2.3 Dialogue field validation

None.

### 17.1.2.4 Configurability (incl setting for roles)

None.

### 17.1.2.5 Dialog Elements State

None.

## 17.1.3 DLG\_TakeOverChangeSession

This dialog presents the list of SOE users that have opened a change session. From this overview the user will select all change sessions that he wants to acquire.

When acquiring change sessions of other users, all the modifications done by them will be brought inside the session of the user that has triggered the takeover.

*Note: this mechanism is available for all SOE users without restrictions.*

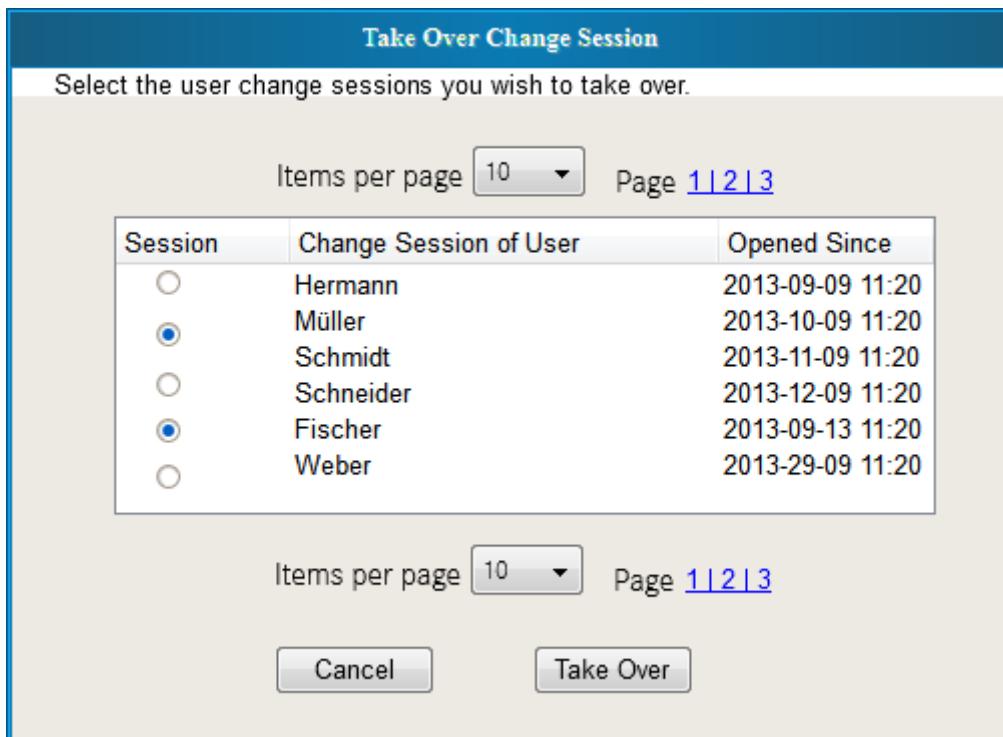


Figure 79: DLG\_TakeOverChangeSession

#### 17.1.3.1 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		From each change session extract the name of the user (field ChangeSession.ownerName). Exclude the user from which session the call is being made. Display the retrieved list inside the table.
Take Over	Button	A user is allowed to take over the change sessions of other users such that all master data elements of the other users will be brought into the change session of the user taking over the sessions. Call AF_TakeOverChangeSession (see chapter 17.5.5).
Cancel	Button	Disposes dialog.

Table 387: Buttons and functions (DLG\_TakeOverChangeSession)

#### 17.1.3.2 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
Table column "Change Session of User"	Label		ChangeSession.ownerName
Table column "Opened Since"	Label		ChangeSession.openedSince

Table 388: Form fields and front-end data objects (DLG\_TakeOverChangeSession)

#### 17.1.3.3 Dialogue field validation

None.

---

#### **17.1.3.4 Configurability (incl setting for roles)**

None.

#### **17.1.3.5 Dialog Elements State**

None.

### **17.1.4 DLG\_InformOfMasterDataChanges**

The component Change Session Management needs a dialog for administrative purpose (DLG\_InformOfAllMasterDataChanges), which has to have firewall protection. This dialog is available only for operations and no authentication is required.

The dialog offers the functionality to inform all dependent systems about a master data change with a provided button. Pushing the button calls AF\_InformOfMasterDataChange for all existing combinations of MasterDataResourceTypeEnum and MasterDataDataTypeEnum with an empty Subset. The possible combinations are described in Informing adjacent systems about master data changes (-> see chapter 2.3.12).

A screenshot of the required dialog is not defined.

## **17.2 External View - Offered Interfaces**

None.

## **17.3 External View - Consumed Interfaces**

None.

## **17.4 Internal View - Offered Interfaces**

### **17.4.1 IIF\_GetChangeSessionForUser**

This interface will retrieve the change session for a given user.  
Internally calls AF\_GetChangeSessionForUser (→ section 17.5.2).

### **17.4.2 IIF\_OpenNewChangeSession**

This interface will open the change session for a given user.  
Internally calls AF\_OpenNewChangeSession (→ section 17.5.3).

### **17.4.3 IIF\_IsChangeSessionOpenedForUser**

This interface will retrieve the change session for a given user.  
Internally calls AF\_IsChangeSessionOpenedForUser (→ section 17.5.1).

### **17.4.4 IIF\_ReleaseChangeSession**

This AF triggers each component for which master data has been modified, to release the modifications.  
Internally calls AF\_ReleaseChangeSession (→ section 17.5.7).

## 17.5 Implementation

### 17.5.1 AF\_IsChangeSessionOpenedForUser

This AF computes if a change session object is already associated for the currently logged on user. If the user has no change session open the AF returns TRUE otherwise FALSE.

#### 17.5.1.1 Sequence Description

Calls AF\_GetChangeSessionForUser (see chapter 17.5.2). In case the AF throws the error SESSION\_001, then return FALSE else return TRUE.

#### 17.5.1.2 Input

Parameter Name	Type / Length / BOM	Description
userID	String	ID of the SOE user

Table 389: Application function input (AF\_IsChangeSessionOpenedForUser)

#### 17.5.1.3 Output

Parameter Name	Type / Length / BOM	Annotation
opened	Boolean	TRUE if a change session exists for the user FALSE otherwise

Table 390: Application function output (AF\_IsChangeSessionOpenedForUser)

#### 17.5.1.4 Exceptions

None.

### 17.5.2 AF\_GetChangeSessionForUser

This AF retrieves the change session object for the given user.

#### 17.5.2.1 Sequence Description

Check for a matching change session object that has been opened by the user received as input parameter (ChangeSession.createdBy == userID).

In case no change session was found than exception SESSION\_001 is thrown.

#### 17.5.2.2 Input

Parameter Name	Type / Length / BOM	Description
userID	String	ID of the SOE user

Table 391: Application function (IIF\_GetChangeSessionForUser)

#### 17.5.2.3 Output

Parameter Name	Type / Length / BOM	Annotation
changeSession	ChangeSession	The change session associated for the user (identified by userID)

---

Table 392: Application function output (IIF\_GetChangeSessionForUser)

#### 17.5.2.4 Exceptions

- if the user has no change session associated the error SESSION\_001 is returned

### 17.5.3 AF\_OpenNewChangeSession

This AF creates a change session for the given SOE user.

#### 17.5.3.1 Sequence Description

##### Step1: Check if the user already opened the change session

In case that the call to AF\_IsChangeSessionOpenedForUser returns TRUE then throw error SESSION\_002. Execution stops here.

##### Step 2: Create the change session for user

A change session (ChangeSession object) is created and persisted for the given user.

#### 17.5.3.2 Input

Parameter Name	Type / Length / BOM	Description
userID	String	ID of the SOE user

Table 393: Application function input (IIF\_OpenNewChangeSession)

#### 17.5.3.3 Output

Parameter Name	Type / Length / BOM	Annotation
changeSession	ChangeSession	The change session associated for the user (identified by userID)

Table 394: Application function output (IIF\_OpenNewChangeSession)

#### 17.5.3.4 Exceptions

- if the user has already opened a change session, the error SESSION\_002 is returned

### 17.5.4 AF\_DeleteChangeSession

This AF triggers each component for which master data has been modified, to discard the modifications.

#### 17.5.4.1 Sequence Description

When a change session is deleted, the following actions are triggered:

- Discard all modifications done inside the change session:
  - Fetch all elements that are inside the change session of the user
  - Loop for each element:
    - o If the element has a productive form then delete the changed form and all the changed forms of the associated elements that have been created inside the change session.

- Else delete the changed form of the element.
- Delete the change session

#### **17.5.4.2 Input**

Name	Type / Length / BOM	Description
userID	String	ID of the SOE user.

#### **17.5.4.3 Output**

None.

#### **17.5.4.4 Exceptions**

None.

### **17.5.5 AF\_TakeOverChangeSession**

This AF retrieves from the component-pool the components for which master data has been modified.

#### **17.5.5.1 Sequence Description**

1. Take over the SourceChangeSession to the TargetChangeSession. All changed entities and elements related to the SourceChangeSession are taken over to the TargetChangeSession
2. Delete the SourceChangeSession.

#### **17.5.5.2 Input**

Name	Type / Length / BOM	Description
TargetChangeSession	ChangeSession	Change session of the user that wants to take over a session from another user.
SourceChangeSession	ChangeSession	change session that the user wants to acquire

#### **17.5.5.3 Output**

None.

#### **17.5.5.4 Exceptions**

- if the logged on user doesn't have an opened change session, then the error SESSION\_001 is thrown.

### **17.5.6 AF\_RetrieveModifiedElements**

This AF retrieves from the components for which master data has been modified, information about the modified master data elements inside each component.

#### **17.5.6.1 Sequence Description**

1. Load all elements that have been modified during the change session
2. Extract from the modified elements the following attributes (if available):

- **element type:** mapping in the table below

Modified Element	Element Type
ModelSeries	Model Series
Equipment	Equipment
Document	Document
DocumentTemplate	Document Template
DocumentBlock	Document Block
DocumentImage	Document Image
UserAgreementServiceAssignment	UserAgreementServiceAssignment
CustomTag	Custom Tag
MBcCountry	MBconnect Country
Service	Service
ServiceAssignmentRule	Service Assignment Rule
ResourceAuthorization	Resource Authorization

- **element name:** mapping in the table below

Element	Business-ID(s) of Element
ModelSeries	ModelSeries.modelSeriesID
Equipment	code
Document	Document.documentID
DocumentTemplate	DocumentTemplate.title
DocumentBlock	DocumentBlock.blockName
DocumentImage	DocumentImage.name
UserAgreementServiceAssignment	Document.documentID
CustomTag	CustomTag.tagType
MBcCountry	MBcCountry.countryCode
Service	Service.ID
ServiceAssignmentRule	ServiceAssignmentRule.serviceAssignmentRuleID
ResourceAuthorization	ResourceAuthorization.resourceType

- **Attribute#1** (if available as business key)

Element	Business-ID(s) of Element
ModelSeries	-
Equipment	-
Document	-
DocumentTemplate	-
DocumentBlock	DocumentBlock.country
DocumentImage	VersionedDocumentImage.versionID
UserAgreementServiceAssignment	Service.serviceName
CustomTag	-
MBcCountry	MBcCountry.name
Service	Service.name
ResourceAuthorization	ResourceAuthorization.resourceName
ServiceAssignmentRule	-

- **Attribute#2** (if available as business key)

Element	Business-ID(s) of Element

Element	Business-ID(s) of Element
ModelSeries	Not available
Equipment	
Document	-
DocumentTemplate	DocumentTemplate.country
DocumentBlock	DocumentBlock.locale
UserAgreementServiceAssignment	UserAgreement.versionID
CustomTag	-
MBcCountry	-
Service	-
ResourceAuthorization	-
ServiceAssignmentRule	-

- **Attribute#3** (if available as business key)

Element	Business-ID(s) of Element
ModelSeries	Not available
Equipment	
Document	-
DocumentTemplate	DocumentTemplate.locale
DocumentBlock	DocumentBlock.docType
CustomTag	CustomTag.DocumentTypeEnum
UserAgreementServiceAssignment	-
MBcCountry	-
Service	-
ResourceAuthorization	-
ServiceAssignmentRule	-

### 17.5.6.2 Input

Name	Type / Length / BOM	Description
changeSession	ChangeSession	The change session for which the modified master data elements shall be retrieved.

### 17.5.6.3 Output

Name	Type / Length / BOM	Description
<b>List of information about modified document elements: List&lt;String&gt;</b>		
elementType	String	Element Type name of the modified document element
elementName	String	Element name of the modified document element
Attribute#1	String	The value of the master data attribute – usually is this identifier part of the primary key. This attribute is used for additional identification of the modified element.
Attribute#2	String	The value of the master data attribute – usually is this identifier part of the primary key. This attribute is used for additional identification of the modified element.
Attribute#3	String	The value of the master data attribute – usually is this identifier part of the primary key. This attribute is used for additional identification of the modified element.

### 17.5.6.4 Exceptions

None.

## 17.5.7 AF\_ReleaseChangeSession

This AF triggers each component for which master data has been modified, to release the modifications.

---

Additionally IIF\_TriggerReplications is triggered for each type of master data that has changed.

### 17.5.7.1 Sequence Description

#### Step1: Loop for all changed elements

- Perform duplicate checks:

Check if the element to be released (the changed form) already exists as a productive form, based on the business key that is specific to each element (E.g. For a model series to be released, check to see if there already exists the given model series based on the attributes ModelSeries.modelSeriesID).

In case a duplicate has been detected, the release process will be aborted with the error SESSION\_003.

- Perform element specific checks:

***Check if the assignment(s) between service(s) and user agreement(s) has changed***

In case the entity “UserAgreementsServiceAssignment” has been modified than retrieve the UserAgreement entity and the Service entities that have been assigned to it (based on the algorithm described in the “Loading of master data elements”) and call the AF\_CheckUserServiceAssignment (from Component Service Management).

- If an element has been modified then make the changed form become the new productive form by taking over the values from the changed form. The changed form is deleted.
- If the element has been created: Create the productive form from the changed form. Then delete the changed form.
- If the element has been marked for deletion: Delete the productive and the changed form.

#### Step 2: Trigger SOE Regions with replication relevant changed master data

If there are changes associated to	Call IIF_TriggerReplications (→ see section 14.5.1) with the following input parameter MasterDataReplicationTypeEnum:
ModelSeries master data	MODELSERIES
SalesType master data	SAlestype
BodyType master data	BODYTYPE
Equipment master data	EQUIPMENT
MBconnect countries data	COUNTRY
ServiceMaster master data	SERVICEMASTER
ServiceCategory master data	SERVICECATEGORY
Service master data	SERVICE
ServiceAssignmentRule master data	SERVICEASSIGNMENTRULE
Generic master data	GENERICMASTERDATA
UserAgreementServiceAssignment master data	USERAGREEMENTSERVICEASSIGNMENT
DocumentTemplate master data	DOCUMENTTEMPLATE
Document master data	DOCUMENT
DocumentBlock master data	DOCUMENTBLOCK
Trigger2DocAssignment master data	TRIGGER2DOCASSIGNMENT
CustomTag master data	CUSTOMTAG

---

If there are changes associated to	Call IIF_TriggerReplications (→ see section 14.5.1) with the following input parameter MasterDataReplicationTypeEnum:
DocumentImage master data	DOCUMENTIMAGE
ServiceAuthorization master data	RESOURREAUTHORIZATION

Table 395: Trigger SOE Regions with changed master data

**Step3: Delete change session.**

#### 17.5.7.2 Input

Name	Type / Length / BOM	Description
changeSession	ChangeSession	The change session for which the modified master data elements shall be retrieved.

#### 17.5.7.3 Output

None.

#### 17.5.7.4 Exceptions

- If duplicate elements have been identified, the error SESSION\_003 will be thrown.

### 17.5.8 AF\_DiscardModifications

This AF triggers each component for which master data has been modified, to discard the modifications.

#### 17.5.8.1 Sequence Description

**Loop for each element:**

- If the element has a productive form then delete the changed form and all the changed forms of the associated elements that have been created inside the change session.
- Else delete the changed form of the element.

#### 17.5.8.2 Input

Name	Type / Length / BOM	Description
changeSession	ChangeSession	The change session for which the modified master data elements shall be retrieved.

#### 17.5.8.3 Output

None.

#### 17.5.8.4 Exceptions

- If duplicate elements have been identified, the error SESSION\_003 will be thrown.

## 17.6 Batches

None.

## 17.7 Error Messages

Message Id	Fault Title	Fault Message
------------	-------------	---------------

---

SESSION_001	No change session	You need to open a change session.
SESSION_002	Change session already opened	You already have opened a change session.
SESSION_003	Duplicate element detected	The element: <element> you have created/modified already exists. Please discard your modification.
SESSION_005	Cannot discard modification.	Adding the element <1> cannot be discarded because it is referenced by one or more elements of your change session.

Table 396: Error messages of component Component “Users and Organizations”

# 18 Component “Master Data Replication Export”

## 18.1 Dialogs

### 18.1.1 DLG\_ReplicationExportOverview

#### 18.1.1.1 General Description

The component Master Data Replication Export offers a dialog for administrative purpose (DLG\_ReplicationExportOverview).

The dialog offers the functionality to monitor the export of master data to SOE Regions.

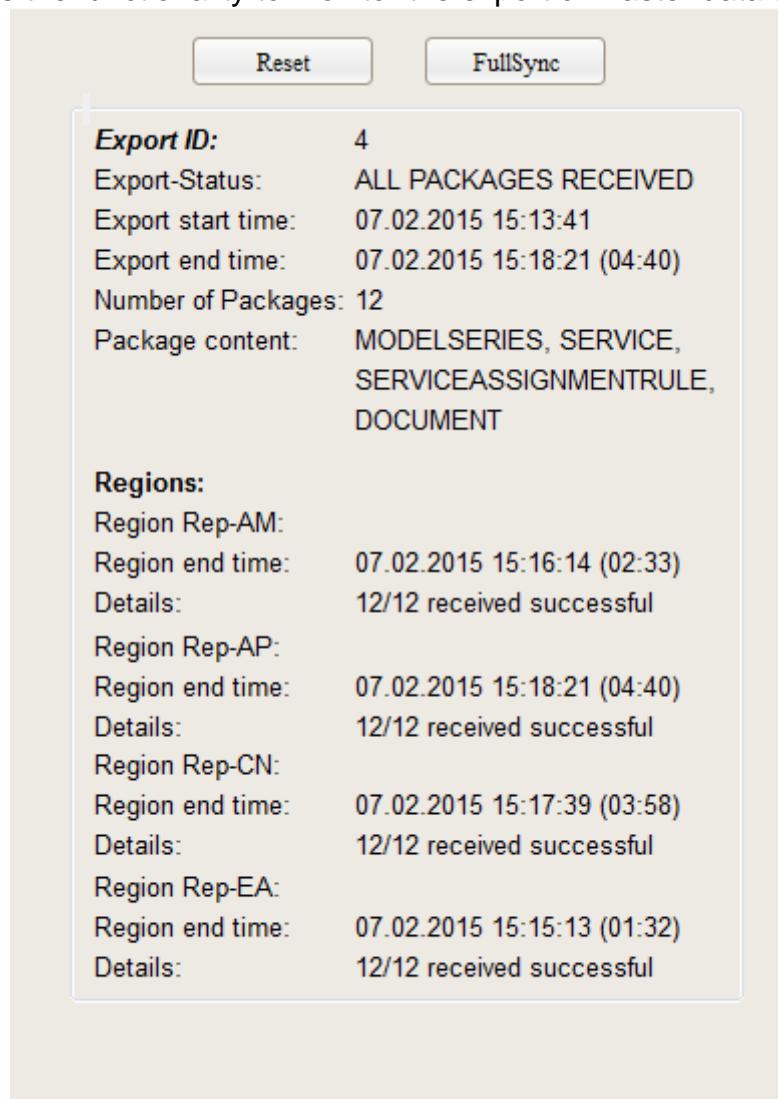


Figure 80: DLG\_ReplicationExportOverview

### 18.1.1.2 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		Initialize the fields below as follows in 18.1.1.3.
Reset	Button	Reset the export overview. Calls AF_ResetExportOverview (→ see chapter 18.5.4).
FullSync	Button	Trigger all SOE Regions to do a full synchronization of master data. Calls IIF_TriggerReplications (→ see chapter 18.4.1) with empty MasterDataReplicationTypeEnum.

Table 397: Buttons and functions (new dialogue)

### 18.1.1.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
- List of exports (beginning with the newest entry)			
Export ID:	Label	Lists the exportID of a specific export.	exportStatus.exportID
Export-Status:	Label	Lists the status of a specific export.	If the label "Export end time:" is filled: "ALL PACKAGES RECEIVED" Else: "REPLICATION IN PROGRESS"
Export start time:	Label	Lists the time when the export started.	exportStatus.startDate
Export end time:	Label	Lists the time when the amount of received packages matches the amount of packages for the export in all regions.	If all exportStatus.triggeredRegion.exportPackage.confirmedDate of each exportStatus.triggeredRegion are filled: The latest exportStatus.triggeredRegion.exportPackage.confirmedDate Else: Leave it empty
Runtime	Label	Lists the runtime of a finished export.	If the label "Export end time:" is filled: The date of the label "Export end time" - exportStatus.startDate
Number of Packages:	Label	Lists the number of packages of a specific export.	exportStatus.numberOfPackages
Package content	Label	Lists the content of all packages.	exportStatus.content
o List of regions			
Region	List of Label	Lists the regionID of a specific region.	"Rep-" + exportStatus.triggeredRegion.regionID
Region end time:	List of Label	Lists the time when the amount of received packages matches the amount of packages for the export in a region.	If all exportStatus.triggeredRegion.exportPackage.confirmedDate of this Region are filled: The latest exportStatus.triggeredRegion.exportPackage.confirmedDate Else: Leave it empty
Runtime	List of Label	Lists the runtime of a finished export in a region.	If the label "Region end time:" is filled: The date of the label "Region end time" - exportStatus.startDate
Details	List of Label	Lists the details of a specific export for a region.	Number of packages with filled exportStatus.triggeredRegion.exportPackage.confirmedDate + "/" + exportStatus.triggeredRegion.exportPackage + "received successful"

Table 398: Form fields and front-end data objects (new dialogue)

### 18.1.1.4 Dialogue field validation

None.

---

### **18.1.1.5 Configurability (incl setting for roles)**

None.

### **18.1.1.6 Dialog Elements State**

None.

## **18.2 External View - Offered Interfaces**

### **18.2.1 IF\_SOEMDM\_UpdateReplicationStatus**

#### **18.2.1.1 General Description**

**Communication type:** Asynchronously

This interface is necessary for a SOE Region to inform the SOE MDM about a received package.

Internally AF\_UpdateReplicationStatus (→ see chapter 18.5.3) is called.

#### **18.2.1.2 Input**

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
exportID	Mand.	Integer	5	-	Example: "11", "131", "1002"	The unique ID of an export.
replicationID	Mand.	String	2	-	Example: "EU", "CN"	The replicationID of a SOE Region.
PackageID	Mand.	Integer	4	-	Example: "2", "10", "41"	The ID of a received package within an export.

Table 399: IF\_SOE\_UpdateReplicationStatus input

#### **18.2.1.3 Output**

None.

#### **18.2.1.4 Exceptions**

None.

## **18.3 External View - Consumed Interfaces**

### **18.3.1 IF\_SOE\_UpdateMasterData**

#### **18.3.1.1 General Description**

**Communication type:** Asynchronously

This interface is called whenever a set of master data has changed. The master data is transported through export packets, which are identified by exportIDs.

#### **18.3.1.2 Input**

See IF\_SOE\_UpdateMasterData (→ see chapter 19.2.1 in Component “Master Data Replication Import”).

---

### 18.3.1.3 Output

None.

### 18.3.1.4 Exception

None.

## 18.4 Internal View - Offered Interfaces

### 18.4.1 IIF\_TriggerReplications

Internally calls AF\_TriggerReplications (→ see chapter 18.5.1).

## 18.5 Implementation

### 18.5.1 AF\_TriggerReplications

#### 18.5.1.1 General Description

This application function collects replication relevant changed master data entity and triggers the export of this master data to SOE Regions.

#### 18.5.1.2 Sequence Description

**If the input parameter MasterDataReplicationTypeEnum is empty:**

A full synchronization with all replication relevant Master Data is necessary.

Call AF\_ExportMasterDataBatch (→ see chapter 18.5.2).

**If the input parameter MasterDataReplicationTypeEnum is filled:**

A full synchronization with specific replication relevant Master Data is necessary.

Call AF\_ExportMasterDataBatch (→ see chapter 18.5.2) with

MasterDataReplicationTypeEnum as input parameter.

#### 18.5.1.3 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
MasterDataReplicationType	Opt.	MasterDataReplicationTypeEnum		-	{MODELSERIES, SALESTYPE, BODYTYPE, EQUIPMENT, SERVICE, SERVICEMASTER, SERVICECATEGORY, SERVICEASSIGNMENTRULE, GENERICMASTERDATA, USERAGREEMENTSERVICEASSIGNMENT, DOCUMENTITEMPLATE, DOCUMENT, DOCUMENTBLOCK, TRIG-	The master data replication type determines the entity that has changed.

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
					GER2DOCASSIGNMENT, CUSTOMTAG, DOCUMENTIMAGE, COUNTRY, SERVICEAUTHORIZATION}	

Table 400: AF\_TriggerReplications input

#### 18.5.1.4 Output

None.

#### 18.5.1.5 Exceptions

None.

### 18.5.2 AF\_ExportMasterDataBatch

#### 18.5.2.1 General Description

This application function performs the export of replication relevant changed master data to SOE Regions.

#### 18.5.2.2 Sequence Description

##### Step 1: Get all replication relevant changed master data

If the input parameter is empty or the input parameter MasterDataReplicationTypeEnum =	remember the entities received by calling
COUNTRY	IIF_GetCountries
BODYTYPE	IIF_GetBodyTypes
MODELSERIES	IIF_GetModelSeries
EQUIPMENT	IIF_GetEquipment
SALESTYPE	IIF_GetSalesType
SERVICEMASTER	IIF.GetServiceMasters
SERVICE	IIF_GetServices
SERVICECATEGORY	IIF.GetServiceCategories
SERVICEASSIGNMENTRULE	IIF.GetServiceAssignmentRules
GENERICMASTERDATA	IIF_GetGenericMasterData
DOCUMENTTEMPLATE	IIF_GetDocumentTemplates
DOCUMENTIMAGE	IIF_GetDocumentImages
CUSTOMTAG	IIF_GetCustomTags
DOCUMENTBLOCK	IIF_GetDocumentBlocks
DOCUMENT	IIF_GetDocuments
USERAGREEMENTSERVICEASSIGNMENT	IIF.GetUserAgreementServiceAssignment
SERVICEAUTHORIZATION	IIF.GetServiceAuthorizations
TRIGGER2DOCASSIGNMENT	IIF_GetTrigger2DocAssignments

Table 401: Receiving all replication relevant change master data entities

##### Step 2: Store the export details in the SOE database

- Create <exportStatus> with a unique <exportStatus.exportID>, with <exportStatus.numberOfPackages> given as the number of the separated export packages, with <exportStatus.startDate> given as the current time.
- Create <exportStatus.content> and fill it with all given

---

MasterDataReplicationTypeEnum as input parameter.

If the input parameter MasterDataReplicationTypeEnum is empty, fill <exportStatus.content> with “ALL ENTITIES”.

- Create all <exportStatus.triggeredRegion> with <triggeredRegion.regionID> given as PROP\_REGIONS\_TO\_TRIGGER (→ see chapter 2.3.15).
- For each export package create a <exportPackage> with a consecutively numbered <exportPackage.packageID> in each <triggeredRegion>.

#### **Step 4: Export the master data to SOE Regions**

Export all remembered entities separated in several packages by calling IF\_SOE\_UpdateMasterData (→ see chapter 18.3.1) with the unique exportID and the number of packages and each packageID to all SOE Regions given as PROP\_REGIONS\_TO\_TRIGGER (→ see chapter 2.3.15).

#### **18.5.2.3 Input**

None.

#### **18.5.2.4 Output**

None.

#### **18.5.2.5 Exceptions**

None.

### **18.5.3 AF\_UpdateReplicationStatus**

#### **18.5.3.1 General Description**

This application function updates the replication status of the master data export from SOE MDM to the SOE Regions.

#### **18.5.3.2 Sequence Description**

Create <exportPackage.confirmedDate> with the current time for the <exportPackage.packageID> given as input parameter packageID, for <triggeredRegion> given as input parameter regionID, in <exportStatus> given as input parameter exportID.

#### **18.5.3.3 Input**

See external interface IF\_SOEMDM\_UpdateReplicationStatu (→ see chapter 18.2.1).

#### **18.5.3.4 Output**

See external interface IF\_SOEMDM\_UpdateReplicationStatu (→ see chapter 18.2.1).

#### **18.5.3.5 Exceptions**

See external interface IF\_SOEMDM\_UpdateReplicationStatu (→ see chapter 18.2.1).

---

## **18.5.4 AF\_ResetExportOverview**

### **18.5.4.1 General Description**

This application function resets the export overview in the DLG\_ReplicationExportOverview (→ see chapter 18.1.1) and additionally deletes the export status entities.

### **18.5.4.2 Sequence Description**

Delete all <exportStatus> entities, if all <exportPackage>s have a filled <exportPackage.confirmedDate> in all <triggeredRegion>s and if the latest <exportPackage.confirmedDate> is smaller than two days.

### **18.5.4.3 Input**

None.

### **18.5.4.4 Output**

None.

### **18.5.4.5 Exceptions**

None.

## **18.6 Batches**

Batchname	Called Application Function	Description	Change
Batch "Export Master Data"	AF_ExportMasterDataBatch	<u>Execution time:</u> - <u>Execution frequency:</u> - This batch is triggered by AF_TriggerReplications.	New

Table 402: Batches

## **18.7 Error Messages**

None.

# 19 Component “Master Data Replication Import”

## 19.1 Dialogs

### 19.1.1 DLG\_ReplicationImportOverview

#### 19.1.1.1 General Description

The component Master Data Replication Import offers a dialog for administrative purpose (DLG\_ReplicationImportOverview).

The dialog offers the functionality to monitor the import of master data received by a SOE Region.

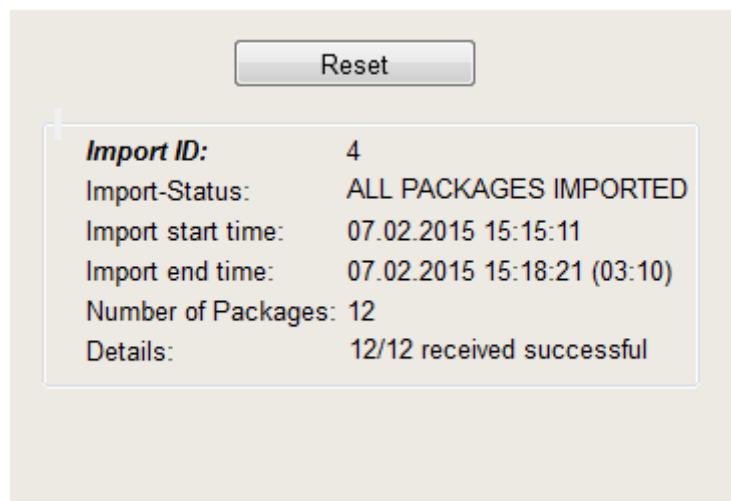


Figure 81: DLG\_ReplicationImportOverview

#### 19.1.1.2 Buttons and functions

Linked Label / Button Labelling	Type	Action Description
<init>		Initialize the fields below as follows in 19.1.1.3.
Reset	Button	Reset the import overview. Calls AF: AF_ResetImportOverview (→ see chapter 19.5.4).

Table 403: Buttons and functions (new dialogue)

### 19.1.1.3 Form fields and front-end data objects (AS08 Reference)

Linked Label	Type	Details / Default	Name Business Object. Attribut (AS08 reference)
- List of imports			
Import ID:	Label	Lists the importID of a specific import.	importDataBuffer.exportID
Import-Status:	Label	Lists the status of a specific import.	If importDataBuffer.isImported = TRUE: "ALL PACKAGES IMPORTED" Else If Number of <importDataBuffer.importPackage>s = importDataBuffer.numberOfPackages: "ALL PACKAGES RECEIVED" Else: "RECEIVING PACKAGES"
Import start time:	Label	Lists the time when the import started.	importDateBuffer.startDate
Import end time:	Label	Lists the time when the import finished.	If importDateBuffer.endDate is filled: importDataBuffer.endDate + "(" + importDataBuffer.endDate - importDataBuffer.startDate + ")" Else Leave it empty
Number of Packages:	Label	Lists the number of packages of a specific import.	importDataBuffer.numberOfPackages
Details:	Label	Lists the details of a specific import.	Number of <importDataBuffer.importPackage>s "/" importDataBuffer.numberOfPackages + "received successful"

Table 404: Form fields and front-end data objects (new dialogue)

### 19.1.1.4 Dialogue field validation

None.

### 19.1.1.5 Configurability (incl setting for roles)

None.

### 19.1.1.6 Dialog Elements State

None.

## 19.1.2 DLG\_InformOfMasterDataChanges

The component Change Session Management needs a dialog for administrative purpose (DLG\_InformOfAllMasterDataChanges), which has to have firewall protection. This dialog is available only for operations and no authentication is required.

The dialog offers the functionality to inform all dependent systems about a master data change with a provided button. Pushing the button calls AF\_InformOfMasterDataChange for all existing combinations of MasterDataResourceTypeEnum and MasterDataDataTypeEnum with an empty Subset. The possible combinations are described in Informing adjacent systems about master data changes (-> see chapter 2.3.12).

A screenshot of the required dialog is not defined.

## 19.2 External View - Offered Interfaces

### 19.2.1 IF\_SOE\_UpdateMasterData

#### 19.2.1.1 General Description

**Communication type:** Asynchronously

This interface is called whenever a set of master data has changed. The master data is transported through export packets, which are identified by exportIDs.

Internally AF\_ImportMasterData (→ see chapter 19.5.1) is called.

#### 19.2.1.2 Input

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
exportID	Mand.	Integer	5	-	Example: "11", "131", "1002"	The unique ID of an export.
NumberOfPackages	Mand.	Integer	3	-	Example: "13", "1", "102"	The number of packages within an export.
packageID	Mand.	Integer	4	-	Example: "2", "10", "41"	The ID of a package within an export.
<b>Content:</b>						
- List<ModelSeries> (0..*) – Optional						
modelSeriesID	Mand.	ModelSeriesDT		ModelSeries.modelSeriesID		The ID of a Model Series.
ProductGroup	Mand.	ProductGroupEnum		ModelSeries.ProductGroup		The product group of the model series.
maintainedInSoe	Mand.	Boolean		ModelSeries.maintainedInSoe		This flag is set to true if the ModelSeries is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
outOfProduction	Mand.	Boolean		ModelSeries.outOfProduction		This flag is set to true if the sales type is out of production. The flag is false by default.
○ Inner List of <MPC>						
gemsOutledID	Opt.	gemsOutletID		MPC.gemsOutledID		The GemsOutledID of a MPC
- List <SalesType> (0..*) – Optional						
Baumuster	Mand.	BaumusterDT		SalesType.Baumuster		The baumuster of a sales type.
NST	Opt.	NstDT		SalesType.Nst	Example: „CH1“	The NST code of the sales type (for passenger cars, i.e. product group 'P').
SR1	Opt.	Sr1DT		SalesType.SR1	Example: MG3+VP1+XZ1	The SR1 of the sales type (for transporters, i.e. product group 'T').
ModelNsrExtension	Opt.	ModelNsrExtensionDT		SalesType.ModelNsrExtension	Example: „BAD“	The ModelNsrExtension of the sales type (for transporters, i.e. product group 'T').
maintainedInSoe	Mand.	Boolean		SalesType.maintainedInSoe		The NST of a sales type. This flag is set to true if the SalesType is not available in AMDS and is

Parameter Name	Mand./ Opt.	Format	Length	Data Mo-del	Possible Values	Annotation
						exclusively maintained in SOE. The flag is false by default.
o Inner List of <SalesTypeDescription>						
Locale	Opt.	LocaleDT		SalesType Descripti-on.locale		The locale for which a SalesTypeDescription is valid.
Description	Opt.	String	256	SalesType Descripti-on.descripti-on		The (localized) description of a sales type.
maintainedInSoe	Opt.	Boolean		SalesType Descripti-on.maintai nedInSoe		This flag is set to true if the SalesTypeDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
o Inner List of <MPC>						
gemsOutletlD	Opt.	gemsOutletId DT		MPC.gems OutletID		The GemsOutledId of a MPC.
bodyTypeID	Opt.	BodyTypeDT		BodyType. bodyTyp-eID		The ID of a BodyType.
modelSeriesID	Opt.	ModelSeriesDT		ModelSerie s.modelSer iesID		The ID of a ModelSeries.
- List <BodyType> (0..*) – Optional						
bodyTypeID	Mand.	BodyTypeDT		BodyType. BodyTyp-eID		The ID of a BodyType.
ProductGrou p	Mand.	ProductGroup Enum		BodyType. ProductGro up		The product group of the body type.
maintainedInSOE	Mand.	Boolean		BodyType. maintained InSoe		This flag is set to true if the BodyType is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
o Inner List of <BodyTypeDescription>						
Locale	Opt.	LocaleDT		BodyType Descripti-on.locale		The locale for which a BodyTypeDescription is valid.
Description	Opt.	String	256	BodyType Descripti-on.descripti-on		The (localized) description of a body type.
maintainedInSoe	Opt.	Boolean		BodyType Descripti-on.maintai nedInSoe		This flag is set to true if the BodyTypeDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
- List <Equipment> (0..*) – Optional						
Code	Mand.	EquipmentCod eDT		Equip-ment.Code		The Code of the equipment.
ProductGrou p	Mand.	ProductGroup Enum		Equip-ment.Prod uctGroup		The product group of the equipment.
maintainedInSoe	Mand.	Boolean		Equip-ment.maint ainedInSoe		This flag is set to true if the Equipment is not available in AMDS and is

Parameter Name	Mand./ Opt.	Format	Length	Data Mo-del	Possible Values	Annotation
						exclusively maintained in SOE. The flag is false by default.
relevantFor ServiceAssignmentRule	Mand.	Boolean		Equip-ment.relev antForServ iceAssignm entRule		The flag is set to true if the equipment code is relevant for maintaining the service assignment rules. Only equipment codes with this flag set to true are available on DLG_ServiceAssignmentR uleDetails. The flag is false by default.
o Inner List of <EquipmentDescription>						
Locale	Opt.	LocaleDT		Equipment Descripti-on.locale		The locale for which an EquipmentDescription is valid.
Description	Opt.	String	256	Equipment Descripti-on.descripti on		The (localized) description of an Equipment.
maintainedInSoe	Opt.	Boolean		Equipment Descripti-on.maintai nedInSoe		This flag is set to true if the EquipmentDescription is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
- List <ServiceMaster> (0..*) – Optional						
serviceMasterId	Mand.	Integer		Service-Mas-ter.service MasterId		The ID of a ServiceMas-ter.
serviceCategoryID	Mand.	Integer	-	ServiceCat egory.Servi ceCategory ID	1, 2, 3	The ServiceCategoryID identifies the service category the service is assigned to.
businessArea	Mand.	BusinessArea Enum		Service-Mas-ter.busines sArea	"B2C", "B2B"	The business area of the service master. If the business area is set to "B2C" then the services assigned to this service master are available for regular customers. If the business area is set to "B2B" then the services assigned to this service master are available for business partners.
activateAutomatically	Mand.	Boolean		Service-Mas-ter.activate Automatica- lly		States if service(s) assigned to the service master is/are activated automatically when the vehicle is registered to a customer.
confirmationTokenNeeded	Mand.	Boolean		Service-Mas-ter.confirm ationToken Needed		If "TRUE" service(s) assigned to the service master need(s) a confirmed connection between the user and a vehicle through a verification token in. Such service(s) must be activated only after a successful

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
						verification.
personalVerificationNeeded	Mand.	Boolean		Service-Master.personalVerificationNeeded		If "TRUE" a retailer has to verify the identity of the customer in order to use service(s) assigned to the service master.
trustLevel	Mand.	Integer		Service-Master.trustLevel		Determines the minimal trust level of a vehicle registration in order to use the service(s) assigned to the service master.
o Inner List of <String(256)>						
name	Opt.			Service-Master.name		Translation of a service master name (i18n).
o Inner List of custom properties						
customProperty	Opt.	String		Service-Master.customProperties		The name of the custom property.
o Inner List of <ProfileDataFieldItem>						
requiredCustomer-Information	Opt.	AbstractProfile DataFieldRelationship		Service-Master.requiredCustomerInformation.fieldID		List with the mandatory fields for service(s) assigned to the service master that relate to attributes of the customer profile.
requiredVehicle-Information	Opt.	AbstractProfile DataFieldRelationship		Service-Master.requiredVehicleInformation.fieldID		List with the mandatory fields for service(s) assigned to the service master that relate to attributes of a vehicle.
o Inner List of <ProfileDataFieldGroup>						
requiredCustomer-Information	Opt.	AbstractProfile DataFieldRelationship		Service-Master.requiredCustomerInformation.groupID		List with the mandatory fields for service(s) assigned to the service master that relate to attributes of the customer profile.
requiredVehicle-Information	Opt.	AbstractProfile DataFieldRelationship		Service-Master.requiredVehicleInformation.groupID		List with the mandatory fields for service(s) assigned to the service master that relate to attributes of a vehicle.
- List <Service> (0..*) – Optional						
serviceID	Mand.	Integer		Service.serviceID		The ID of a Service.
partNumber	Opt.	String		Service.partNumber		The part number of a Service ("Sachnummer").
versionNumber	Mand.	Integer		Service.versionNumber		The Version of a Service.
licenseRequired	Mand.	Boolean		Service.licenseRequired		Tells whether a license is required or not in order to use this service.

Parameter Name	Mand./ Opt.	Format	Length	Data Mo-del	Possible Values	Annotation
contractDuration	Opt.	Int		Service.contractDuration		Describes the contract duration in months, in case a contract for this service is established. This attribute is optional.
contractStartTrigger	Mand.	ContractStartTriggerEnum		Service.contractStartTrigger		Indicates what event triggers the begin of a contract related to this service.
enabledFrom	Mand.	DateDT		Service.enabledFrom		The start date of a service (including this day). From this date the service is available of the sales' point of view.
enabledTo	Mand.	DateDT		Service.enabledTo		The last date of a service (including this day). From this date the service is not available of the sales' point of view.
technicalActivationPath	Mand.	TechnicalActivationPathEnum		Service.technicalActivationPath	"ACTIVATE_VIA_DAIVB", "ACTIVATE_VIA_FBS", "ACTIVATE_VIA_OMADM", "ACTIVATE_VIA_ADAPTER_BACKEND", "NO_ACTIVATION_NEEDED"	Determines the technical activation path, i.e. the system, that has to be contacted in order to activate or deactivate the service in the vehicle.
serviceClass	Mand.	ServiceClassEnum		Service.serviceClass	CONNECT_SERVICE, ADAPTER_SERVICE	Mercedes connect me-services are available for Mercedes connect me-vehicles exclusively. Mercedes connect me-vehicles are vehicles that have a communication module. Adapter-services are available for adapter vehicles exclusively. Adapter-vehicles are vehicles that have an OBD II adapter-interface but no communication module.
serviceMasterId	Mand.	Integer		Service-Master.serviceMasterId		The ID of the ServiceMaster.
○ Inner List of <String(4096)>						
description	Opt.			Service.description		Translation of a service's description (i18n).
○ Inner List of <MbCountry>						
countryCode	Opt.	CountryCodeDT		MbcCountry.countryCode		The CountryCode of a Country.
<b>List of Service Categories (each of the following parameters exist per service category)</b>						
serviceCategoryID	Mand.	Integer	-	ServiceCategory.ServiceCategory	1, 2, 3	The ServiceCategoryID identifies the service category the service is as-

Parameter Name	Mand./ Opt.	Format	Length	Data Mo-del	Possible Values	Annotation
				ID		signed to.
sortOrder	Mand.	Integer	-	ServiceCategory.Sort Order	-	The version of a service.
<b>Inner List of service category names (each of the following entries exist per locale)</b>						
Locale	Mand.	String	5	-	Examples: de_DE, de_AT, fr_CH, de_CH, en, de, fr	Locale (language in combination with or without a country). Specifies which language the service master name and the service description is are supposed to be created in.
serviceCategoryName	Mand.	String	-	ServiceCategory.name		The localized service category name.
<b>- List &lt;ServiceAssignmentRule&gt; (0..*) – Optional</b>						
serviceAssignmentRuleId	Mand.	Integer		ServiceAssignmentRule.serviceAssignmentRuleId		The ID of a ServiceAssignment Rule.
Description	Mand.	String	50	ServiceAssignmentRule.description		A name for a rule. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
longDescription	Mand.	String	1024	ServiceAssignmentRule.longDescription		A comment on the rule that explains or summarizes the rule in greater detail. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
modelSeriesID	Opt.	ModelSeriesDT		ModelSeries.modelSeriesID		The ID of a ModelSeries.
<b>o Inner List of &lt;YearCodeCombination&gt;</b>						
ModelYearCode	Opt.	EquipmentCodeDT		ModelYearCode.YearCodeCombination.ModelYearCode		The Code of a ModelYear.
ChangeYearCode	Opt.	String		YearCodeCombination.ChangeYearCode		The Code of a ChangeYear
<b>o Inner List of &lt;AndTerm&gt;</b>						
AndTermOrder	Opt.	Integer		AndTerm.displayOrder		The sequence number of OR terms within Boolean expression. Used to restore the OR terms in the sequence originally created.
<b>▪ Inner List of &lt;OrTerm&gt;</b>						
Code	Opt.	EquipmentCodeDT		OrTerm.code		The Code of an Equipment

Parameter Name	Mand./ Opt.	Format	Length	Data Mo-del	Possible Values	Annotation
isNegated	Opt.	Boolean		OrTerm.negated		Boolean flag that indicates whether the equipment code is negated within Boolean expression
OrTermOrder	Opt.	Integer		OrTerm.displayOrder		The sequence number of equipment code within OR term of Boolean expression. Used to restore equipment codes in the OR term in the sequence originally created.
o SalesTypeCondition						
baumuster-Wildcard	Opt.	String	10	SalesType Condition.b aumusterW ildcard		Search condition for a matching Baumuster. This pattern may contain wild-cards.
▪ Inner List of <NST>						
NST	Opt.	NstDT		SalesType Condition.NST		Search condition for a National Sales Type.
useConditionBaumuster -Wildcards	Opt.	Boolean		SalesType Condition.u seConditionBaumuste rWildcards		Boolean flag that indicates if the search condition under BaumusterWildcard shall be used during a search (TRUE) or ignored (FALSE).
useConditionNst	Opt.	Boolean		SalesType Condition.u seConditionNst		Boolean flag that indicates if the search condition under Nst shall be used during a search (TRUE) or ignored (FALSE).
o ConsumerCountryCondition						
▪ Inner List of <MbcCountry>						
countryCode	Opt.	CountryCodeD T		MbcCountry.countryC ode		The CountryCode of a Country.
useCondition	Opt.	Boolean		Consumer CountryCo ndition.use Condition		Tells whether the country condition shall be taken in account (true) or not (false) when the rule is evaluated.
o Inner List of <Service>						
ServiceID	Opt.	Integer		Servi-ce.servicel D		The ID of a Service.
- List <GenericMasterData> (0..*) – Optional						
Key	Mand.	String		GenericMa sterData.K ey		The Key for GenericMasterData assigned to a service.
Data	Mand.	Binary		GenericMa sterData.d ata		Binary attribute storing the data that is used as generic master data for the service.
Description	Opt.	String		GenericMa sterData.d escription		The Description of a GenericMasterData.
- List <UserAgreementServiceAssignment> (0..*) – Optional						
DocumentID	Mand.	String	50	Docu-ment.docu mentID		The DocumentID of a Document.
VersionID	Mand.	Integer		Versioned Docu-		The VersionID of a VersionedDocument.

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
				ment.versi onID		
o Inner List of <Service>						
ServiceID	Mand.	Integer		Service.ServiceID		The ServiceID of a Service.
- List <DocumentTemplate> (0..*) – Optional						
Country	Mand.	CountryCodeDT		DocumentTempl ate.country. countryCode		The Country of a DocumentTemplate
Locale	Mand.	LocaleDT		DocumentTempl ate.locale		The Locale of a DocumentTemplate.
Title	Mand.	String	50	DocumentTempl ate.title		The Title of a DocumentTemplate.
o Inner List of <VersionedDocumentTemplate>						
versionId	Opt.	Integer		Versioned DocumentTempl ate.versionID		Version Id of the document template. This version is independent of the VersionedDocument's version id.
Pdf	Opt.			Versioned DocumentTempl ate.pdf		Binary attribute storing the PDF document that is used as template for the document. This is the actual template data.
- List <Document> (0..*) – Optional						
DocumentID	Mand.	String	50	Docu ment.docu mentID		The ID of a Document.
DocType	Mand.	DocumentTypeEnum		Docu ment.docType		The Type of a Document.
isUserAgreement	Mand.	Boolean		Docu ment.isUserAgreement		Indicates whether a document represents a user agreement or not. (This is necessary for special functional treatment of user agreements.)
o Inner List of <VersionedDocument>						
versionId	Opt.	Integer		Versioned Document.versi onId		The VersionID of a Versioned Document.
documentXmlDefinition	Opt.	XML		Versioned Document.docu mentXmlDefinition		Root entity of the document definition. It represents the brace for all contained elements.
validFrom	Opt.	DateDT		Versioned Document.validFrom		The validFrom Date of a VersionedDocument.
▪ Inner List of <String(50)>						
Title	Opt.	String	50	Versioned Document.title		Translation of a versioned document's title (i18n)
▪ Inner List of <MbcCountry>						
Country		CountryCodeD		MbcCountr		The Country of a Ver-

Parameter Name	Mand./ Opt.	Format	Length	Data Mo-del	Possible Values	Annotation
		T		y.countryCode		sioned Document
o UserAgreement						
Enabled	Opt.	Boolean		UserAgreement.enabled		Optional: Boolean flag that indicates if the user agreement is enabled and thus be available to be processed, e.g. for signing. It also indicates whether the dates for a user agreement can still be changed or not. Once the flag is TRUE, it is not possible to change the dates any more.
informByEMailDate	Opt.	DateDT		UserAgreement.informByEMailDate		Optional: Date when the notifications of user agreement changes are sent out via email (for customers with mail address).
informByLetterDate	Opt.	DateDT		UserAgreement.informByLetterDate		Optional: Date when the notifications of user agreement changes are sent out via letter (for customers without mail).
validFromForExisting-Customer	Opt.	DateDT		UserAgreement.validFromForExistingCustomer		Optional: Start of validity of a user agreement for existing customer. Note, "validFrom" is used for all new clients. This field here is only used for existing customers instead.
- List <DocumentBlock> (0..*) – Optional						
BlockName	Mand.	String	50	DocumentBlock.blockName		The Name of a Document Block.
Country	Mand.	CountryCodeDT		DocumentBlock.country.countryCode		The Country of a Document Block.
DocType	Mand.	DocumentTypeEnum		DocumentBlock.docType		The DocType of a Document Block.
Locale	Mand.	LocaleDT		DocumentBlock.locale		The Locale of a Document Block.
BlockType	Mand.	DocumentBlockTypeEnum		DocumentBlock.blockType		The BlockType of a Document Block.
1. Inner List of <VersionedDocumentBlock>						
VersionId	Opt.	Integer		VersionedDocumentBlock.versionID		The VersionID of a Versioned Document Block.
xmlDefinition	Opt.	String		VersionedDocumentBlock.xmlDefinition		The XMLDefinition of a Versioned Document Block.
- List <Trigger2DocAssignment> (0..*) – Optional						
triggerType	Mand.	DocumentTriggerEnum		Trigger2DocAssignment.triggerType		The triggerType of a Trigger2DocAssignment.

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
				triggerType		
1. Inner List of <OutputTypeAssignment>						
outputType	Mand.	OutputTypeEnum		OutputTypeAssignment.outputType		The outputType of a OutputTypeAssignment.
documentID	Mand.	String	50	Document.documentElementID		The DocumentID of a Document
addRelevantUserAgreements	Opt.	Boolean		LetterOutputAssignment.addRelevantUserAgreements / EMailOutputAssignment.addRelevantUserAgreements		Optional: Flag indicating if relevant user agreements shall be added.
shortMessageNotificationId	Opt.	String	50	InputMessageOutputAssignment.shortMessageNotification.notificationId		Optional: Id of the notification.
- List <CustomTag> (0..*) – Optional						
Country	Mand.	CountryCodeDT		CustomTag.country.countryCode		The country of a Custom Tag.
DocumentType	Mand.	DocumentTypeEnum		CustomTag.documentType		The DocumentType of a Custom Tag.
TagType	Mand.	CustomTagEnum		CustomTag.tagType		The TagType of a Custom Tag.
xmlDefinition	Mand.	String		CustomTag.xmlDefinition		The XMLDefinition of a Custom Tag.
- List <DocumentImage> (0..*) – Optional						
Name	Mand.	String	50	DocumentImage.name		The name of a DocumentImage.
1. Inner List of <VersionedDocumentImage>						
versionId	Opt.	Integer		VersionedDocumentImage.versionID		The binary representation of the image that is represented by the VersionedDocumentImage.
Image	Opt.	Binary		VersionedDocumentImage.image		Version Id of the image. This version is independent of the VersionedDocument's version id.
- List <MbcCountry> (0..*) – Optional						
countryCode	Mand.	CountryCodeDT		MbcCountry.countryCode		The CountryCode of a Country.
mbConnectEnabled	Mand.	Boolean		MbcCountry.mbConnectEnabled		Flag that indicates if the country is supported by MBconnect.

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
1. Inner List of <String(50)>						
Name	Opt.	String	50	MbcCountry.name		Translation of a country's name (i18n).
- List <ServiceAuthorization> (0..*) – Optional						
Right	Mand.	AuthorizationAccessRightEnum		AccessRight.Right		The Right on a Resource..
ResourceName	Mand.	String		ResourceAuthorization.resourceName		The name of the resource.
Entitlement Name	Mand.	InternalEntitlementEnum		EntitlementRole.entitlementName		Name of an entitlement role.
- List <ShortMessageNotification> (0..*) - Optional						
NotificationId	Mand.	String	50	ShortMessageNotification.notificationId		Id of the notification.
o Inner List of <String(935)>						
NotificationText	Mand.	String	935	ShortMessageNotification.notificationText		Translation of the notification text (i18n).

Table 405: IF\_SOE\_UpdateMasterData input

### 19.2.1.3 Output

None.

### 19.2.1.4 Exceptions

None.

## 19.3 External View - Consumed Interfaces

### 19.3.1 IF\_SOEMDM\_UpdateReplicationStatus

#### 19.3.1.1 General Description

Communication type: Asynchronously

This interface is necessary for a SOE Region to inform the SOE MDM about a received package.

#### 19.3.1.2 Input

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
exportID	Mand.	Integer	5	-	Example: "11", "131", "1002"	The unique ID of an export.
replicationID	Mand..	String	5	-	Example: "EU", "CN"	The replicationID of a SOE Region.
packageID	Mand.	Integer	4	-	Example: "2", "10", "41"	The ID of a received package within an export.

Table 406: IF\_SOE\_UpdateReplicationStatus input

---

### **19.3.1.3 Output**

None.

### **19.3.1.4 Exceptions**

None.

## **19.3.2 IF\_TSB\_InformOfMasterDataChange**

### **19.3.2.1 General Description**

**Communication type:** Asynchronously

This interface is called whenever a set of master data has changed. The provider of the interface informs the adjacent systems (like McM<sub>R</sub>, McM<sub>P</sub> or CPD) of the fact that the master data has changed. This notification is enriched by an attribute determining the kind of master data. Adjacent systems can use this interface in order to update their data sets.

For the moment, the following master data types are supported. The adjacent system will need to call the associated getter interface in order to retrieve the new master data set (see chapter Informing adjacent systems about master data changes).

- <SERVICES, AUTHORIZATIONDATA>: This value informs the adjacent systems that the authorization related master data of the resource service has changed.
- <SERVICES, SERVICEMASTERDATA>: This value informs the adjacent systems that the service and/or the service master or service category related master data has changed.
- <SERVICES, SERVICECATEGORIES>: This value informs adjacent systems that service categories have changed.
- <SERVICES, VEHICLEMARKETMASTERDATA>: This value informs the adjacent systems that the market related vehicle products have changed.
- <SERVICES, VEHICLECOUNTRYMASTERDATA>: This value informs the adjacent systems that the country related vehicle products have changed.
- <MBCONNECTCOUNTRIES, MBCCOUNTRYMASTERDATA>: This value informs the adjacent systems that the MBconnect Countries have changed.
- <ACCOUNTDATASUPPORT, COUNTRYFIELDS>: This value informs the adjacent systems that the country specific fields have changed.
- <DOCUMENTS, USERAGREEMENTSERVICEASSIGNMENT>: This value informs the adjacent systems that the assignment of user agreements and services has changed.

- <DOCUMENTS; USERAGREEMENTMASTERDATA>: This value informs the adjacent systems that the user agreement related master data have changed.

### 19.3.2.2 Input

Parameter Name	Mand./ Opt.	Format	Length	Data Model	Possible Values	Annotation
MasterDataResourceTypeEnum	Mand.	MasterDataResourceTypeEnum		-	{SERVICES, MBCONNECTCOUNTRIES, ACCOUNTDATASUPPORT, DOCUMENTS}	The resource type determines for which resource the data has changed.
dataType	Mand.	MasterDataDataTypeEnum		-	{AUTHORIZATIONDATA, SERVICEMASTERDATA, SERVICECATEGORIES, SERVICEFIELDS, VEHICLEMARKETMASTERDATA, VEHICLECOUNTRYMASTERDATA, MBCCOUNTRYMASTERDATA, COUNTRYFIELDS, USERAGREEMENTSERVICEASSIGNMENT, USERAGREEMENTMASTERDATA }	The data type determines the kind of data set that has changed for the given resource type.
subSet	Opt.	List of String	-	-	-	If used, it narrows the changed master data to a given subset. If no subset is given, all instances of that resource type have changed.  Example: The subset specifies the service IDs that have been changed.

### 19.3.2.3 Output

None.

### 19.3.2.4 Exception

None.

---

## 19.4 Internal View - Offered Interfaces

### 19.4.1 IIF\_InformOfMasterDataChange

This interface is called by the components which master data is not under change session control (currently only AccountDataSupport Component) whenever the master data has changed.

Internally the AF\_InformOfMasterDataChange (→ section 19.5.2) is called.

## 19.5 Implementation

### 19.5.1 AF\_ImportMasterData

#### 19.5.1.1 General Description

This application function imports master data into a data buffer, updates the replication status and triggers the update of the master data in the replication.

#### 19.5.1.2 Sequence Description

##### **Step 1: Save the imported Master Data**

Create an <importDataBuffer> with <importDataBuffer.exportID> giving as input parameter exportID, with <importDataBuffer.startDate> giving as current date and time with <importDataBuffer.numberOfPackages> giving as input parameter numberOfPackages. For each received package create an <importDataBuffer.importPackage> with <importDataBuffer.importPackage.packageID> giving as input parameter packageID and the received data in <importDataBuffer.importPackage.data>.

##### **Step 2: Update the replication status**

For each received package, call IF\_SOEMDM\_UpdateReplicationStatus (→ see section 15.3.1) with the exportID, the number of packages (both given as input parameter), the unique replicationID of the SOE Region given as PROP\_REGIONID (→ see section 2.3.15) and with the packageID.

##### **Step 3: Update the Master Data**

If the number of packages given as input parameter is the number of received packages for one exportID, call AF\_UpdateReplicationsMasterDataBatch (→ see section 19.5.3) with the exportID as input parameter and create <importDataBuffer.endDate> giving as current date and time for the received exportID.

#### 19.5.1.3 Input

See external interface IF\_SOE\_UpdateMasterData (→ see section 19.2.1).

#### 19.5.1.4 Output

See external interface IF\_SOE\_UpdateMasterData (→ see section 19.2.1).

#### 19.5.1.5 Exceptions

See external interface IF\_SOE\_UpdateMasterData (→ see section 19.2.1).

## 19.5.2 AF\_InformOfMasterDataChange

This interface is called whenever a set of master data has changed. For a detailed description, see IF\_TSB\_InformOfMasterDataChange (see section 19.3.2).

### 19.5.2.1 Sequence Description

Call IF\_TSB\_InformOfMasterDataChange with the given parameters. No response is required.

### 19.5.2.2 Input

Parameter Name	Type / Length / BOM	Annotation
MasterDataResourceTypeEnum	MasterDataResourceTypeEnum	The resource type determines for which resource the data has changed.
dataType	MasterDataDataTypeEnum	The data type determines the kind of data set that has changed for the given resource type.
subSet	List of String	If used, it narrows the changed master data to a given subset. If no subset is given, all instances of that resource type have changed.  Example: The subset specifies the service IDs that have been changed.

Table 407: AF\_InformOfMasterDataChange input

### 19.5.2.3 Output

None.

### 19.5.2.4 Exceptions

None.

## 19.5.3 AF\_UpdateReplicationsMasterDataBatch

### 19.5.3.1 General Description

This application function updates all changed replication relevant Master Data and informs the adjacent systems about master data changes.

### 19.5.3.2 Sequence Description

#### **Step 1: Update all changed master data entities**

##### **Step 1.1: Add all new entities**

Go through Table 408 in the normal order and use ADD as parameter updateMode for each call.

##### **Step 1.2: Update all changed entities**

Go through Table 408 in the normal order and use UPDATE as parameter updateMode for each call.

##### **Step 1.3: Delete all removed entities**

Go through Table 408 in the inverse order and use DELETE as parameter updateMode for each call.

If	is inside the import data buffer (importPackage.data), call	with	and updateMode (ADD / UPDATE / DELETE)
<MbcCountry>	IIF_UpdateCountries	<MbcCountry>	
<BodyType>	IIF_UpdateBodyTypes	<BodyType>	
<ModelSeries>	IIF_UpdateModelSeries	<ModelSeries>	
<Equipment>	IIF_UpdateEquipment	<Equipment>	
<SalesType>	IIF_UpdateSalesTypes	<SalesType>	
<ServiceCategory>	AF_UpdateServiceCategories	<ServiceCategory>	
<ServiceMaster>	IIF_UpdateServiceMasters	<ServiceMaster>	
<Service>	IIF_UpdateServices	<Service>	
<ServiceAssignmentRule>	IIF_UpdateServiceAssignmentRules	<ServiceAssignmentRule>	
<GenericMasterData>	IIF_UpdateGenericMasterData	<GenericMasterData>	
<DocumentTemplate>	IIF_UpdateDocumentTemplates	<DocumentTemplate>	
<DocumentImage>	IIF_UpdateDocumentImages	<DocumentImage>	
<CustomTag>	IIF_UpdateCustomTags	<CustomTag>	
<DocumentBlock>	IIF_UpdateDocumentBlocks	<DocumentBlock>	
<Document>	IIF_UpdateDocuments	<Document>	
<UserAgreementServiceAssignment>	IIF_UpdateUserAgreementServiceAssignments	<UserAgreementServiceAssignment>	
<ServiceAuthorization>	IIF_UpdateServiceAuthorizations	<ServiceAuthorization>	
<Trigger2DocAssignment>	IIF_UpdateTrigger2DocAssignments	<Trigger2DocAssignment>	

Table 408: Order and way how to update master data

### Step 2: Informing adjacent systems about master data changes

If there are changes associated to	Call AF_InformOfMasterDataChange (→ see chapter 19.5.2) with the following input parameters		
	MasterDataResourceTypeEnum	dataType	Subset
<BodyType>, <ModelSeries>, <Equipment>, <SalesType>	SERVICES	VEHICLEMARKETMASTERDATA	-
<Service>, <ServiceAssignmentRule>	SERVICES	VEHICLECOUNTRYMASTERDATA	-
<MbcCountry>	MBCONNECTCOUNTRIES	MBCOUNTRYMASTERDATA	-
<Service>, <ServiceMaster>, <ServiceCategory>	SERIVCES	SERICEMASTERDATA	-
<ServiceCategory>	SERIVCES	SERVICECATEGORIES	-
<ServiceAuthorization>	SERVICES	AUTHORIZATIONDATA	-
<Service>, <ServiceMaster>	SERVICES	SERVICEFIELDS	-
<UserAgreementServiceAssignment>	DOCUMENTS	USERAGREEMENTSERVICEASIGNMENT	-
<Document> with <Document.isUserAgreement=true>	DOCUMENTS	USERAGREEMENTMASTERDATA	-

Table 409: Informing adjacent systems about master data changes

### Step 3: Update importDataBuffer Status

---

Set the importDataBuffer.isImported=TRUE for the exportID giving as import parameter.

### 19.5.3.3 Input

Parameter Name	Type / Length / BOM	Description
exportID	dataBuffer.exportID	The unique ID of an export.

Table 410: AF\_UpdateReplicationMasterDataBatch input

### 19.5.3.4 Output

None.

### 19.5.3.5 Exceptions

None.

## 19.5.4 AF\_ResetImportOverview

### 19.5.4.1 General Description

This application function resets the import overview in the DLG\_ReplicationImportOverview (→ see chapter 19.1.1.) and additionally deletes the import data buffer.

### 19.5.4.2 Sequence Description

Delete all <importDataBuffer> entities with <importDataBuffer.isImported> = TRUE and <importDataBuffer.endDate> smaller than two days.

### 19.5.4.3 Input

None.

### 19.5.4.4 Output

None.

### 19.5.4.5 Exceptions

None.

## 19.6 Batches

Batchname	Called Application Function	Description	Change
Batch "Update Replications Master Data"	AF_UpdateReplicationsMasterDataBatch	<u>Execution time:</u> - <u>Execution frequency:</u> - This batch starts with a successful import of master data.	New

Table 411: Batches

## 19.7 Error Messages

None.

---

## 20 Component „Outbound Messaging“

This component manages the outbound communication (e.g. letter and email based communication between Daimler AG and the end customer).

For this it manages the connections with 3rd party systems like:

- Print Server
- Mail Server

Additionally this component logs particular details regarding the data transferred between SOE and these 3rd party systems.

### 20.1 Dialogs

None.

### 20.2 External View - Offered Interfaces

None.

### 20.3 External View - Consumed Interfaces

#### 20.3.1.1 IF\_PrintServer\_GetPrintServiceReport

**Communication type:** Asynchronously

The communication is file based.

This interface will read all CSV files available at this location:

*PRINT\_SERVER\_OUTBOUND*.

The print server provider will upload information about the main processing times occurred in the process of sending a letter.

The following information needs to be provided for each processed PDF document:

- Reference number
- Timestamp when the document has been received
- Timestamp when the document has been printed
- Timestamp when the document has been delivered to the postal service
- Timestamp when the document has been processed by the postal service

The above information is written by the consumer system inside a CSV file. The delimiter used is the semi column character: ";".

It must be ensured by the print server provider that:

- the CSV files are generated with unique names. Used Naming schema:  
„ProcessedDocuments\_“+ Timestamp (when the file was generated) + „.csv“.
- For a reference number all timestamps are either filled (meaning the letters have been successfully processed) or not filled (an error encountered during processing)

*Note: the CSV-files will be stored by the consumer system at this directory path:  
PRINT\_SERVER\_OUTBOUND.*

*The files stored at this path will be read by SOE when the “Acknowledge documents sent as letters”-Batch runs.*

#### 20.3.1.1.1 Input

---

None.

### 20.3.1.2 Output

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
List of CSV files containing a list of processed letters. Each line inside a CSV file contains the following structured information:						
receivedAt	Opt.	String		CommunicationLogBook.receivedAt	e.g. '2014-01-15 8:00:00' (a date incl. at least hours, minutes and seconds.)	Timestamp when the document has been received by the print service supplier.
printedAt	Opt.	String		CommunicationLogBook.printedAt	e.g. '2014-01-15 8:00:00' (a date incl. at least hours, minutes and seconds.)	Timestamp when the document has been printed by the print service supplier.
deliveredToPostalServiceAt	Opt.	String		CommunicationLogBook.deliveredToPostalServiceAt	e.g. '2014-01-15 8:00:00' (a date incl. at least hours, minutes and seconds.)	Timestamp when the document has been handed over to the postal service
processedByPostalServiceAt	Opt.	String		CommunicationLogBook.processedByPostalServiceAt	e.g. '2014-01-15 8:00:00' (a date incl. at least hours, minutes and seconds.)	Timestamp when the document has been processed by the postal service
referenceNumber	Mand.	String[10]		CommunicationLogBook.referenceNumber		Reference number used in the customer communication process. This number is generated by SOE and is used to uniquely identify the letter that has been generated for a customer.

### 20.3.1.3 Exceptions

None.

## 20.3.1.2 IF\_PrintServer\_SendLetter

**Communication type:** Asynchronous

This interface uploads the document on the print server to be printed and eventually be sent as a letter to the customer. The print service is responsible for printing the documents from SOE, putting them into envelopes and then sending them out via letter mail. The document to be printed is uploaded at this location: PRINT\_SERVER\_INBOUND. Note: the location is created by the print server provider.

The print server provider polls X-time a day the content of the newly added documents and starts to process them.

---

*Note: the polling frequency is a technical detail and will be defined by Daimler IT within the test integration phase.*

Each PDF document for which the print service has started, will be deleted by the print server provider from the location where it was uploaded. The document will be archived by the print service provider at this location: PRINT\_SERVER\_ARCHIVE.

*Note: The processing duration of the daily workload is configurable, see PROCESSLETTER\_DURATION.*

#### 20.3.1.2.1 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
pdfLetter	Mand.	Binary PDF	-	-	-	The letter clusters all part documents (cover letter, user agreement(s) + AGB)

Table 412: IF\_Printserver\_SendLetter Input

#### 20.3.1.2.2 Output

None.

#### 20.3.1.2.3 Exceptions

None.

### 20.3.1.3 IF\_Mailserver\_SendEmail

This interface sends an email to the customer via SMTP to inform him about certain events, e.g. that new legal documents are available. Emails are sent in plain text only.

**Communication type:** Synchronous

#### 20.3.1.3.1 Input

The configured sender in the application configuration is used as sender information for the email (see PROP\_EMAIL\_CONFIG, chapter 2.3.15)

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
Recipients	Mand.	List of E-Mail-Adress es	-	-	Hans.Klein@example.tbd	List of recipients that will receive the mail
Subject	Mand.	String	-	-		The email's subject
Body	Mand.	String	-	-		The email's body.
List of Attachments	Opt.	Binary	-	-		List of attachments that are sent together with the mail. SOE only sends PDF attachments.

#### 20.3.1.3.2 Output

None.

---

(Mail is sent to recipient.)

### 20.3.1.3.3 Exceptions

None.

## 20.4 Internal View - Offered Interfaces

### 20.4.1 IIF\_SendLetter

**Communication type:** synchronous

This internal interface has the responsibility to establish the connection with the print server. Additionally it creates tracking information about the outgoing letter.

The communication details (protocol to be used, authorization and authentication) are not scope of this specification and will be defined by Daimler IT within the development phase.

Internally call AF\_SendLetter (→ section 20.5.3).

### 20.4.2 IIF\_SendEmail

This interface sends an email to the customer via SMTP to inform him about certain events, e.g. that new legal documents are available. Emails are sent in plain text only. Additionally it creates tracking information about the outgoing email.

This internal interface is implemented by AF\_SendEmail (→ section 20.5.4).

## 20.5 Implementation

### 20.5.1 AF\_UpdateSentLettersStatusBatch

This application function reads and updates the details related to processing information on the print server provider side. Additionally it logs possible errors for each letter for which no processing information is received in a certain time frame (see condition below).

#### 20.5.1.1 Sequence Description

Get all documents stored at this path PRINT\_SERVER\_OUTBOUND by calling IF\_PrintServer\_GetPrintServiceReport.

##### Step 1: Update logbook status from CSV

For each CSV-document stored execute the following:

- Match all reference numbers inside the CSV-document against the CommunicationLogBook entities and update the following attributes: receivedAt, printedAt, deliveredToPostalServiceAt, processedByPostalServiceAt based on the read values.
- Delete the CSV File from the location PRINT\_SERVER\_OUTBOUND and copy it to location PRINT\_SERVER\_CSVARCHIVE

##### Step 2: Log WARNINGS for overdue documents

Additionally, identify and log possible errors in the letter sending process. For this the following condition must be evaluated for each letter that was delivered in a defined time frame:

- Condition: the letter was delivered more than <PROCESS LETTER DURATION>

---

Days ago but not older than (PROCESS LETTER DURATION +2) Days and no matching reference number has been found inside the CSV file:

```
(CommunicationLogBook.deliveredAt + PROCESS LETTER DURATION < current timestamp) AND (CommunicationLogBook.deliveredAt + PROCESS LETTER DURATION + 2 > current timestamp) AND (CommunicationLogBook.receivedAT is EMPTY OR CommunicationLogBook.printedAT is EMPTY OR CommunicationLogBook.deliveredToPostalServiceAt is EMPTY OR CommunicationLogBook.processedByPostalServiceAt is EMPTY)
```

- If the condition is evaluated to TRUE then log the following warning in the batch log file:  
[WARN+ "for the document with the following reference number <referenceNumber> no details regarding the processing of the letter has been received"]

The log-entries will be written inside a file named after the combination of the <batch that triggered this AF> and the timestamp when this AF was called.

Note: Periodically Daimler IT will monitor this log file to ensure that the letter has been sent out to the customer successfully. When warnings are identified, the print server provider will be contacted.

#### 20.5.1.2 Input

None.

#### 20.5.1.3 Output

None.

#### 20.5.1.4 Exceptions

None.

### 20.5.2 AF\_ReportAboutMonthlyProcessedLetters

This application function creates a report containing:

- the sum of letters that has been sent in the last month by SOE
- the sum of letters that has been sent by SOE and successfully processed by the print server provider.

The information inside this report is used when monthly statistical information about the amount of outgoing letters is required.

#### 20.5.2.1 Sequence Description

Select all CommunicationLogBook entries that have been delivered in the last month and sum up these entries.

Log the information in the report:

[INFO+ "in the last month <sum> letters have been sent to the print server provider"]

From this set, sum up the entries that fulfil the condition:

- Condition: no matching reference number has been found inside the CSV file:  
(CommunicationLogBook.receivedAT is EMPTY OR CommunicationLogBook.printedAT is EMPTY OR

---

CommunicationLogBook.deliveredToPostalServiceAt is EMPTY OR  
CommunicationLogBook.processedByPostalServiceAt is EMPTY)

If the sum is greater than one, log the information in the batch log file:

[WARN+ "in the last month for <sum> letters no processing information details have been received."]

The report is named according to the combination of the <batch that triggered this AF> and the timestamp when this AF was called.

Note: Periodically Daimler IT will monitor this log file to ensure that the letter has been sent out to the customer successfully. When warnings are identified, the print server provider will be contacted.

### 20.5.2.2 Input

None.

### 20.5.2.3 Output

None.

### 20.5.2.4 Exceptions

None.

## 20.5.3 AF\_SendLetter

### 20.5.3.1 Sequence Description

The following entities are created:

- A CommunicationBackLog entity of type OutputType.LETTER
- A DocumentReference entity for each pair of document reference values

Establish the connection to the File-Server and upload the document by calling IF\_PrintServer\_SendLetter.

### 20.5.3.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
pdfLetter	Mand.	Binary PDF	-	-	-	The letter clusters all part documents (cover letter, user agreement(s) + AGB)
customerID	Mand.	UserType.UserID				ID of the customer
generatedAt	Mand.	TimestampDT				Timestamp when the document-bundle has been generated by SOE
referenceNumber	Mand.	String		CommunicationLogBook.referenceNumber		The reference number to be used by the customer in the communication with DAIMLER
List of CommunicationLogBookMetaData Pair<Key, Value> optional						
Key	Mand.	String	-	CommunicationLogBookMetaData.key	-	Defines a further business attribute that was relevant in the letter generation process
Value	Mand.	String	-	CommunicationLogBookMetaData.value	-	Defines the value of this business attribute

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of pairs &lt;String, long&gt;</b>						
documentID	String.	String				identifies the template of the document that is part of the pdf letter
version	Mand.	long				Identifies the template of the document that is part of the pdf letter

### 20.5.3.3 Output

None.

### 20.5.3.4 Exceptions

None.

## 20.5.4 AF\_SendEmail

### 20.5.4.1 Sequence Description

The following entities are created:

- A CommunicationBackLog entity of type OutputType.EMAIL
- A DocumentReference entity for each pair of document reference values
- A CommunicationLogBookMetaData entity for each pair of additional information

Send the email by calling IF\_PrintServer\_SendEmail.

### 20.5.4.2 Input

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>Recipients</b>						
Recipients	Mand.	List of E-Mail-Addreses	-	-	Hans.Klein@example.tbd	List of recipients that will receive the mail
Subject	Mand.	String	-	-		The email's subject
Body	Mand.	String	-	-		The email's body.
List of Attachments	Opt.	Binary	-	-		List of attachments that are sent together with the mail. SOE only sends PDF attachments.
customerID	Mand.	UserType.UserID				ID of the customer
generatedAt	Mand.	TimestampDT				Timestamp when the document-bundle has been generated by SOE
referenceNumber	Mand.	String		CommunicationLogBook.referenceNumber		The reference number to be used by the customer in the communication with DAIMLER
<b>List of CommunicationLogBookMetaData Pair&lt;Key, Value&gt; optional</b>						
Key	Mand.	String	-	CommunicationLogBookMetaData.key	-	Defines a further business attribute that was relevant in the letter generation process
Value	Mand.	String	-	CommunicationLogBookMetaData.value	-	Defines the value of this business attribute

Parameter Name	Mand./Opt.	Format	Length	Data Model	Possible Values	Annotation
<b>List of pairs &lt;String, long&gt;</b>						
documentID	String .	String				identifies the template of the document that is part of the pdf letter
version	Mand. .	long				Identifies the template of the document that is part of the pdf letter

#### 20.5.4.3 Output

None.

#### 20.5.4.4 Exceptions

None.

### 20.6 Batches

Batchname	Called Application Function	Description
Batch "Acknowledge documents sent as letters"	AF_UpdateSentLettersStatusBatch	Execution time: night time Execution frequency: daily
Batch "Monthly report outgoings vs. processed letters"	AF_ReportAboutMonthlyProcessedLetters	Execution time: night time (first working day of each month + PROCESS LETTER DURATION) Execution frequency: 30 day pace

Table 413: Batches (new batches)

### 20.7 Error Messages

None.

## 21 Business Object Model

## 21.1 Overview

## Type: Package «Domain»

## Package: Domain Model

The business object model shows the business objects of SOE, their attributes, and the relationships between them.

The description of the BOM elements is organized according to the functional components of the business architecture, since each domain contains semantically related elements. For each component one or more UML class diagrams illustrate the contained business objects, their relationships, and in case that it is beneficial for the understandability, also their relationships to other domains. Furthermore, each business object and its attributes are described in detail.

## **SOE Base Entities Overview**

This diagram shows the main entities of SOE and their relationships to entities that belong to different systems like CPD, GSSN/Dealer Directory or ODC.

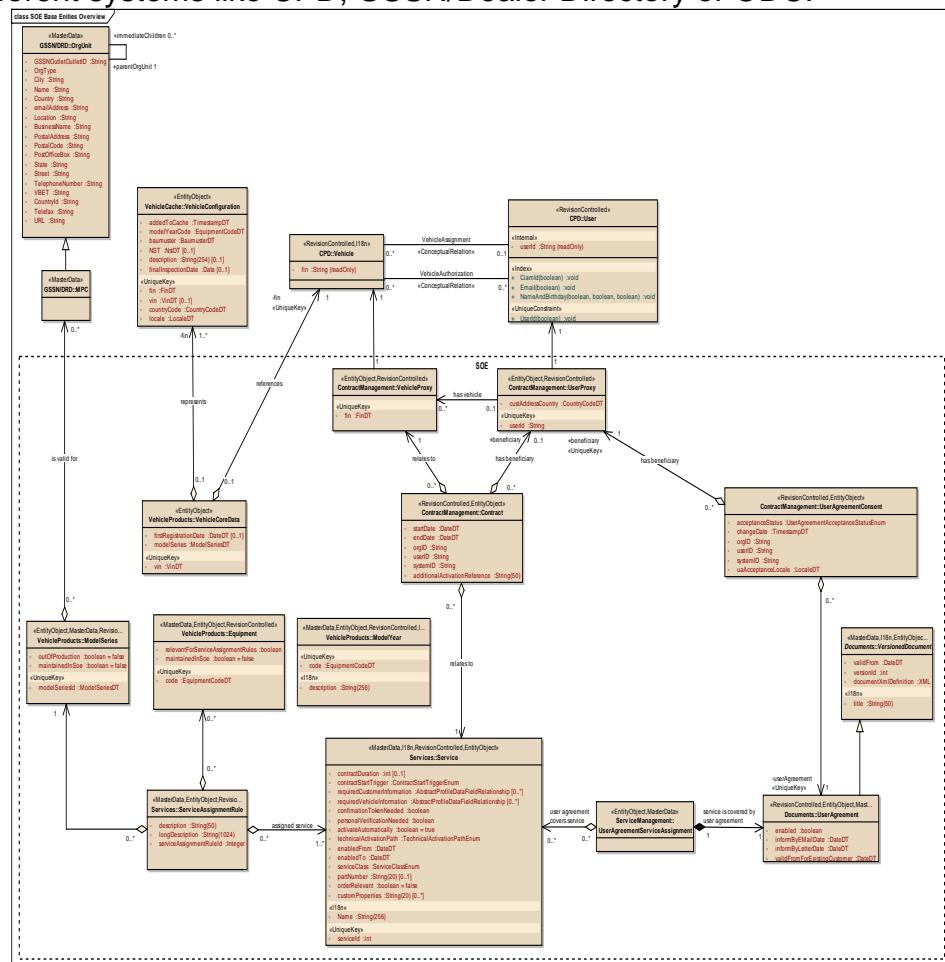


Figure 82: Overview

## SOE Overview

Overview of all functional components and their contained entities.

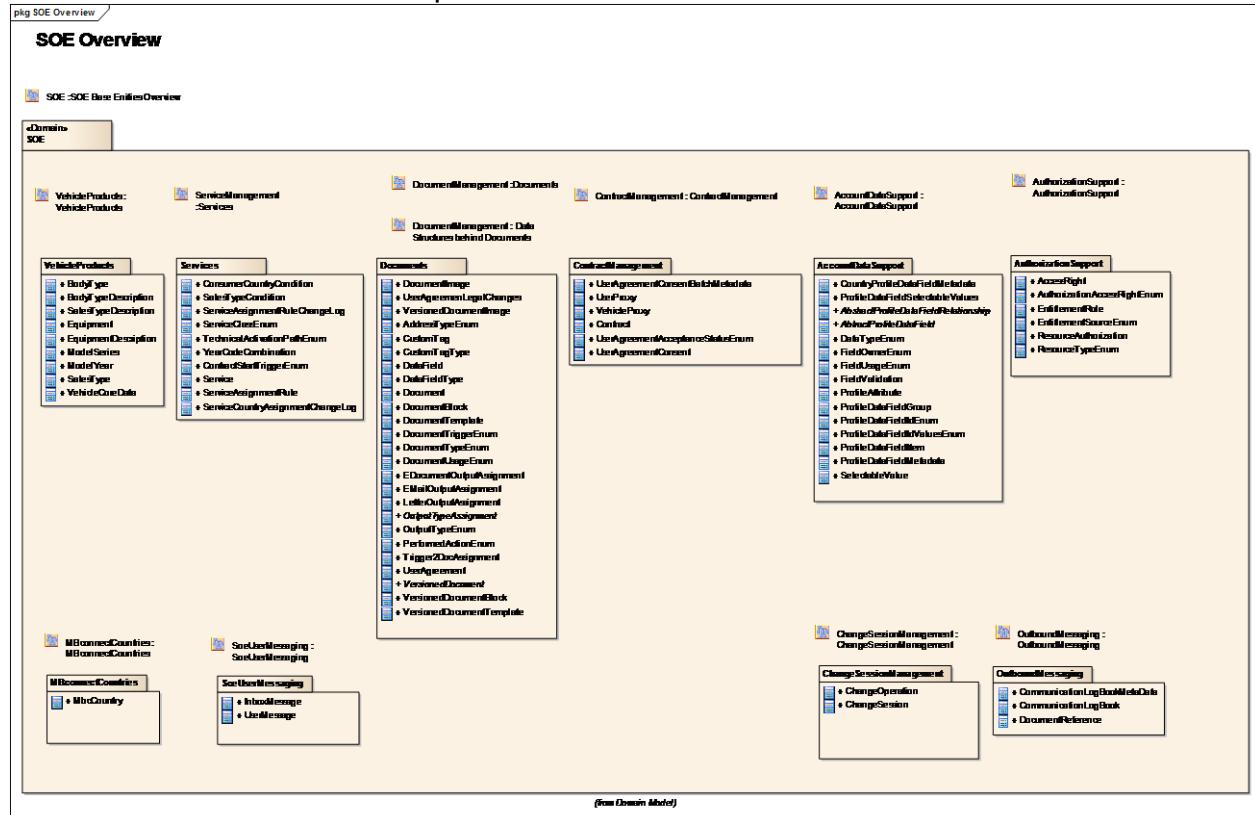


Figure 83: Overview

## **21.2 GenericMasterData**

Type: Package

Package: SOE

This component stores generic master data as key value pairs. The values are stored as binary BLOB-Objects whereas the keys are stored as strings.

Change History:

- new since [SRS036]

## GenericMasterData

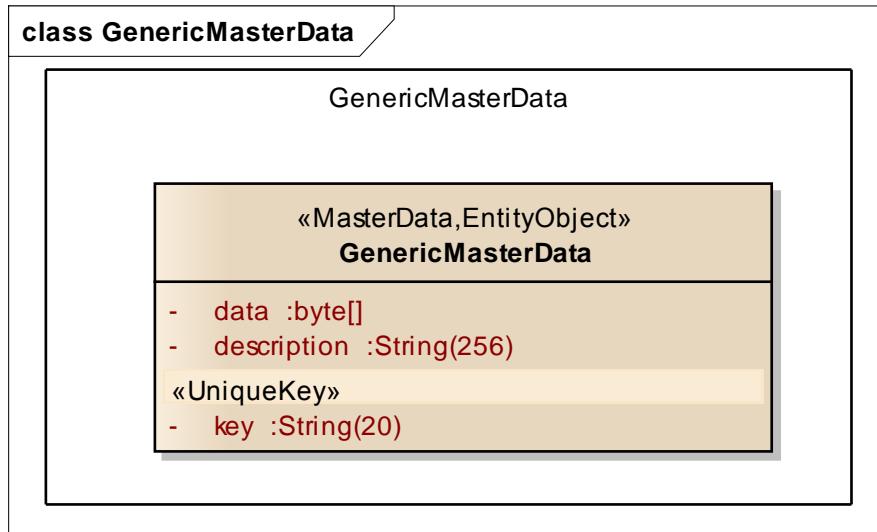


Figure 84: GenericMasterData

### 21.2.1 GenericMasterData

Type: [Class](#)

Stereotype: «MasterData,EntityObject»

Contains generic master data as key value pairs.

Change History:

- new since [SRS036]

#### Attributes

Attribute	Type	Notes
<b>key</b>	String(20) «UniqueKey»	The key of the generic master data.
<b>data</b>	byte[]	The stored data.
<b>description</b>	String(256)	Description for this generic master data.

## 21.3 MasterDataReplicationExport

Type: [Package](#)

Package: SOE

## MasterDataReplicationExport

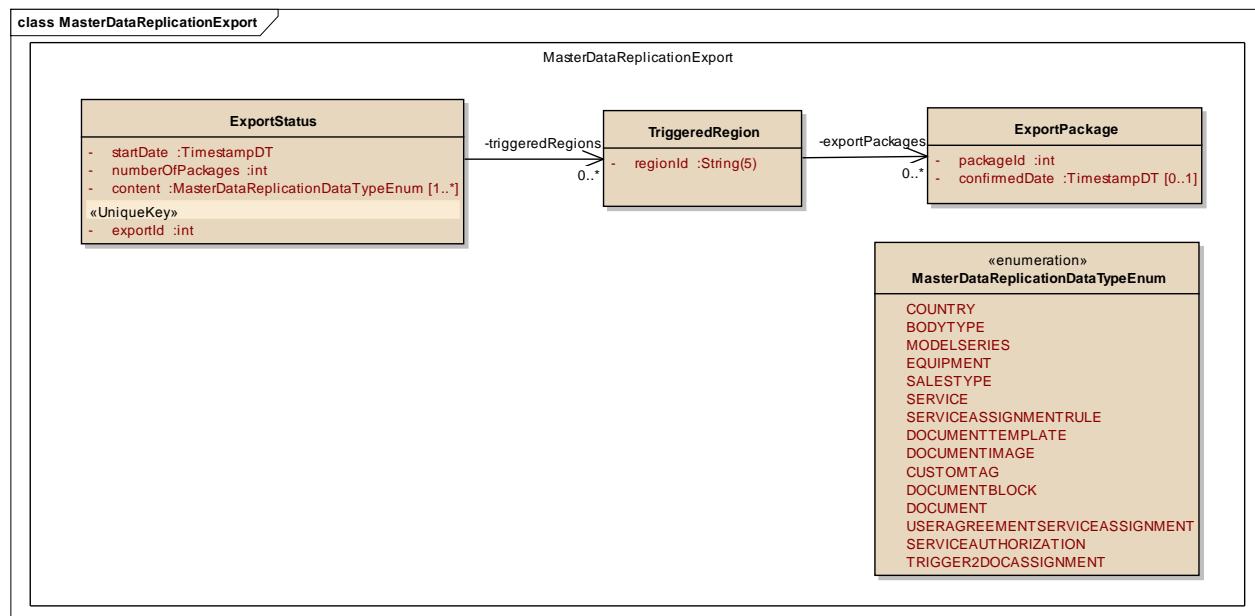


Figure 85: MasterDataReplicationExport

### 21.3.1 ExportPackage

Type: **Class**

Stereotype:

Represents a confirmed Package.

Change history:

new in [SRS035]

#### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: Type: TriggeredRegion	Role: exportPackages Cardinality: 0..* Type: ExportPackage	

#### Attributes

Attribute	Type	Notes
<b>packageId</b>	int	The package id.
<b>confirmedDate</b> [0..1]	TimestampDT	Tells when/wheather the package was confirmed.

---

### 21.3.2 ExportStatus

Type: Class

Stereotype:

Represents the status of a data replication export. Holds a list of confirmed regions, which means that those regions have sent a response message.

Change history:

new in [SRS035]

#### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: Type: ExportStatus	Role: triggeredRegions Cardinality: 0..* Type: TriggeredRegion	

#### Attributes

Attribute	Type	Notes
exportId	int «UniqueKey»	The id of the data replication export. For each export a new instance is stored.
startDate	TimestampDT	The date / time when the first package was sent.
numberOfPackages	int	The number of packages that were/are to be sent with this export.
content [1..*]	MasterDataReplication DataTypeEnum	The datatypes which were/are to be exported.

### 21.3.3 MasterDataReplicationDataTypeEnum

Type: Enumeration

Stereotype: «enumeration»

Contains all data types which can be replicated.

Change history:

new in [SRS035]

---

**Attributes**

Attribute	Type	Notes
COUNTRY	«enum»	
BODYTYPE	«enum»	
MODELSERIES	«enum»	
EQUIPMENT	«enum»	
SALESTYPE	«enum»	
SERVICE	«enum»	
SERVICEASSIGNMENT RULE	«enum»	
DOCUMENTTEMPLATE	«enum»	
DOCUMENTIMAGE	«enum»	
CUSTOMTAG	«enum»	
DOCUMENTBLOCK	«enum»	
DOCUMENT	«enum»	
USERAGREEMENTSERVICEASSIGNMENT	«enum»	
SERVICEAUTHORIZATION	«enum»	
TRIGGER2DOCASSIGNMENT	«enum»	

### 21.3.4 TriggeredRegion

*Type:* Class

*Stereotype:*

Represents a region which has sent confirmation messages.

Change history:  
new in [SRS035]

#### Connections

Connector	Source	Target	Description
<b>Association</b> Source -> Destination	Role: triggeredRegions Cardinality: 0..* Type: ExportStatus	Role: triggeredRegions Cardinality: 0..* Type: TriggeredRegion	
<b>Association</b> Source -> Destination	Role: exportPackages Cardinality: 0..* Type: TriggeredRegion	Role: exportPackages Cardinality: 0..* Type: ExportPackage	

#### Attributes

Attribute	Type	Notes
regionId	String(5)	The id of the region.

## 21.4 MasterDataReplicationImport

Type: **Package**  
Package: SOE

### MasterDataReplicationImport

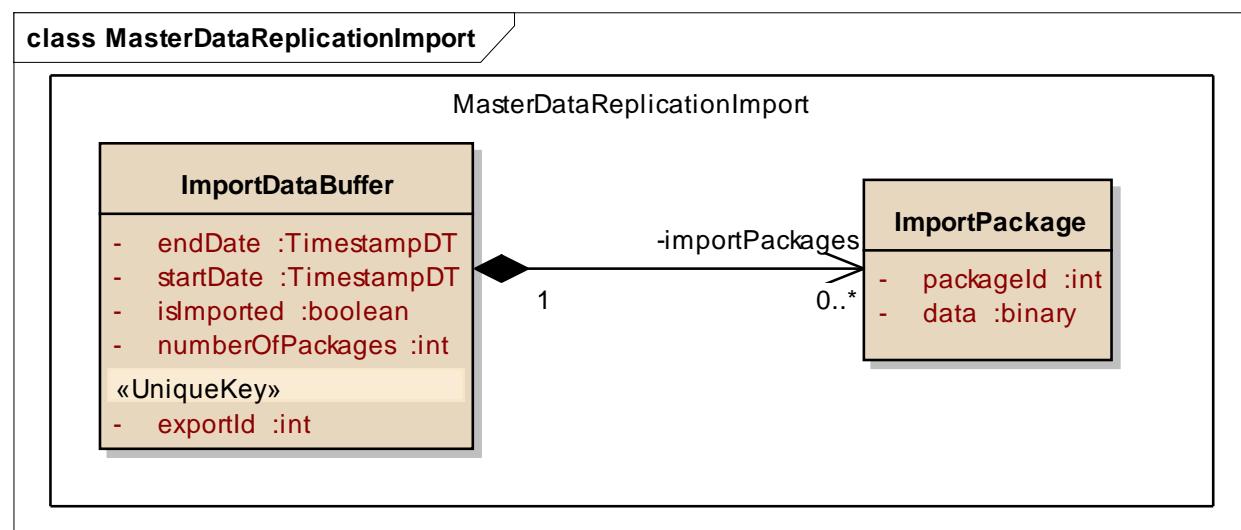


Figure 86: MasterDataReplicationImport

---

## 21.4.1 ImportDataBuffer

Type: Class

Stereotype:

Represents received data replication data from one export id. For each export id a separate element is created. The element holds the link to all data packages which have been received for this import.

Change history:

new in [SRS035]

### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1 Type: ImportDataBuffer	Role: importPackages Cardinality: 0..* Type: ImportPackage	

### Attributes

Attribute	Type	Notes
<b>endDate</b>	TimestampDT	Date when the final package was received.
<b>exportId</b>	int «UniqueKey»	Id of the export a passed in by the data replication interface.
<b>startDate</b>	TimestampDT	Date when the first package was received.
<b>isImported</b>	boolean	Tells whether the received / buffered data already has been imported by the import batch.
<b>numberOfPackages</b>	int	The expected amount of total packages.

## 21.4.2 ImportPackage

Type: Class

Stereotype:

Represents a received data package from the data replication mechanism.

Change history:

new in [SRS035]

## **Connections**

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1 Type: ImportDataBuffer	Role: importPackages Cardinality: 0..* Type: ImportPackage	

## Attributes

Attribute	Type	Notes
packageId	int	The id of the package.
data	binary	The data.

### 21.4.3 SoeUserMessaging

Type: **Package**

Package: SOE

SOE User messaging comprises the informational messages that a SOE user (i.e. a master data manager) gets.

## SoeUserMessaging

## SoeUserMessaging

Overview of the entities and relationships inside SoeUserMessaging.

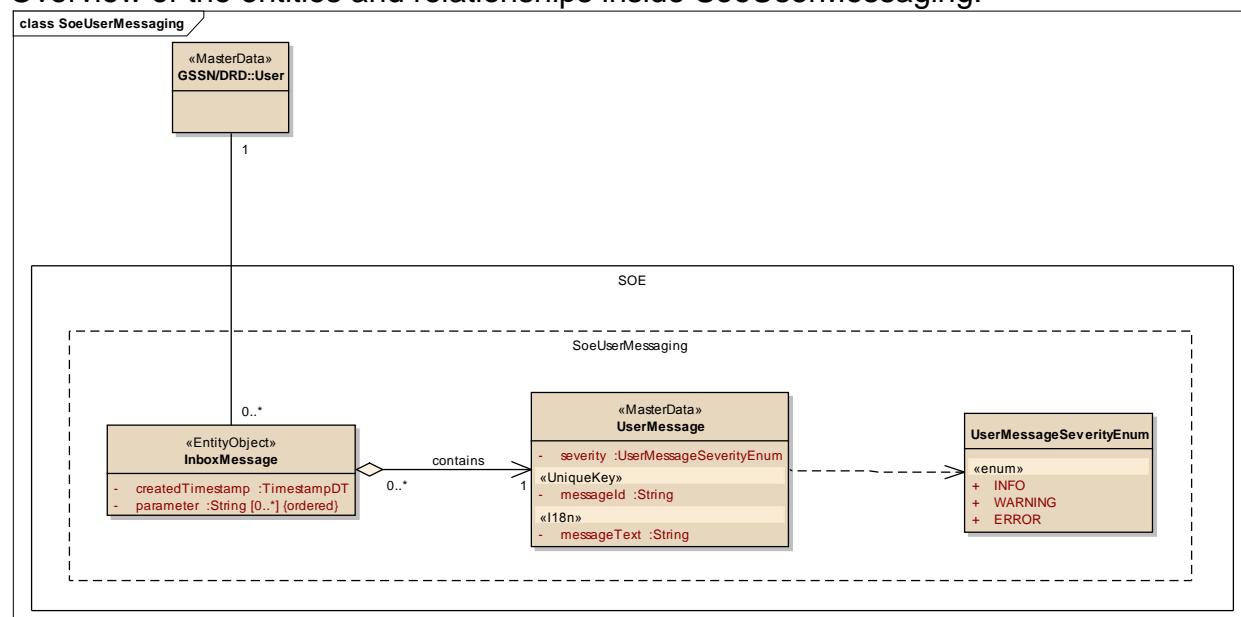


Figure 87: MasterDataReplicationImport

---

### 21.4.3.1 **InboxMessage**

Type: **Class**

Stereotype: «EntityObject»

Represents a message contained in a user's inbox. These messages are used to inform a user asynchronously about certain events that affects his work.

Change History:

- new since SRS016

#### **Connections**

Connector	Source	Target	Description
<b>Association</b> Unspecified	Role: Cardinality: 1 Type: User	Role: Cardinality: 0..* Type: InboxMessage	
<b>Association</b> contains Source -> Destination	Role: Cardinality: 0..* Type: InboxMessage	Role: Cardinality: 1 Type: UserMessage	References the UserMessage that a User's inbox contains.  Change History: <ul style="list-style-type: none"><li>• new since [SRS016]</li></ul>

#### **Attributes**

Attribute	Type	Notes
<b>createdTimestamp</b>	TimestampDT	The timestamp when a message for a user was created i.e. placed in a user's inbox.
<b>parameter</b> [0..*]	String	Parameters that are used when generating the textual message.

### 21.4.3.2 **UserMessage**

Type: **Class**

Stereotype: «MasterData»

A UserMessage entity represents a specific message. It is identified by an id and has a fixed severity assigned to it.

The descriptive message text is internationalized and may contain place holders that are replaced when displaying message to a user.

Change History:

- new since SRS016

---

### Connections

Connector	Source	Target	Description
<b>Association</b> contains Source -> Destination	Role: Cardinality: 0..* Type: InboxMessage	Role: Cardinality: 1 Type: UserMessage	References the UserMessage that a User's inbox contains.  Change History: <ul style="list-style-type: none"><li>• new since [SRS016]</li></ul>

### Attributes

Attribute	Type	Notes
<b>messageId</b>	String «UniqueKey»	The id that represents the textual message for a user. When showing the message to a user, the message text is derived from the id and displayed to the user in the appropriate language. The messages text may contain place holders for parameters.
<b>severity</b>	UserMessageSeverityEnum	The severity that belongs to a message.
<b>messageText</b>	String «I18n»	The internationalized message text for a message id. The messages text may contain place holders for parameters.

## 21.5 ChangeSessionManagement

Type:            Package  
Package:        SOE

### ChangeSessionManagement

Overview of the entities that are part of the ChangeSessionManagement. Besides of that, the entities are shown that can be referenced by a ChangeOperation. I.E. the entities that can take part in a change session.

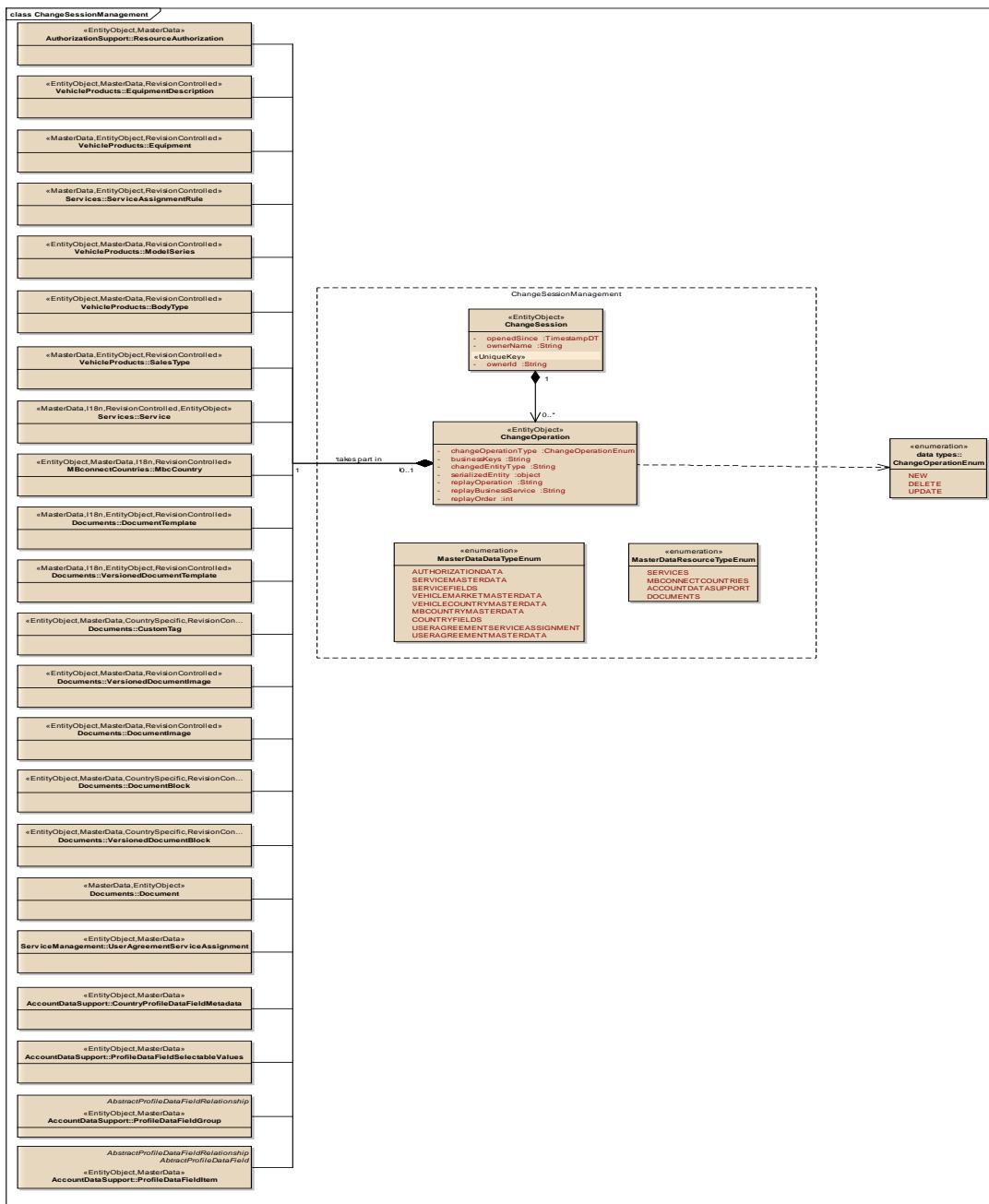


Figure 88: ChangeSessionManagement

## 21.5.1 ChangeOperation

*Type:* Class

*Stereotype:* «EntityObject»

This object represents the type of modification done on a specific master data element inside a change session. It is referenced by all master data elements that are modified inside the change session.

---

The operation type indicates what kind of modification has been done upon the master data element:

- If “NEW” then the master data element has been created inside the change session
- If “UPDATE” then the master data element has been changed inside the change session
- If “DELETE” then the master data element has been marked as deleted inside the change session

Change History:

- new since [SRS005]

#### *Connections*

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1 Type: ChangeSession	Role: Cardinality: 0..* Type: ChangeOperation	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: VersionedDocumentTemplate	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: Service	Change History: • new since
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: DocumentImage	Change History: • new since [SRS018]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: DocumentBlock	Change History: • new since [SRS005]
<u>Association</u> takes	Role:	Role:	Change History: • new since

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
part in Unspecified	Cardinality: 0..1 Type: ChangeOperation	Cardinality: 1 Type: Equipment	[SRS005]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: UserAgreementService Assignment	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: CustomTag	Change History:
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ResourceAuthorization	Change History: • new since [SRS005]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: EquipmentDescription	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: MbcCountry	Change History: • new since [SRS005]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ServiceAssignmentRule	Change History: • new since [SRS005]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ModelSeries	Change History: • new since [SRS005]

Connector	Source	Target	Description
	ChangeOperation		
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..* Type: ChangeOperation	Role: Cardinality: 1 Type: DocumentTemplate	Change History: • new since [SRS005]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: VersionedDocumentBlock	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: VersionedDocumentImage	Change History: • new since [SRS018]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ProfileDataFieldItem	Change History: • new since [SRS014]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: SalesType	Change History: • new since [SRS030]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: CountryProfileDataFieldMetadata	Change History: • new since [SRS014]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type:	Change History: • new since [SRS014]

---

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
	ChangeOperation	ProfileDataFieldSelectableValues	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ProfileDataFieldGroup	Change History: • new since [SRS014]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: BodyType	Change History: • new since [SRS030]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: Document	Change History: • new since [SRS005]

#### **Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>changeOperationType</b>	ChangeOperationEnum	
<b>businessKeys</b>	String	
<b>changedEntityType</b>	String	Type name of the changed entity.
<b>serializedEntity</b>	object	Serialized value of the DTO.
<b>replayOperation</b>	String	Operation that has to be called on the data owning component when releasing a change session.
<b>replayBusinessService</b>	String	
<b>replayOrder</b>	int	Cardinal order inside a change session. All replay operations will be called in this order when releasing a change session.

## **21.5.2 ChangeSession**

*Type:* **Class**

*Stereotype:* «EntityObject»

---

This object represents a change session.

Change History:

- new since [SRS005]

#### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1 Type: ChangeSession	Role: Cardinality: 0..* Type: ChangeOperation	

#### Attributes

Attribute	Type	Notes
<b>ownerId</b>	String «UniqueKey»	
<b>openedSince</b>	TimestampDT	The time stamp when the change session gets created.
<b>ownerName</b>	String	

## 21.6 OutboundMessaging

Type: [Package](#)  
Package: SOE

### OutboundMessaging

Overview of the entities managed by component OutboundMessaging.

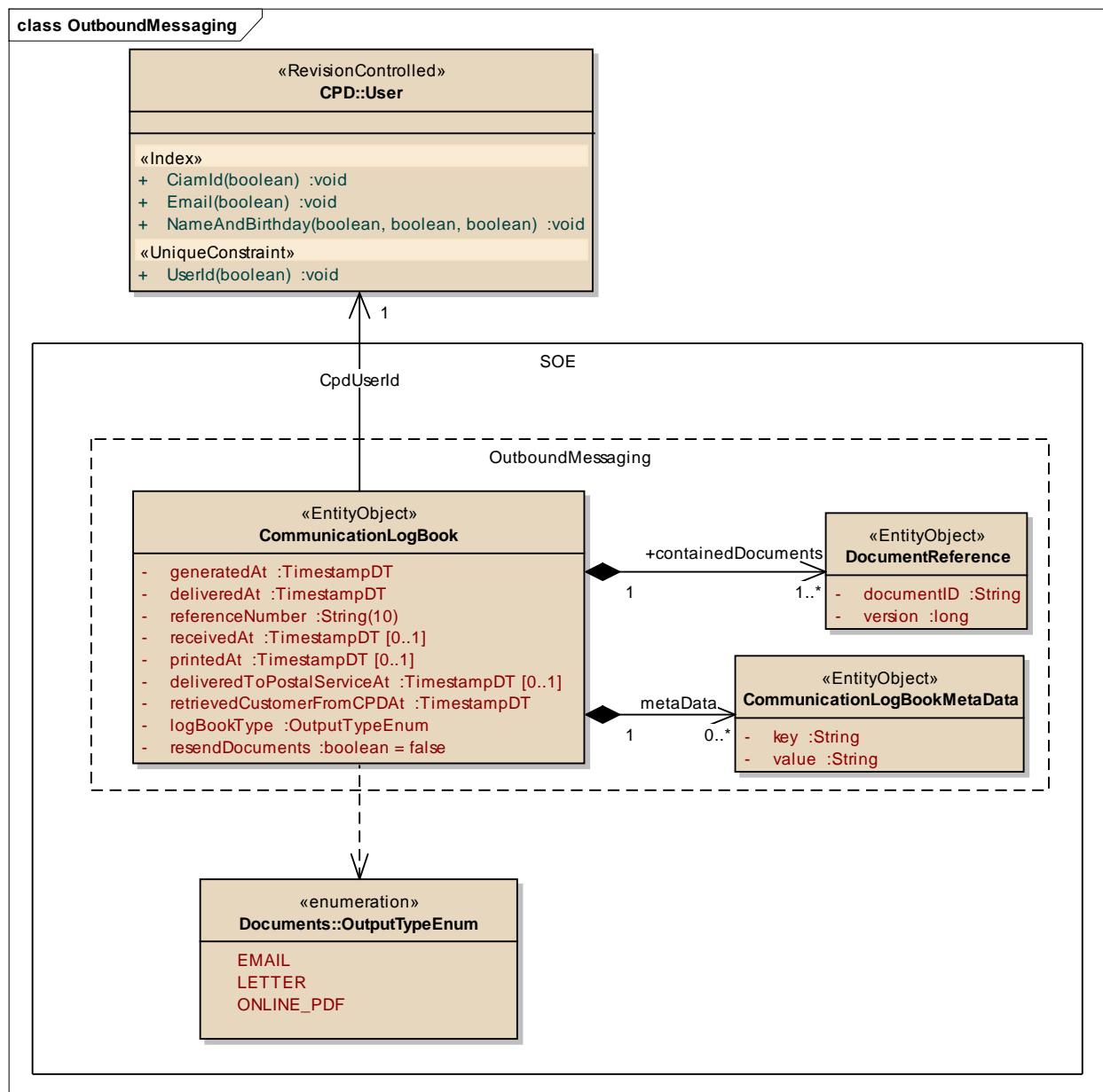


Figure 89: OutboundMessaging

## 21.6.1 CommunicationLogBook

Type: [Class](#)

Stereotype: «EntityObject»

This entity stores tracking information about the PDF Bundle that was sent as a letter to a customer.

Change History:

- New since SRS 007

### Connections

Connector	Source	Target	Description
<b>Association</b> Source -> Destination	Role: Cardinality: 1 Type: CommunicationLogBook	Role: containedDocuments Cardinality: 1..* Type: DocumentReference	
<b>Association</b> CpdUserId Source -> Destination	Role: Cardinality: Type: CommunicationLogBook	Role: Cardinality: 1 Type: User	
<b>Association</b> metaData Source -> Destination	Role: Cardinality: 1 Type: CommunicationLogBookMetaData	Role: Cardinality: 0..* Type: CommunicationLogBookMetaData	

### Attributes

Attribute	Type	Notes
<b>generatedAt</b>	TimestampDT	Timestamp when the document-bundle has been generated by SOE
<b>deliveredAt</b>	TimestampDT	Timestamp when the document-bundle has been transmitted to the print service supplier
<b>referenceNumber</b>	String(10)	Reference number used in the customer communication process. This number is generated by SOE and is internally used to uniquely identify letter that has generated for a customer.
<b>receivedAt</b> [0..1]	TimestampDT	Timestamp when the document has been received by the print service supplier
<b>printedAt</b> [0..1]	TimestampDT	Timestamp when the document has been printed by the print service supplier
<b>deliveredToPostalServiceAt</b> [0..1]	TimestampDT	Timestamp when the document has been handed over to the postal service
<b>retrievedCustomerFromCPDAt</b>	TimestampDT	timestamp when the customer Address has been fetched from CPD
<b>logBookType</b>	OutputTypeEnum	Restricted to values:

Attribute	Type	Notes
<b>resendDocuments</b>	boolean	<p>{OutputType.EMAIL, OutputType.LETTER}</p> <p>Flag that indicates if the document(s) bundled by a referenceNumber must be re-scheduled for sending. This is the case if a problem occurred during the print processing. After resending such a document this flag is reset to "false".</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS007]</li> </ul>

## 21.6.2 CommunicationLogBookMetaData

Type: **Class**

Stereotype: «EntityObject»

Stores metadata for a communication logbook entry. For instance a FIN or a TRIGGER event of the document generation. This data is needed when re-generating and - sending documents.

Change History:

New since SRS 007 v1.02

### *Connections*

Connector	Source	Target	Description
<b>Association</b> metaData Source -> Destination	Role: Cardinality: 1 Type: CommunicationLogBook	Role: Cardinality: 0..* Type: CommunicationLogBookMetaData	

### *Attributes*

Attribute	Type	Notes
<b>key</b>	String	
<b>value</b>	String	

---

### 21.6.3 DocumentReference

Type: Class

Stereotype: «EntityObject»

This entity stores references to documents that are part of a sent letter.

Change History:

- New since SRS 007

#### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1 Type: CommunicationLogBook	Role: containedDocuments Cardinality: 1..* Type: DocumentReference	

#### Attributes

Attribute	Type	Notes
documentID	String	ID of the document that is part of a letter
version	long	Version number of the sent document

## 21.7 MBconnectCountries

Type: Package

Package: SOE

### MBconnectCountries

Overview of the entities managed by component MBconnectCountries.

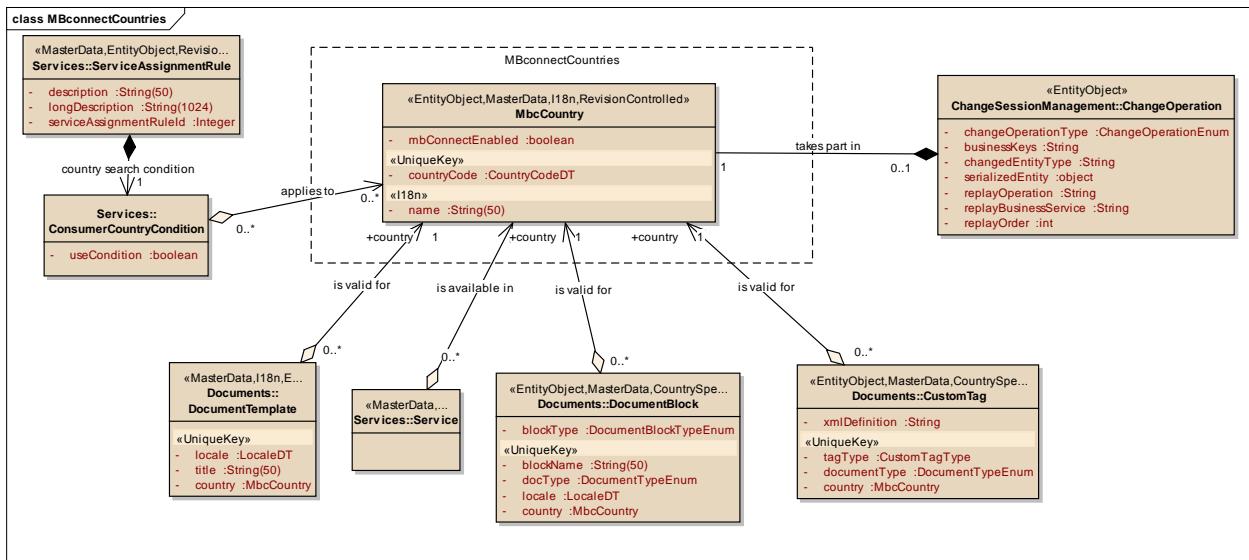


Figure 90: MBconnectCountries

## 21.7.1 MbcCountry

Type: Class

Stereotype: «EntityObject,MasterData,I18n,RevisionControlled»

Information about a country and whether it is supported by MBconnect.

Change history:

- New since [SRS001]

### Connections

Connector	Source	Target	Description
<u>Association</u> applies to Source -> Destination	Role: Cardinality: 0..* Type: <b>ConsumerCountryCondition</b>	Role: Cardinality: 0..* Type: <b>MbcCountry</b>	Defines which countries the rule applies to.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS032]</li> </ul>
<u>Association</u> is available in Source -> Destination	Role: Cardinality: 0..* Type: <b>VersionedDocument</b>	Role: countries Cardinality: 0..* Type: <b>MbcCountry</b>	Denominates the MBconnect countries for which a certain document is valid for.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS022]</li> </ul>
<u>Association</u> is available in	Role: Cardinality: 0..*	Role: Cardinality: 1..*	References the country in which a service is available.

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
Source -> Destination	Type: Service	Type: MbcCountry	Change History: <ul style="list-style-type: none"><li>new since [SRS011]</li></ul>
<u>Aggregation</u> is valid for Destination -> Source	Role: country Cardinality: 1 Type: MbcCountry	Role: Cardinality: 0..* Type: CustomTag	The country for which a CustomTag is valid.  Change history: <ul style="list-style-type: none"><li>new in [SRS001]</li></ul>
<u>Aggregation</u> is valid for Destination -> Source	Role: country Cardinality: 1 Type: MbcCountry	Role: Cardinality: 0..* Type: DocumentBlock	The country for which a DocumentBlock is valid.  Change history: <ul style="list-style-type: none"><li>new in [SRS001]</li></ul>
<u>Aggregation</u> is valid for Destination -> Source	Role: country Cardinality: 1 Type: MbcCountry	Role: Cardinality: 0..* Type: DocumentTemplate	The country for which a DocumentTemplate is valid.  Change history: <ul style="list-style-type: none"><li>new in [SRS001]</li></ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: MbcCountry	Change History: <ul style="list-style-type: none"><li>new since [SRS005]</li></ul>

#### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>countryCode</b>	CountryCodeDT «UniqueKey»	The country code of the country.
<b>mbConnectEnabled</b>	boolean	Flag that indicates if the country is supported by MBconnect.
<b>name</b>	String(50) «I18n»	Internationalized name of the country.

## 21.8 AccountDataSupport

Type: [Package](#)  
 Package: SOE

### AccountDataSupport

Overview of all entities in the functional domain AccountDataSupport.

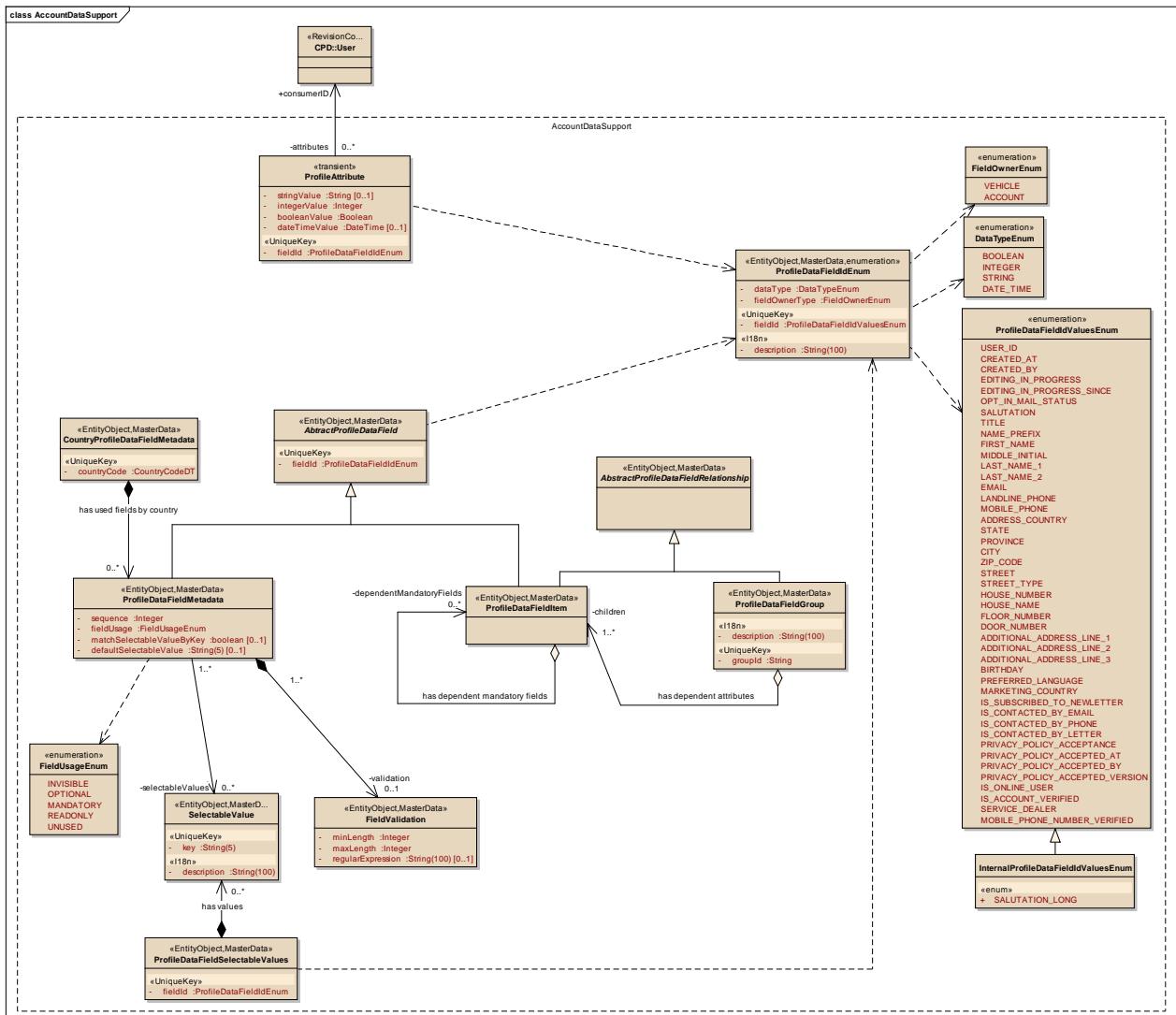


Figure 91: AccountDataSupport

## 21.8.1 AbstractProfileDataFieldRelationship Class

Type:

Class

Stereotype: «EntityObject,MasterData»

Abstract base class for building dependency structures with profile data fields.

Change History:

- New since [SRS003]

### Connections

Connector	Source	Target	Description
<b>Generalization</b>	Role:	Role:	

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
Source -> Destination	Cardinality: Type: ProfileDataFieldItem	Cardinality: Type: AbstractProfileDataFieldRelationship	
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: ProfileDataFieldGroup	Role: Cardinality: Type: AbstractProfileDataFieldRelationship	
<u>Association</u> has required customer fields Source -> Destination	Role: Cardinality: 0..* Type: Service	Role: requiredCustomerInformation Cardinality: 0..* Type: AbstractProfileDataFieldRelationship	<b>Mandatory fields for a service.</b>
<u>Association</u> has required vehicle fields Source -> Destination	Role: Cardinality: 0..* Type: Service	Role: requiredVehicleInformation Cardinality: 0..* Type: AbstractProfileDataFieldRelationship	<b>Mandatory fields for a service.</b>

## 21.8.2 AbstractProfileDataField

Type: **Class**

Stereotype: «EntityObject,MasterData»

Abstract base class for profile data fields.

Change History:

- new since [SRS003]

### Connections

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type:	Role: Cardinality: Type:	

Connector	Source	Target	Description
	ProfileDataFieldMetadata	AbstractProfileDataField	
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: ProfileDataFieldItem	Role: Cardinality: Type: AbstractProfileDataField	

#### Attributes

Attribute	Type	Notes
<b>fieldId</b>	ProfileDataFieldIdEnum «UniqueKey»	The ID of the field.  Change History: <ul style="list-style-type: none"><li>• [SRS003]: moved from CustomerDataField</li></ul>

### 21.8.3 CountryProfileDataFieldMetadata

Type: Class

Stereotype: «EntityObject,MasterData»

Groups all ProfileDataFieldMetadata that belongs to a certain country.

ChangeHistory:

- new since [SRS014]

#### Connections

Connector	Source	Target	Description
<u>Association</u> has used fields by country Source -> Destination	Role: Cardinality: Type: CountryProfileDataFieldMetadata	Role: Cardinality: 0..* Type: ProfileDataFieldMetadata	References the fields used by a certain country.  Change History: <ul style="list-style-type: none"><li>• new since [SRS014]</li></ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: CountryProfileDataField Metadata	Change History: <ul style="list-style-type: none"><li>• new since [SRS014]</li></ul>

---

**Attributes**

Attribute	Type	Notes
<b>countryCode</b>	CountryCodeDT «UniqueKey»	ISO code if the country for which this rule applies.

## 21.8.4      **DataTypeEnum**

*Type:*      **Enumeration**

*Stereotype:*    «enumeration»

**Attributes**

Attribute	Type	Notes
<b>BOOLEAN</b>	«enum»	
<b>INTEGER</b>	«enum»	
<b>STRING</b>	«enum»	
<b>DATE_TIME</b>	«enum»	

## 21.8.5      **FieldOwnerEnum**

*Type:*      **Enumeration**

*Stereotype:*    «enumeration»

Describes possible values for the ownership of a field.

Change History:

- new in [SRS003]

**Attributes**

Attribute	Type	Notes
<b>VEHICLE</b>	«enum»	Value if a field relates to a vehicle
<b>ACCOUNT</b>	«enum»	Value if the fields relates to an account/customer profile field.

---

## 21.8.6 FieldUsageEnum

Type: Enumeration

Stereotype: «enumeration»

Enumeration type which determines how a field is used.

### Attributes

Attribute	Type	Notes
INVISIBLE	«enum»	Field not needed for the requested country
OPTIONAL	«enum»	Field needed, but does not have to be filled
MANDATORY	«enum»	Field needed and must be provided
READONLY	«enum»	When a field is visible but must not be editable by the user.
UNUSED	«enum»	A field marked as "unused" is not utilized in a country. I.e. that field generally exists in SOE, but will not be provided to a calling system as a country field. Nevertheless the field exists and it would be possible for a master data manager to change that state if a country's requirement will change.  Change History: <ul style="list-style-type: none"><li>• New since [SRS014]</li></ul>

## 21.8.7 FieldValidation

Type: Class

Stereotype: «EntityObject,MasterData»

Through a FieldValidation it is possible to define rules for the valid content of a CustomerDataField.

Change History:

- 
- [SRS003]: removed attribute "dependencyCheck"

#### *Connections*

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1..* Type: ProfileDataFieldMetadata	Role: validation Cardinality: 0..1 Type: FieldValidation	

#### *Attributes*

Attribute	Type	Notes
<b>minLength</b>	Integer	Determines the minimum length of the field.
<b>maxLength</b>	Integer	Maximum length of the provided value. Can be used to limit the maximum input length of a text box.
<b>regularExpression</b> [0..1]	String(100)	Optionally provides a regular expression to validate against. The regular expression is supported through the standard java library for regular expressions (see java.util.regex package).

## 21.8.8 ProfileAttribute

Type: Class

Stereotype: «transient»

This entity encapsulates an attribute of the UserType.

Change History:

- Renamed from "CutomerAttribute" to "ProfileAttribute" [SRS003]

#### *Connections*

Connector	Source	Target	Description
<u>Association</u> Destination -> Source	Role: consumerID Cardinality: Type: User	Role: attributes Cardinality: 0..* Type: ProfileAttribute	

---

#### Attributes

Attribute	Type	Notes
<b>fieldId</b>	ProfileDataFieldIdEnum «UniqueKey»	Id of the field to which is referenced.
<b>stringValue</b> [0..1]	String	The value if dataType is set to STRING.
<b>integerValue</b>	Integer	The value if dataType is set to INTEGER.
<b>booleanValue</b>	Boolean	The value if dataType is set to BOOLEAN.
<b>dateTimeValue</b> [0..1]	DateTime	The value if dataType is set to DATE_TIME.

## 21.8.9 ProfileDataFieldGroup

Type: Class AbstractProfileDataFieldRelationship

Stereotype: «EntityObject,MasterData»

This entity groups profile data fields. The semantics is that at least one of the referenced ProfileDataFields must not be empty.

Note: It is not possible to nest ProfileDataFieldGroups within other ProfileDataFieldGroups. This is an intentional restriction to simplify the handling.

Change History:

- New since [SRS003]

#### Connections

Connector	Source	Target	Description
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: ProfileDataFieldGroup	Role: Cardinality: Type: AbstractProfileDataFieldRelationship	
<u>Association</u> has dependent attributes Source -> Destination	Role: Cardinality: Type: ProfileDataFieldGroup	Role: children Cardinality: 1..* Type: ProfileDataFieldItem	The semantics is that at least one of the referenced ProfileDataFields must not be empty.  Change History: <ul style="list-style-type: none"><li>• New since [SRS003]</li></ul>
<u>Association</u> takes part in	Role: Cardinality: 0..1	Role: Cardinality: 1	Change History: <ul style="list-style-type: none"><li>• new since</li></ul>

---

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
Unspecified	Type: ChangeOperation	Type: ProfileDataFieldGroup	[SRS014]

**Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>description</b>	String(100) «I18n»	<p>Internationalized name for a ProfileDataFieldGroup which is used in the SOE-UI only. Therefore the i18n value does not need to be translated into all MBconnect markets. This value can be maintained by the SOE-UI.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• [SRS014] moved the description from AbstractProfileDataFieldRelationship to ProfileDataFieldGroup in order to show that only for groups a description can be maintained by the SOE-UI in contrast to the description of a ProfileDataFieldId</li> </ul>
<b>groupId</b>	String «UniqueKey»	Group id that is used to identify a group. (It's the functional key for a group)

### 21.8.10    **ProfileDataFieldIdEnum**

*Type:*    **Enumeration**

*Stereotype:*    «EntityObject,MasterData,enumeration»

This data type brings together a fieldId with its data type and field owner type. The relationship is steady and constant, i.e. a given fieldId has always a defined type and a defined field owner type. That cannot (and does not need to be) changed dynamically.

Change History:

- new since [SRS003]

**Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>fieldId</b>	ProfileDataFieldIdValue sEnum «UniqueKey»	
<b>dataType</b>	DataTypeEnum	Enum that represents the data type for the fieldId. Depending on this value either integerValue, stringValue, booleanValue or dateStringValue are filled and valid.
<b>description</b>	String(100) «I18n»	<p>Internationalized name for a ProfileDataFieldEnum which is used in the SOE-UI only. Therefore the i18n value does not need to be translated into all MBconnect markets and does not need to be stored in the database. It is sufficient that the description is stored in resource bundles for the SOE-UI in the languages supported by the system.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• [SRS014] moved the description from AbstractProfileDataFieldRelationship to ProfileDataFieldIdEnum in order to show that only for groups a description can be maintained by the SOE-UI in contrast to the description of a ProfileDataFieldId which cannot be changed through the system's UI.</li> </ul>
<b>fieldOwnerType</b>	FieldOwnerEnum	<p>Determines the type to which this field relates to.</p> <p>Change History</p> <ul style="list-style-type: none"> <li>• new since [SRS003]</li> </ul>

## 21.8.11 ProfileDataFieldIdValuesEnum

*Type:* Enumeration

*Stereotype:* «enumeration»

---

Enum with all possible fields that can occur in a UserType.  
(Note: Not all fields must be used in all markets.)

#### Change History

- renamed from "CustomerDataFieldIdEnum" zu "ProfileDataFieldIdValuesEnum"  
[SRS003]

#### Connections

Connector	Source	Target	Description
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: InternalProfileDataFieldIdValuesEnum	Role: Cardinality: Type: ProfileDataFieldIdValue sEnum	

#### Attributes

Attribute	Type	Notes
USER_ID	«enum»	
CREATED_AT	«enum»	
CREATED_BY	«enum»	
EDITING_IN_PROGRESS	«enum»	
EDITING_IN_PROGRESS_SINCE	«enum»	
OPT_IN_MAIL_STATUS	«enum»	
SALUTATION	«enum»	
TITLE	«enum»	
NAME_PREFIX	«enum»	
FIRST_NAME	«enum»	
MIDDLE_INITIAL	«enum»	
LAST_NAME_1	«enum»	
LAST_NAME_2	«enum»	

---

Attribute	Type	Notes
<b>EMAIL</b>	«enum»	
<b>LANDLINE_PHONE</b>	«enum»	
<b>MOBILE_PHONE</b>	«enum»	
<b>ADDRESS_COUNTRY</b>	«enum»	
<b>STATE</b>	«enum»	
<b>PROVINCE</b>	«enum»	
<b>CITY</b>	«enum»	
<b>ZIP_CODE</b>	«enum»	
<b>STREET</b>	«enum»	
<b>STREET_TYPE</b>	«enum»	
<b>HOUSE_NUMBER</b>	«enum»	
<b>HOUSE_NAME</b>	«enum»	
<b>FLOOR_NUMBER</b>	«enum»	
<b>DOOR_NUMBER</b>	«enum»	
<b>ADDITIO-NAL_ADDRESS_LINE_1</b>	«enum»	
<b>ADDITIO-NAL_ADDRESS_LINE_2</b>	«enum»	
<b>ADDITIO-NAL_ADDRESS_LINE_3</b>	«enum»	
<b>BIRTHDAY</b>	«enum»	
<b>PREFERRED_LANGUAGE</b>	«enum»	
<b>MARKE-TING_COUNTRY</b>	«enum»	

Attribute	Type	Notes
<b>IS_SUBSCRIBED_TO_NEWSLETTER</b>	«enum»	
<b>IS_CONTACTED_BY_EMAIL</b>	«enum»	
<b>IS_CONTACTED_BY_PHONE</b>	«enum»	
<b>IS_CONTACTED_BY LETTER</b>	«enum»	
<b>PRIVACY_POLICY_ACCEPTANCE</b>	«enum»	
<b>PRIVACY_POLICY_ACCEPTED_AT</b>	«enum»	
<b>PRIVACY_POLICY_ACCEPTED_BY</b>	«enum»	
<b>PRIVACY_POLICY_ACCEPTED_VERSION</b>	«enum»	
<b>IS_ONLINE_USER</b>	«enum»	<p>"TRUE" if the user is a regular online user, not a "Heinz".</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• changed from "IS_OFFLINE_USER" to "IS_ONLINE_USER" [SRS003]</li> </ul>
<b>IS_ACCOUNT_VERIFIED</b>	«enum»	<p>Flag to indicate if a user account has been verified or not.</p> <p>Change History</p> <ul style="list-style-type: none"> <li>• new in [SRS003]</li> </ul>
<b>SERVICE DEALER</b>	«enum»	<p>Information about the preferred service dealer.</p> <p>Change History</p> <ul style="list-style-type: none"> <li>* new in [SRS003]</li> </ul>
<b>MOBI-</b>		The mobile number has to have

Attribute	Type	Notes
<b>LE_PHONE_NUMBER _VERIFIED</b>	«enum»	<p>been verified.</p> <p>Change History</p> <ul style="list-style-type: none"> <li>• added in [SRS038]</li> </ul>

### 21.8.12 ProfileDataFieldItem

Type: **Class AbstractProfileDataFieldRelationship, AbstractProfileDataField**

Stereotype: «EntityObject,MasterData»

Specialized entity for referencing ProfileData Fields. This entity can be used to build structures of fields that reference each other.

Change History:

- new since [SRS003]

#### Connections

Connector	Source	Target	Description
<b>Generalization</b> Source -> Destination	Role: Cardinality: Type: ProfileDataFieldItem	Role: Cardinality: Type: AbstractProfileDataFieldRelationship	
<b>Generalization</b> Source -> Destination	Role: Cardinality: Type: ProfileDataFieldItem	Role: Cardinality: Type: AbstractProfileDataField	
<b>Association</b> has dependent attributes Source -> Destination	Role: Cardinality: Type: ProfileDataFieldGroup	Role: children Cardinality: 1..* Type: ProfileDataFieldItem	The semantics is that at least one of the referenced ProfileDataFields must not be empty.  Change History: <ul style="list-style-type: none"> <li>• New since [SRS003]</li> </ul>
<b>Aggregation</b> has dependent mandatory fields Destination -> Source	Role: dependentMandatoryFields Cardinality: 0..* Type: ProfileDataFieldItem	Role: Cardinality: Type: ProfileDataFieldItem	If given, this attribute states a dependency to another field. It says that if that at least one of the referenced other field has to be filled (or checked for boolean fields), if

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
			<p>this field is filled. Example: The field "email" would point to the field "LAND-LINE_PHONE" and "MOBILE_PHONE". The result is, that if the "EMAIL" field contains a value, either "MOBILE_PHONE" or "LAND-LINE_PHONE" (or both) must also contain a value.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS003]</li> </ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ProfileDataFieldItem	Change History: <ul style="list-style-type: none"> <li>• new since [SRS014]</li> </ul>

### 21.8.13 ProfileDataFieldMetadata

Type: Class AbstractProfileDataField

Stereotype: «EntityObject,MasterData»

Entity used to describe a rule for a ProfileDataField. This data structure is used to exchange the allowed formats with UIs.

Change History:

- Renamed from "CustomerDataField" to "ProfileDataFieldMetadata" [SRS003]
- [SRS014] moved countryCode to CountryProfileDataFieldMetadata

#### Connections

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: ProfileDataFieldMetadata	Role: Cardinality: Type: AbstractProfileDataField	
<u>Association</u> Source -> Destination	Role: Cardinality: 1..* Type: SelectableValue	Role: selectableValues Cardinality: 0..* Type: SelectableValue	

Connector	Source	Target	Description
	ProfileDataFieldMetadata		
<u>Association</u> Source -> Destination	Role: Cardinality: 1..* Type: ProfileDataFieldMetadata	Role: validation Cardinality: 0..1 Type: FieldValidation	
<u>Association</u> has used fields by country Source -> Destination	Role: Cardinality: Type: CountryProfileDataFieldMetadata	Role: Cardinality: 0..* Type: ProfileDataFieldMetadata	References the fields used by a certain country.  Change History: <ul style="list-style-type: none"><li>• new since [SRS014]</li></ul>

#### Attributes

Attribute	Type	Notes
<b>sequence</b>	Integer	The sorting sequence of the field intended to be used by the UI.
<b>fieldUsage</b>	FieldUsageEnum	INVISIBLE: Field not needed for the requested country OPTIONAL: Field needed, but does not have to be filled MANDATORY: Field needed and must be provided
<b>matchSelectableValueByKey</b> [0..1]	boolean	Is filled if this CustomerFieldList.Field is an enumerated field (=offers a list of selectable values). It states whether the consuming system must use the key (true) or the description (false) of the enumerated value when mapping the data to the consumer object for CPD.
<b>defaultSelectableValue</b> [0..1]	String(5)	Is filled if this CustomerFieldList.Field is an enumerated field (=offers a list of selectable values). It states which of the selectable values shall be used as default (refers to the code of the respective element). If empty, there is no default.

---

## 21.8.14 ProfileDataFieldSelectableValues

Type: Class

Stereotype: «EntityObject,MasterData»

Groups all SelectableValues that are used/configured for a ProfileDataField. This happens independent of country or language.

Change History:

- new since [SRS014]

### Connections

Connector	Source	Target	Description
<u>Association</u> has values Source -> Destination	Role: Cardinality: Type: ProfileDataFieldSelectableValues	Role: Cardinality: 0..* Type: SelectableValue	References the set of selectable values that are possible for a ProfileDataFieldId.  Change History: <ul style="list-style-type: none"><li>• new since [SRS014]</li></ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ProfileDataFieldSelectableValues	Change History: <ul style="list-style-type: none"><li>• new since [SRS014]</li></ul>

### Attributes

Attribute	Type	Notes
fieldId	ProfileDataFieldIdEnum «UniqueKey»	The ID of the field.  Change History: <ul style="list-style-type: none"><li>• [SRS003]: moved from CustomerDataField</li></ul>

## 21.8.15 SelectableValue

Type: Class

Stereotype: «EntityObject,MasterData»

Through selectable values it is possible to restrict fields to enumeration values. Inside a selectable value the attribute "key" determines the enumeration key where the "description" provides the human readable textual form in a certain locale.

---

#### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1..* Type: ProfileDataFieldMetadata	Role: selectableValues Cardinality: 0..* Type: SelectableValue	
<u>Association</u> has values Source -> Destination	Role: Cardinality: Type: ProfileDataFieldSelectableValues	Role: Cardinality: 0..* Type: SelectableValue	References the set of selectable values that are possible for a ProfileDataFieldId.  Change History: <ul style="list-style-type: none"><li>• new since [SRS014]</li></ul>

#### Attributes

Attribute	Type	Notes
<b>key</b>	String(5) «UniqueKey»	
<b>description</b>	String(100) «I18n»	Human readable name for a selectable value that is used by a UI to fill drop-down-lists.

## 21.9 AuthorizationSupport

Type: Package  
Package: SOE

### AuthorizationSupport

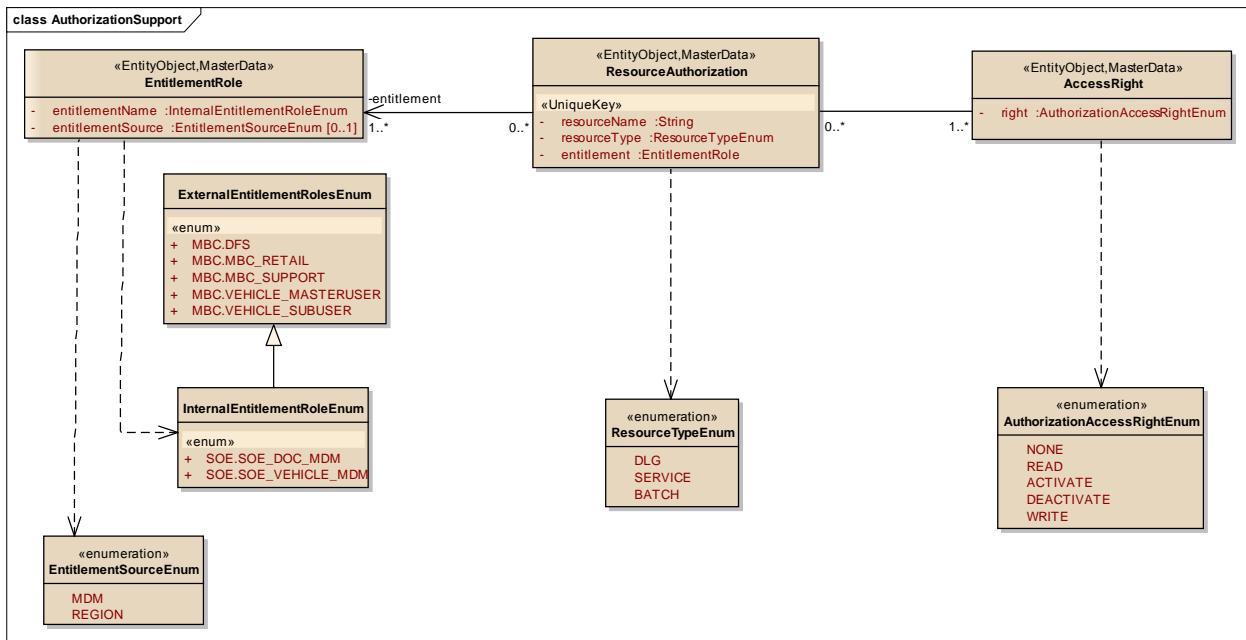


Figure 92: AuthorizationSupport

## 21.9.1 AccessRight

**Type:** Class

**Stereotype:** «EntityObject,MasterData»

Access right on a resource.

### Connections

Connector	Source	Target	Description
<u>Association</u> Unspecified	Role: Cardinality: 0..* Type: <code>ResourceAuthorization</code>	Role: Cardinality: 1..* Type: <code>AccessRight</code>	

### Attributes

Attribute	Type	Notes
<code>right</code>	<code>AuthorizationAccessRightEnum</code>	Right on a resource.

---

## 21.9.2 AuthorizationAccessRightEnum

Type: Enumeration

Stereotype: «enumeration»

Enumeration with the set of all possible rights known to SOE. The rights are ordered by priority, i.e. if somebody has more than one right on a resource, the highest ranking role prevails.

"WRITE" is the highest role, "NONE" is the lowest.

### Attributes

Attribute	Type	Notes
<b>NONE</b>	«enum»	Stands for "no rights". The entitlement role has no right on that resource, i.e. the resource must not be visible to him.
<b>READ</b>	«enum»	Read rights for the resource.
<b>ACTIVATE</b>	«enum»	Right to "activate" something on a resource. (It's a specialized form of the "write" right.)  Change History: <ul style="list-style-type: none"><li>• new since [SRS002]</li></ul>
<b>DEACTIVATE</b>	«enum»	Right to "deactivate" something on a resource. (It's a specialized form of the "write" right.)  Change History: <ul style="list-style-type: none"><li>• new since [SRS002]</li></ul>
<b>WRITE</b>	«enum»	Right to read and change data on a resource.

## 21.9.3 EntitlementRole

Type: Class

Stereotype: «EntityObject,MasterData»

Entitlement roles known to SOE.

---

#### *Connections*

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<b>Association</b> Destination -> Source	Role: entitlement Cardinality: 1..* Type: EntitlementRole	Role: Cardinality: 0..* Type: ResourceAuthorization	The entitlement role which the access right is assigned for.

#### *Attributes*

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>entitlementName</b>	InternalEntitlementRoleEnum	Name of an entitlement role.  Change History: <ul style="list-style-type: none"><li>[SRS010] change type from String to InternalEntitlementRoleEnum</li></ul>
<b>entitlementSource [0..1]</b>	EntitlementSourceEnum	Tells whether the entitlement role applies to MDM or REGION.  Change history: new in [SRS035]

#### **21.9.4 EntitlementSourceEnum**

*Type:* Enumeration

*Stereotype:* «enumeration»

Tells whether the entitlement role applies to MDM or REGION.

Change history:  
new in [SRS035]

#### *Attributes*

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>MDM</b>	«enum»	The central SOE instance.
<b>REGION</b>	«enum»	A regional SOE instance.

---

## 21.9.5 ResourceAuthorization

Type: Class

Stereotype: «EntityObject,MasterData»

Resources known to SOE that are entitled.

### Connections

Connector	Source	Target	Description
<u>Association</u> Unspecified	Role: Cardinality: 0..* Type: ResourceAuthorization	Role: Cardinality: 1..* Type: AccessRight	
<u>Association</u> Destination -> Source	Role: entitlement Cardinality: 1..* Type: EntitlementRole	Role: Cardinality: 0..* Type: ResourceAuthorization	The entitlement role which the access right is assigned for.
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ResourceAuthorization	Change History: • new since [SRS005]

### Attributes

Attribute	Type	Notes
resourceName	String «UniqueKey»	Name of the resource.
resourceType	ResourceTypeEnum «UniqueKey»	Type of the entitled resource.  Change History • new since [SRS002]
entitlement	EntitlementRole «UniqueKey»	The entitlement role which the access right is assigned for.

## 21.9.6 ResourceTypeEnum

Type: Enumeration

Stereotype: «enumeration»

Possible resource types known by SOE.

Change History

- new since [SRS002]

## Attributes

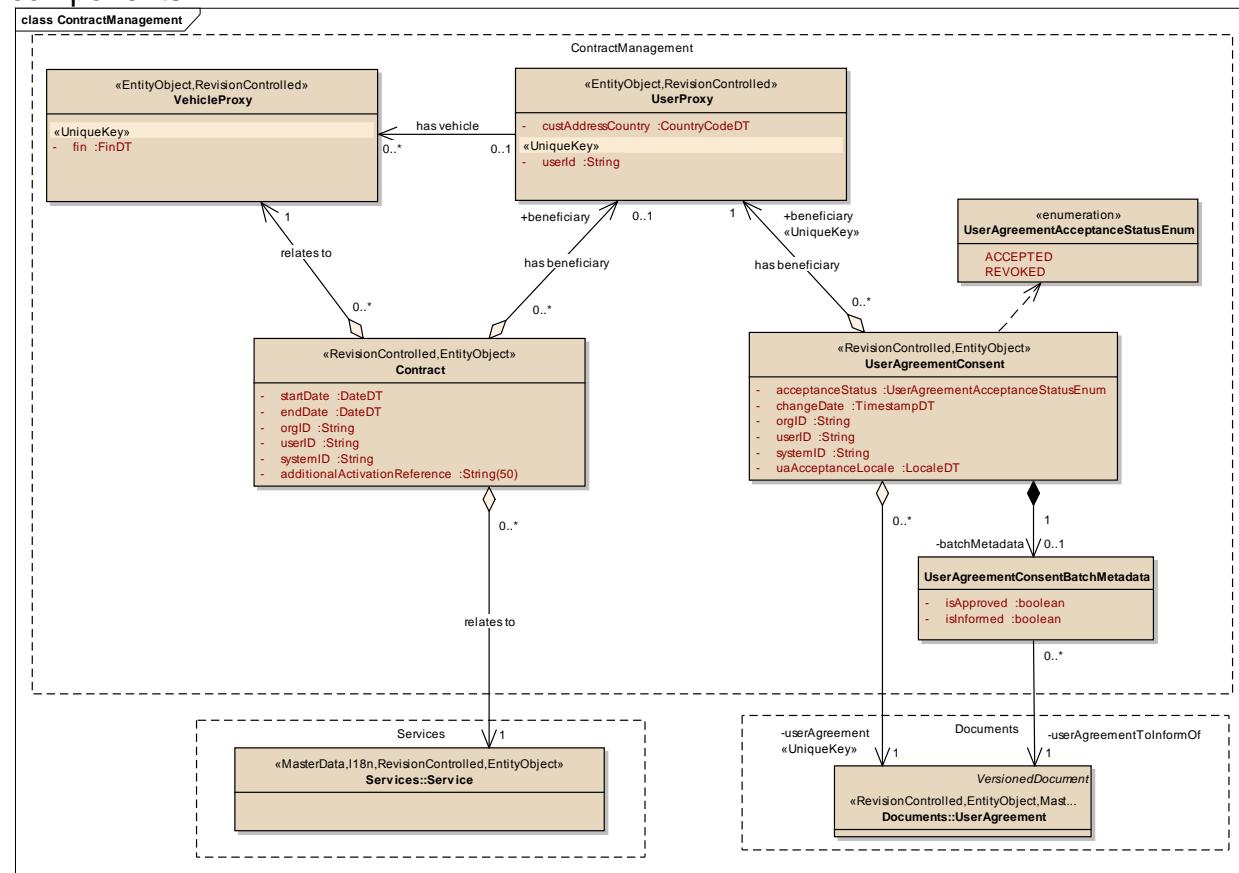
<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>DLG</b>	«enum»	For dialogues.
<b>SERVICE</b>	«enum»	For services.
<b>BATCH</b>	«enum»	For Batches.

## 21.10 ContractManagement

Type: Package  
Package: SOE

## ContractManagement

Overview of the most important SOE entities and their interrelationships among the components.



---

Figure 93: ContractManagement

### 21.10.1 Contract

Type: **Class**

Stereotype: «RevisionControlled,EntityObject»

A contract denotes the relationship between a service and a beneficiary user or a vehicle.

There are services for which the contract belongs to a user and there are services which are bound to a specific vehicle (like "LiveTraffic"). Vehicle bound contract information continues to exist even if a vehicle is not bound to a user any more. (Example "LiveTraffic": Even if a vehicle is sold, the period of time in which the service is free of charge continues to count down. This is independent of a user.)

#### *Connections*

Connector	Source	Target	Description
<b><u>Association</u></b> Source -> Destination	Role: Cardinality: 0..* Type: Contract	Role: Cardinality: 0..1 Type: SalesServicePartner	
<b><u>Association</u></b> Source -> Destination	Role: Cardinality: 0..1 Type: NonTechnicalServiceAvailability	Role: contract Cardinality: 0..1 Type: Contract	
<b><u>Association</u></b> has beneficiary Source -> Destination	Role: Cardinality: 0..* Type: Contract	Role: beneficiary Cardinality: 0..1 Type: UserProxy	This relation associates the user i.e. the beneficiary for which the contract is valid.  Change History: <ul style="list-style-type: none"><li>• changed in [SRS011]: Changed Target from CPD::User to UserProxy</li></ul>
<b><u>Association</u></b> relates to Source -> Destination	Role: Cardinality: 0..* Type: Contract	Role: Cardinality: 1 Type: Service	
<b><u>Association</u></b> relates to Source -> Destination	Role: Cardinality: 0..* Type: Contract	Role: Cardinality: 1 Type: VehicleProxy	For vehicle related contracts this relation associates the vehicle for which the

---

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
on			<p>contract is valid.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• changed in [SRS011]: Changed Target from CPD::Vehicle to VehicleProxy</li> </ul>

#### **Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>startDate</b>	DateDT	Contract's start date. (can be empty)
<b>endDate</b>	DateDT	Contract's end date. (can be empty)
<b>orgID</b>	String	Location where the user agreement was accepted. This information is used for tracing. It's a mere org unit id.
<b>userID</b>	String	The user id of the user that changed the status of a contract. Changes done by the CPD user itself will reflect the same user ID here. For changes that are done by a different user (e.g. a sales person or from the support hotline), this attribute will reflect the person who acts as proxy for the real customer.
<b>systemID</b>	String	System in which the contract has been created.
<b>additionalActivationReference</b>	String(50)	<p>Describes the additional activation reference for the given service.</p> <p>For a service with contract start trigger "initial service activation", the start date for the given service will be set to the given activation date each time a new additional activation reference for a specific user and FIN is given.</p> <p>E.g.: For the service LiveTraffic SD, the additional activation reference will be the ID of the sd card.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS015]</li> </ul>

---

## 21.10.2 UserAgreementAcceptanceStatusEnum

Type: Enumeration

Stereotype: «enumeration»

Enumeration type for the user's consent regarding a user agreement.

### Attributes

Attribute	Type	Notes
ACCEPTED	«enum»	This type indicates that a user has agreed to a user agreement (in a specific version).
REVOKED	«enum»	This type shows that a user has actively revoked a formerly given consent (in a specific version).

## 21.10.3 UserAgreementConsent

Type: Class

Stereotype: «RevisionControlled,EntityObject»

Represents the state of a customer to a user agreement. It also allows to determine which version has been signed by a customer.

### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementConsent	Role: userAgreement Cardinality: 1 Type: UserAgreement	
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: NonTechnicalServiceAvailability	Role: userAgreementConsent Cardinality: 0..1 Type: UserAgreementConsent	
<u>Association</u> Source -> Destination	Role: Cardinality: 1	Role: batchMetadata Cardinality: 0..1	Relation exists whenever a user needs to be in-

---

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
on	Type: UserAgreementConsent	Type: UserAgreementConsent BatchMetadata	formed about legal changes of the user agreement covered by the user agreement consent.  Change history: new in [SRS017]
<u>Association</u> has beneficiary Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementConsent	Role: beneficiary Cardinality: 1 Type: UserProxy	This relation associates the user i.e. the beneficiary who signed an UserAgreementConsent.  Change History: • changed in [SRS011]: Changed Targeted from CPD::User to UserProxy

#### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>acceptanceStatus</b>	UserAgreementAcceptanceStatusEnum	State of the user agreement.
<b>changeDate</b>	TimestampDT	Date and Time information when the state changed of a UserAgreementConsent. This can be both the date of acceptance and the date when the consent was revoked.
<b>orgID</b>	String	Location where the user agreement was accepted. This information is used for tracing. It's a mere org unit id.
<b>userID</b>	String	The user id of the user that changed the status of a user agreement consent. Changes done by the CPD user itself will reflect the same user ID here. For changes that are done by a different user (e.g. a sales person or from the support hotline), this attribute will reflect the person who acts as proxy for the real customer.
<b>systemID</b>	String	System in which the user agreement has been accepted.
<b>uaAcceptanceLocale</b>	LocaleDT	Locale in which the user agreement was accepted. This information is needed for auditing.  Change history:

Attribute	Type	Notes
		<ul style="list-style-type: none"> <li>• new in [SRS001]</li> </ul>

## 21.10.4 UserAgreementConsentBatchMetadata

Type: **Class**

Stereotype:

This entity marks a user agreement consent whose user needs to be informed about an upcoming new user agreement version.

Change history:

new in [SRS017]

### Connections

Connector	Source	Target	Description
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementConsentBatchMetadata	Role: userAgreementToInfor mOf Cardinality: 1 Type: UserAgreement	Points to the version of the user agree- ment which shall be sent to the customer.  Change history: new in [SRS017]
<b>Association</b> Source -> Destination	Role: Cardinality: 1 Type: UserAgreementConsent	Role: batchMetadata Cardinality: 0..1 Type: UserAgreementConsent BatchMetadata	Relation exists whenever a user needs to be in- formed about legal changes of the user agreement covered by the user agree- ment consent.  Change history: new in [SRS017]

### Attributes

Attribute	Type	Notes
<b>isApproved</b>	boolean	Tells that informing the customer about changes of the related user agreement was approved. If set to true, the customer may be informed. If set to false the customer must not be informed.
<b>isInformed</b>	boolean	Tells whether the customer was already informed about the changes of the related user agreement.

---

## 21.10.5 UserProxy

Type: **Class**

Stereotype: «EntityObject, RevisionControlled»

Proxy entity inside SOE for a user in CPD. This entity is needed to be able to add SOE related attributes and relations to a user.

Change History:

- new since [SRS011]

### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 1 Type: UserProxy	Role: Cardinality: 1 Type: User	
<u>Association</u> has beneficiary Source -> Destination	Role: Cardinality: 0..* Type: Contract	Role: beneficiary Cardinality: 0..1 Type: UserProxy	This relation associates the user i.e. the beneficiary for which the contract is valid.  Change History: <ul style="list-style-type: none"><li>• changed in [SRS011]: Changed Target from CPD::User to UserProxy</li></ul>
<u>Association</u> has beneficiary Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementContent	Role: beneficiary Cardinality: 1 Type: UserProxy	This relation associates the user i.e. the beneficiary who signed an UserAgreementContent.  Change History: <ul style="list-style-type: none"><li>• changed in [SRS011]: Changed Target from CPD::User to UserProxy</li></ul>
<u>Association</u> has vehicle Source -> Destination	Role: Cardinality: 0..1 Type: UserProxy	Role: Cardinality: 0..* Type: VehicleProxy	References the vehicles that are registered for a certain user.  Change History: <ul style="list-style-type: none"><li>• new since [SRS011]</li></ul>

### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>userId</b>	String «UniqueKey»	The MBconnect user identifier.
<b>custAddressCountry</b>	CountryCodeDT	Holds the country code of the customer's address in order to reduce CPD calls.  Change History: <ul style="list-style-type: none"><li>• new since [SRS025]</li></ul>

## 21.10.6 VehicleProxy

*Type:* **Class**

*Stereotype:* «EntityObject, RevisionControlled»

Proxy entity inside SOE for a vehicle in CPD. This entity is needed to be able to add SOE related attributes and relations to a Vehicle.

Change History:

- new since [SRS011]

### *Connections*

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<u>Association</u> Source -> Destination	Role: Cardinality: 1 Type: VehicleProxy	Role: Cardinality: 1 Type: Vehicle	
<u>Association</u> has vehicle Source -> Destination	Role: Cardinality: 0..1 Type: UserProxy	Role: Cardinality: 0..* Type: VehicleProxy	References the vehicles that are registered for a certain user.  Change History: <ul style="list-style-type: none"><li>• new since [SRS011]</li></ul>
<u>Association</u> relates to Source -> Destination	Role: Cardinality: 0..* Type: Contract	Role: Cardinality: 1 Type: VehicleProxy	For vehicle related contracts this relation associates the vehicle for which the contract is valid.  Change History: <ul style="list-style-type: none"><li>• changed in [SRS011]: Changed Target from CPD::Vehicle to</li></ul>

Connector	Source	Target	Description
			VehicleProxy

#### Attributes

Attribute	Type	Notes
fin	FinDT «UniqueKey»	The unique FIN of the vehicle.

## 21.11 ServiceManagement

Type: Package  
 Package: SOE

### ServiceManagement

#### Relationships in ServiceManagement

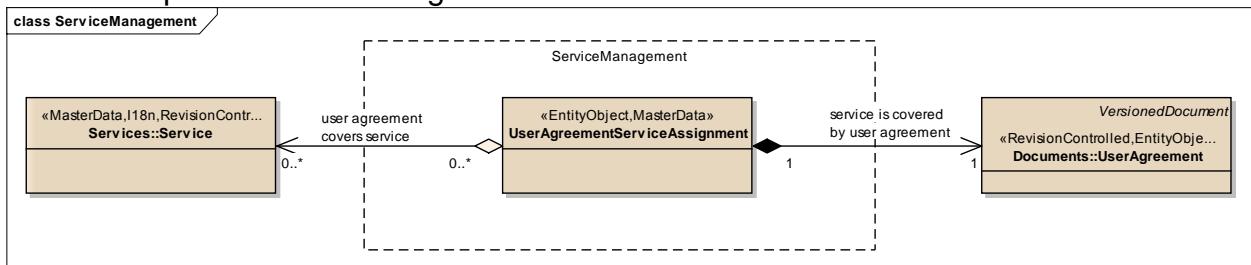


Figure 94: ServiceManagement

### 21.11.1 UserAgreementServiceAssignment

Type: Class

Stereotype: «EntityObject,MasterData»

This entity describes the assignment between a user agreement and a service.

#### Change History:

- new since [SRS008]
- [SRS022] moved from Documents to ServiceManagement

#### Connections

Connector	Source	Target	Description
Association service is covered by user agreement	Role: Cardinality: 1 Type:	Role: Cardinality: 1 Type: UserAgreement	Change History: • new since [SRS008] • [SRS022]:

---

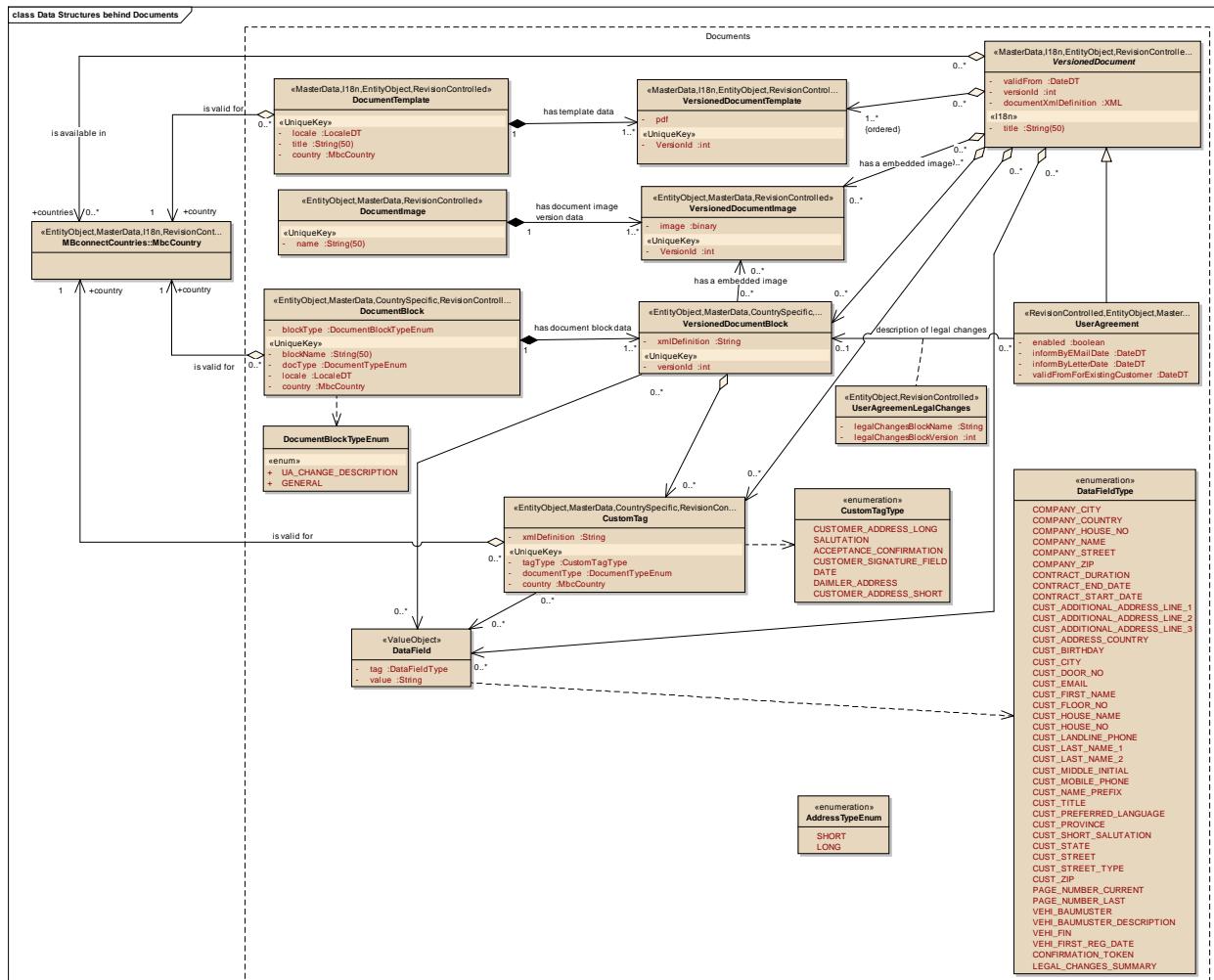
<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
Source -> Destination	UserAgreementService Assignment		changed direction and name
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: UserAgreementService Assignment	
<u>Association</u> user agreement covers service Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementService Assignment	Role: Cardinality: 0..* Type: Service	Change History: • [SRS022]: changed name

## 21.12 Documents

Type: Package  
 Package: SOE

### Data Structures behind Documents

Relationships and data structures of VersionsDocuments.



**Figure 95: Documents**

## Documents

## Relationships for the document handling within SOE.

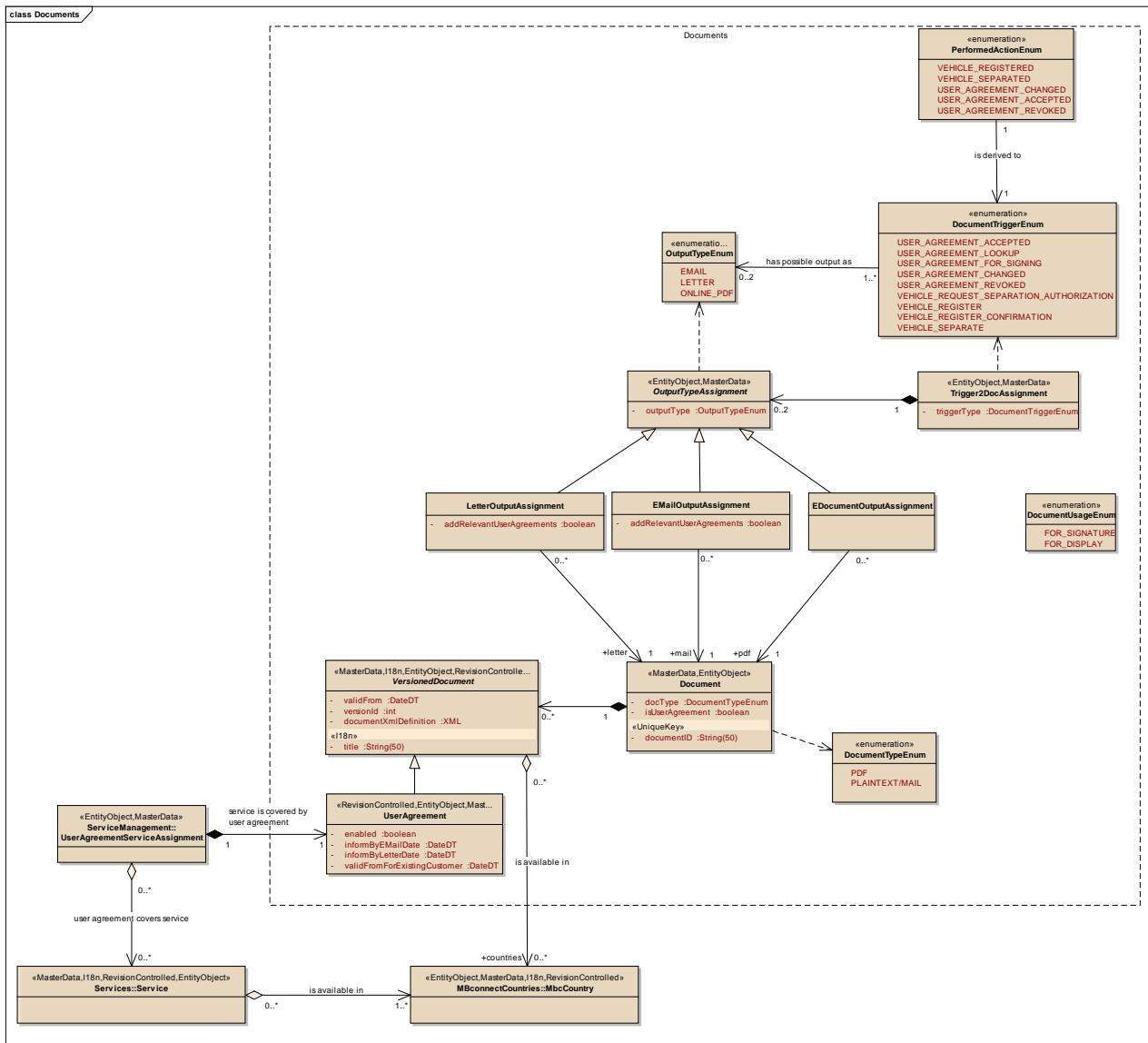


Figure 96: Documents

### 21.12.1 AddressTypeEnum

Type: Enumeration

Stereotype: «enumeration»

Enumeration used to control if the short or long form of an address is needed.

Change History:

- 30.1.2014: renamed from AddressType to AddressTypeEnum to match name convention for enumerations

---

#### **Attributes**

Attribute	Type	Notes
<b>SHORT</b>	«enum»	Represents the short version of an customer address. I.e. all name related <DataFields> are omitted when composing an formatted address.
<b>LONG</b>	«enum»	Represents the long version of a customer address.

#### **21.12.2 CustomTag**

*Type:* **Class**

*Stereotype:* «EntityObject,MasterData,CountrySpecific,RevisionControlled»

Definition of custom tags which can be used in documents. A typical custom tag is the definition of a complete postal address which describes which fields (DataFields) are part of the tag and how these fields are ordered when instantiating a document.

#### **Connections**

Connector	Source	Target	Description
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: CustomTag	Role: Cardinality: 0..* Type: DataField	
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: CustomTag	
<b>Association</b> Source -> Destination	Role: Cardinality: Type: VersionedDocumentBlock	Role: Cardinality: 0..* Type: CustomTag	
<b>Aggregation</b> is valid for Destination -> Source	Role: country Cardinality: 1 Type: MbcCountry	Role: Cardinality: 0..* Type: CustomTag	The country for which a CustomTag is valid.  Change history: • new in [SRS001]
<b>Association</b> takes part in	Role: Cardinality: 0..1	Role: Cardinality: 1	Change History: • new since [SRS005]

---

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
Unspecified	Type: ChangeOperation	Type: CustomTag	

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>tagType</b>	CustomTagType «UniqueKey»	
<b>documentType</b>	DocumentTypeEnum «UniqueKey»	
<b>xmlDefinition</b>	String	
<b>country</b>	MbcCountry «UniqueKey»	

### 21.12.3    **CustomTagType**

*Type:*      **Enumeration**

*Stereotype:*    «enumeration»

Enumeration of all possible custom tags that can be used in documents.

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>CUSTO-MER_ADDRESS_LONG</b>	«enum»	Defines the structure of a customer address. Customer tag type for a whole, formatted customer address. The long version is intended to include not only the rump address (i.e. the street, city, ...) but as well the full name and if necessary the salutation of a person.
<b>SALUTATION</b>	«enum»	Defines the salutation of the customer.
<b>ACCEPTANCE_CONFIRMATION</b>	«enum»	Defines the structure of a confirmation area that contains information about the acceptance of a user agreement by the customer.
<b>CUSTO-MER_SIGNATURE_FIELD</b>	«enum»	Defines the structure of a signature area where the customer signs.
<b>DATE</b>	«enum»	Defines the representation of the date.

Attribute	Type	Notes
DAIMLER_ADDRESS	«enum»	Defines the structure of the Daimler address.
CUSTOMER_ADDRESS_SHORT	«enum»	Defines the structure of a customer address. Customer tag type for a brief formatted customer address. (the short version is intended to be represent a address without name fields, i.e. just the rump of an postal address)

## 21.12.4 DataField

Type: Class

Stereotype: «ValueObject»

A data field consists of tag and value pairs. Data fields are used in document blocks and custom tags.

### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: CustomTag	Role: Cardinality: 0..* Type: DataField	
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: DataField	
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocumentBlock	Role: Cardinality: 0..* Type: DataField	

### Attributes

Attribute	Type	Notes
tag	DataFieldType	Tag which defines the content of a data field.
value	String	A value that is assigned to a data field.

---

## 21.12.5 DataFieldType

Type: Enumeration

Stereotype: «enumeration»

Enumeration of possible data fields that can be used in document blocks.

### Attributes

Attribute	Type	Notes
COMPANY_CITY	«enum»	
COMPANY_COUNTRY	«enum»	
COMPANY_HOUSE_NO	«enum»	
COMPANY_NAME	«enum»	
COMPANY_STREET	«enum»	
COMPANY_ZIP	«enum»	
CONTRACT_DURATION	«enum»	
CONTRACT_END_DATE	«enum»	
CONTRACT_START_DATE	«enum»	
CUST_ADDITIONAL_ADDRESS_LINE_1	«enum»	
CUST_ADDITIONAL_ADDRESS_LINE_2	«enum»	
CUST_ADDITIONAL_ADDRESS_LINE_3	«enum»	
CUST_ADDRESS_COUNTRY	«enum»	
CUST_BIRTHDAY	«enum»	

---

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>CUST_CITY</b>	«enum»	
<b>CUST_DOOR_NO</b>	«enum»	
<b>CUST_EMAIL</b>	«enum»	
<b>CUST_FIRST_NAME</b>	«enum»	
<b>CUST_FLOOR_NO</b>	«enum»	
<b>CUST_HOUSE_NAME</b>	«enum»	
<b>CUST_HOUSE_NO</b>	«enum»	
<b>CUST_LANDLINE_PHONE</b>	«enum»	
<b>CUST_LAST_NAME_1</b>	«enum»	
<b>CUST_LAST_NAME_2</b>	«enum»	
<b>CUST_MIDDLE_INITIAL</b>	«enum»	
<b>CUST_MOBILE_PHONE</b>	«enum»	
<b>CUST_NAME_PREFIX</b>	«enum»	
<b>CUST_TITLE</b>	«enum»	
<b>CUST_PREFERRED_LANGUAGE</b>	«enum»	
<b>CUST_PROVINCE</b>	«enum»	
<b>CUST_SHORT_SALUTATION</b>	«enum»	
<b>CUST_STATE</b>	«enum»	
<b>CUST_STREET</b>	«enum»	
<b>CUST_STREET_TYPE</b>	«enum»	
<b>CUST_ZIP</b>	«enum»	
<b>PA-</b>		

Attribute	Type	Notes
<b>GE_NUMBER_CURRENT</b>	«enum»	
<b>PA-GE_NUMBER_LAST</b>	«enum»	
<b>VEHI_BAUMUSTER</b>	«enum»	
<b>VEHI_BAUMUSTER_DESCRIPTION</b>	«enum»	
<b>VEHI_FIN</b>	«enum»	
<b>VEHI_FIRST_REG_DATE</b>	«enum»	
<b>CONFIRMATION_TOKEN</b>	«enum»	<p>Data field for the customer's configuration token that he has to use to confirm the connection of the vehicle to him in MBconnect.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>new since [SRS002]</li> </ul>
<b>LEGAL_CHANGES_SUMMARY</b>	«enum»	<p>Placeholder for summarizing changes between different versions of legal documents. This data field is intended to be used in cover letters.</p> <p>Change History</p> <ul style="list-style-type: none"> <li>new since [SRS012]</li> </ul>

## 21.12.6 Document

Type: Class

Stereotype: «MasterData,EntityObject»

Static data that define a document. The document definition contains the layout information as well as the static content. In addition to that, it may contain place holders for dynamic content parameters that are used to fill the concrete values.

Change History:

- 
- removed docType from Unique key as it does not match other parts of the sysdocu. implementation never contained docType in the unique key [DW/19.02.2015]

#### Connections

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: EMailOutputAssignment	Role: mail Cardinality: 1 Type: Document	
<b>Association</b> Destination -> Source	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 1 Type: Document	
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: LetterOutputAssignment	Role: letter Cardinality: 1 Type: Document	
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: EDocumentOutputAssignment	Role: pdf Cardinality: 1 Type: Document	
<b>Association</b> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: Document	Change History: • new since [SRS005]

#### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>docType</b>	DocumentTypeEnum	The document type determines the output of a document in the sense of PDF or Plaintext.
<b>documentID</b>	String(50) «UniqueKey»	Name of a document to identify it.
<b>isUserAgreement</b>	boolean	Indicates whether a document represents a user agreement or not. (This is necessary for special functional treatment of user agreements.)

---

## 21.12.7 DocumentBlock

Type: **Class**

Stereotype: «EntityObject,MasterData,CountrySpecific,RevisionControlled»

Definition of a part of a document. A document can be composed of several DocumentBlocks.

### *Connections*

Connector	Source	Target	Description
<u>Aggregation</u> has document block data Destination -> Source	Role: Cardinality: 1..* Type: VersionedDocumentBlock	Role: Cardinality: 1 Type: DocumentBlock	
<u>Aggregation</u> is valid for Destination -> Source	Role: country Cardinality: 1 Type: MbcCountry	Role: Cardinality: 0..* Type: DocumentBlock	The country for which a DocumentBlock is valid.  Change history: <ul style="list-style-type: none"><li>• new in [SRS001]</li></ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: DocumentBlock	Change History: <ul style="list-style-type: none"><li>• new since [SRS005]</li></ul>

### *Attributes*

Attribute	Type	Notes
blockName	String(50) «UniqueKey»	Given name for a document block.
docType	DocumentTypeEnum «UniqueKey»	
locale	LocaleDT «UniqueKey»	
blockType	DocumentBlockTypeEnum	Determines the usage type of a document block.  Change History <ul style="list-style-type: none"><li>• new since [SRS012]</li></ul>
country	MbcCountry	

---

Attribute	Type	Notes
	«UniqueKey»	

## 21.12.8 DocumentImage

Type: Class

Stereotype: «EntityObject,MasterData,RevisionControlled»

Represents an image that can be placed on a document.

Change History:

- new since [SRS018]

### Connections

Connector	Source	Target	Description
<u>Association</u> has document image version data Source -> Destination	Role: Cardinality: 1 Type: DocumentImage	Role: Cardinality: 1..* Type: VersionedDocumentImage	References the individual versions of a document image.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS018]</li> </ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: DocumentImage	Change History: <ul style="list-style-type: none"> <li>• new since [SRS018]</li> </ul>

### Attributes

Attribute	Type	Notes
<b>name</b>	String(50) «UniqueKey»	Maintainable name of a DocumentImage.

## 21.12.9 DocumentTemplate

Type: Class

Stereotype: «MasterData,I18n,EntityObject,RevisionControlled»

The document template represents the "background" of a document. Similar to a background picture, the document's content that is defined by the PDFDocumentDefinition is written on top of the template.

---

This concept allows to have CI conforming backgrounds with static content like brand logo and contact information which should appear on each document.

#### ***Connections***

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<u>Aggregation</u> has template data Destination -> Source	Role: Cardinality: 1..* Type: VersionedDocumentTemplate	Role: Cardinality: 1 Type: DocumentTemplate	
<u>Aggregation</u> is valid for Destination -> Source	Role: country Cardinality: 1 Type: MbcCountry	Role: Cardinality: 0..* Type: DocumentTemplate	The country for which a DocumentTemplate is valid.  Change history: <ul style="list-style-type: none"><li>• new in [SRS001]</li></ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..* Type: ChangeOperation	Role: Cardinality: 1 Type: DocumentTemplate	Change History: <ul style="list-style-type: none"><li>• new since [SRS005]</li></ul>

#### ***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>locale</b>	LocaleDT «UniqueKey»	
<b>title</b>	String(50) «UniqueKey»	Human given name name for a document.
<b>country</b>	MbcCountry «UniqueKey»	The country for which this template is configured.

### **21.12.10 DocumentTriggerEnum**

*Type:* **Enumeration**

*Stereotype:* «enumeration»

Enumeration of the possible trigger events for generation documents.

Change History:

- 30.1.2014: renamed from DocumentTriggerType to DocumentTriggerEnum to match name convention for enumerations

---

### Connections

Connector	Source	Target	Description
<u>Association</u> has possible output as Source -> Destination	Role: Cardinality: 1..* Type: DocumentTriggerEnum	Role: Cardinality: 0..2 Type: OutputTypeEnum	
<u>Association</u> is derived to Source -> Destination	Role: Cardinality: 1 Type: PerformedActionEnum	Role: Cardinality: 1 Type: DocumentTriggerEnum	

### Attributes

Attribute	Type	Notes
<b>USER AGREEMENT ACCEPTED</b>	«enum»	When the user accepts/signs a user agreement.
<b>USER AGREEMENT LOOKUP</b>	«enum»	When the user wants to read a user agreement. (This is necessary for MyMercedes)
<b>USER AGREEMENT FOR SIGNING</b>	«enum»	When a user agreement should be composed and submitted for signing on paper. (Needed for MBC POS)
<b>USER AGREEMENT CHANGED</b>	«enum»	When a user agreement has been changed and released.
<b>USER AGREEMENT REVOKED</b>	«enum»	The user has revoked his acceptance of a user agreement.
<b>VEHICLE REQUEST SEPARATION AUTHORIZATION</b>	«enum»	Whenever a the document is requested to authorize a sales person to separate a vehicle from a customer.
<b>VEHICLE REGISTER</b>	«enum»	When a vehicle is registered in MBconnect for a user.
<b>VEHICLE REGISTER CONFIRMATION</b>	«enum»	When a vehicle is registered in MBconnect for a user in order to trigger printing of the document with the confirmation token for the customer.  Change History: <ul style="list-style-type: none"><li>• new since [SRS002]</li></ul>
<b>VEHICLE SEPARATE</b>		When a vehicle is disconnected i.e.

---

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
	«enum»	separated from a customer.

### 21.12.11 DocumentTypeEnum

*Type:* **Enumeration**

*Stereotype:* «enumeration»

The DocumentType defines which form of output are supported by SOE.

Change History:

- 30.1.2014: renamed from DocumentType to DocumentTypeEnum to match name convention for enumerations

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>PDF</b>	«enum»	for PDF-Documents
<b>PLAINTEXT/MAIL</b>	«enum»	for plain text mails

### 21.12.12 DocumentUsageEnum

*Type:* **Enumeration**

*Stereotype:* «enumeration»

Enumeration used in the interface to indicate the variant of a legal document: With or without a signature.

Change History:

- 30.1.2014: renamed from DocumentUsageType to DocumentUsageEnum to match name convention for enumerations

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>FOR_SIGNATURE</b>	«enum»	When the document is targeted for printing and signing by a customer in the requesting system. (I.e. the

---

Attribute	Type	Notes
		document will show any signature fields)
<b>FOR_DISPLAY</b>	«enum»	When the document is targeted for display only in the requesting system. (I.e. the document will not show any signature fields)

### 21.12.13 EDocumentOutputAssignment

*Type:* Class OutputTypeAssignment

*Stereotype:*

Output assignment for documents that are generated as PDF and returned electronically to the caller. In this case SOE does not send documents to customers directly.

Connections

Connector	Source	Target	Description
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: EDocumentOutputAssignment	Role: Cardinality: Type: OutputTypeAssignment	
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: EDocumentOutputAssignment	Role: pdf Cardinality: 1 Type: Document	

### 21.12.14 EMailOutputAssignment

*Type:* Class OutputTypeAssignment

*Stereotype:*

Output assignment for documents that are sent out via email. In this case SOE sends out documents to customers directly.

Connections

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<b><u>Association</u></b> Source -> Destination	Role: Cardinality: 0..* Type: EMailOutputAssignment	Role: mail Cardinality: 1 Type: Document	
<b><u>Generalization</u></b> Source -> Destination	Role: Cardinality: Type: EMailOutputAssignment	Role: Cardinality: Type: OutputTypeAssignment	

#### ***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>addRelevantUserAgreements</b>	boolean	If true, the relevant user agreements are compiled and are sent as attachment together with the main document.

### **21.12.15 LetterOutputAssignment**

**Type:** Class OutputTypeAssignment

**Stereotype:**

Output assignment for letter communication (whenever SOE needs to trigger printing and sending out letter post). In this case SOE sends out documents to customers directly.

#### ***Connections***

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<b><u>Generalization</u></b> Source -> Destination	Role: Cardinality: Type: LetterOutputAssignment	Role: Cardinality: Type: OutputTypeAssignment	
<b><u>Association</u></b> Source -> Destination	Role: Cardinality: 0..* Type: LetterOutputAssignment	Role: letter Cardinality: 1 Type: Document	

---

#### Attributes

Attribute	Type	Notes
<b>addRelevantUserAgreements</b>	boolean	If true, the relevant user agreements are compiled and are sent as attachment together with the main document.

### 21.12.16 OutputTypeAssignment

Type: Class

Stereotype: «EntityObject,MasterData»

The OutputTypeAssignment is used to relate documents to triggers.  
Semantically there are several types of assignments.

#### Connections

Connector	Source	Target	Description
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: EDocumentOutputAssignment	Role: Cardinality: Type: OutputTypeAssignment	
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: EMailOutputAssignment	Role: Cardinality: Type: OutputTypeAssignment	
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: LetterOutputAssignment	Role: Cardinality: Type: OutputTypeAssignment	
<u>AssociationClass</u> Destination -> Source	Role: Cardinality: 0..2 Type: OutputTypeAssignment	Role: Cardinality: 1 Type: Trigger2DocAssignment	

#### Attributes

Attribute	Type	Notes
<b>outputType</b>	OutputTypeEnum	

---

Attribute	Type	Notes

### 21.12.17 OutputTypeEnum

Type: Enumeration

Stereotype: «enumeration»

Output type of the document: email, letter, online document,...

Change History:

- 30.1.2014: renamed from OutputType to OutputTypeEnum to match name convention for enumerations

#### *Connections*

Connector	Source	Target	Description
<u>Association</u> has possible output as Source -> Destination	Role: Cardinality: 1..* Type: DocumentTriggerEnum	Role: Cardinality: 0..2 Type: OutputTypeEnum	

#### *Attributes*

Attribute	Type	Notes
EMAIL	«enum»	output as email
LETTER	«enum»	Output as printed letter.
ONLINE_PDF	«enum»	Output as electronic document (PDF).

### 21.12.18 PerformedActionEnum

Type: Enumeration

Stereotype: «enumeration»

This enumeration is used in internal interfaces to inform the Documents component which documents have to be generated.

## Change History:

- 30.1.2014: renamed from PerformedActionType to PerformedActionEnum to match name convention for enumerations

### Connections

Connector	Source	Target	Description
<u>Association</u> is derived to Source -> Destination	Role: Cardinality: 1 Type: PerformedActionEnum	Role: Cardinality: 1 Type: DocumentTriggerEnum	

### Attributes

Attribute	Type	Notes
<b>VEHICLE_REGISTERED</b>	«enum»	When the document(s) for vehicle registration are requested.
<b>VEHICLE_SEPARATED</b>	«enum»	When the document(s) for separating a user from a vehicle are requested.
<b>USER AGREEMENT CHANGED</b>	«enum»	Whenever the document(s) are requested that need to be generated upon a change of a user agreement.
<b>USER AGREEMENT ACCEPTED</b>	«enum»	Whenever the document(s) are requested that need to be generated when a user accepted a user agreement.
<b>USER AGREEMENT REVOKED</b>	«enum»	Whenever the document(s) are requested that need to be generated when a user revokes the agreement of a user agreement.

## 21.12.19 Trigger2DocAssignment

Type: Class

Stereotype: «EntityObject,MasterData»

Configuration of the relationship between a DocumentTriggerType and OutputTypeAssignments.

### Connections

Connector	Source	Target	Description
-----------	--------	--------	-------------

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<u><a href="#">AssociationClass</a></u> Destination -> Source	Role: Cardinality: 0..2 Type: OutputTypeAssignment	Role: Cardinality: 1 Type: Trigger2DocAssignment	

***Embedded Elements***

<b>Element</b>	<b>Detail</b>	<b>Notes</b>
Part <anonymous>	Version: 1.0	

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>triggerType</b>	DocumentTriggerEnum	

## 21.12.20 UserAgreementLegalChanges

*Type:* [AssociationClass](#)

*Stereotype:* «EntityObject, RevisionControlled»

Reference to the document blocks that describe the legal changes that belong to a certain UserAgreement. The referenced VersionedDocumentBlocks are described by the combination of legalChangesBlockName and legalChangesBlockVerison

Change History:

- new since [SRS012]

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>legalChangesBlockName</b>	String	The name of the document block where the summary of changes are described that exist in a specific user agreement version. That block is used in the cover letter to summarize the changes for a customer.  Change History: • new since [SRS012]
<b>legalChangesBlockVersion</b>	int	The version id of the document block that is referenced by the at-

Attribute	Type	Notes
		<p>tribute legalChangesBlockName (for more details, see description there).</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS012]</li> </ul>

## 21.12.21 UserAgreement

Type: Class VersionedDocument

Stereotype: «RevisionControlled, EntityObject, MasterData»

The UserAgreement is a special type of a VersionedDocument.

Beispiele:

- Basisdienste
- Live Traffic
- Geofencing
- Remote

### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementConsentBatchMetadata	Role: userAgreementToInfor mOf Cardinality: 1 Type: UserAgreement	Points to the version of the user agree- ment which shall be sent to the customer.  Change history: new in [SRS017]
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementConsent	Role: userAgreement Cardinality: 1 Type: UserAgreement	
<u>Generalization</u> Source -> Destination	Role: Cardinality: Type: UserAgreement	Role: Cardinality: Type: VersionedDocument	
<u>AssociationClass</u> de- scription of legal changes	Role: Cardinality: 0..* Type: UserAgreement	Role: Cardinality: 0..1 Type:	Reference to the document blocks that describe the legal changes that

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
Source -> Destination		VersionedDocumentBlock	<p>belong to a certain UserAgreement. The referenced VersionedDocument Blocks are described by the combination of legalChangesBlockName and legalChangesBlockVersion</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS012]</li> </ul>
<u>Association</u> service is covered by user agreement Source -> Destination	<p>Role: Cardinality: 1 Type: UserAgreementService Assignment</p>	<p>Role: Cardinality: 1 Type: UserAgreement</p>	<p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS008]</li> <li>• [SRS022]: changed direction and name</li> </ul>

#### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>enabled</b>	boolean	Boolean flag that indicates if the user agreement is enabled and thus be available to be processed, e.g. for signing. It also indicates whether the dates for a user agreement can still be changed or not. Once the flag is TRUE, it is not possible to change the dates any more.
<b>informByEMailDate</b>	DateDT	Date when the notifications of user agreement changes are sent out via email (for customers with mail address).
<b>informByLetterDate</b>	DateDT	Date when the notifications of user agreement changes are sent out via letter (for customers without mail).
<b>validFromForExisting Customer</b>	DateDT	Start of validity of a user agreement for existing customer. Note, "validFrom" is used for all new clients. This field here is only used for existing customers instead.

## 21.12.22 VersionedDocument

Type: Class

---

**Stereotype:** «MasterData,I18n,EntityObject,RevisionControlled»

A Document is either a UserAgreement or TermsOfUse Document. Documents can be visualised in several ways. E.g. as PDF-documents or as e-mails.

#### **Connections**

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<b>Association</b> Destination -> Source	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 1 Type: Document	
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 1..* Type: VersionedDocumentTemplate	
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: VersionedDocumentBlock	
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: CustomTag	
<b>Generalization</b> Source -> Destination	Role: Cardinality: Type: UserAgreement	Role: Cardinality: Type: VersionedDocument	
<b>Association</b> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: DataField	
<b>Association</b> has a embedded image Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: VersionedDocumentImage	Reference from a document block to an image that should appear inside the block.  Change History: <ul style="list-style-type: none"><li>• new since [SRS018]</li></ul>
<b>Association</b> is available in Source -> Destination	Role: Cardinality: 0..* Type:	Role: countries Cardinality: 0..* Type: MbcCountry	Denominates the MBconnect countries for which a certain document is valid

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
on	VersionedDocument		for. Change History: <ul style="list-style-type: none"> <li>• new since [SRS022]</li> </ul>

#### **Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>title</b>	String(50) «I18n»	Document name/title.
<b>validFrom</b>	DateDT	Beginn des Gültigkeitszeitraumes eines Dokumentes
<b>versionId</b>	int	Functional Version Id of a document
<b>documentXmlDefinition</b>	XML	Root entity of the document definition. It represents the brace for all contained elements.

### **21.12.23 VersionedDocumentBlock**

*Type:* [Class](#)

*Stereotype:* «EntityObject,MasterData,CountrySpecific,RevisionControlled»

This entity contains the data for a document block. There exists a separate entry for each document block version.

#### **Connections**

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: VersionedDocumentBlock	
<u>Association</u> Source -> Destination	Role: Cardinality: Type: VersionedDocumentBlock	Role: Cardinality: 0..* Type: CustomTag	
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocumentBlock	Role: Cardinality: 0..* Type: DataField	

Connector	Source	Target	Description
	ock		
<u>AssociationClass</u> description of legal changes Source -> Destination	Role: Cardinality: 0..* Type: UserAgreement	Role: Cardinality: 0..1 Type: VersionedDocumentBlock	Reference to the document blocks that describe the legal changes that belong to a certain UserAgreement. The referenced VersionedDocument Blocks are described by the combination of legalChangesBlockName and legalChangesBlockVersion  Change History: <ul style="list-style-type: none"><li>• new since [SRS012]</li></ul>
<u>Association</u> has a embedded image Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocumentBlock	Role: Cardinality: 0..* Type: VersionedDocumentImage	Reference from a document block to an image that should appear inside the block.  Change History: <ul style="list-style-type: none"><li>• new since [SRS018]</li></ul>
<u>Aggregation</u> has document block data Destination -> Source	Role: Cardinality: 1..* Type: VersionedDocumentBlock	Role: Cardinality: 1 Type: DocumentBlock	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: VersionedDocumentBlock	

#### Attributes

Attribute	Type	Notes
<b>versionId</b>	int «UniqueKey»	Version Id of the document block.
<b>xmlDefinition</b>	String	

## 21.12.24 VersionedDocumentImage

Type: Class

---

**Stereotype:** «EntityObject, MasterData, RevisionControlled»

Represents a version of a document image. Each version contains a binary image file.

Change History:

- new since [SRS018]

#### *Connections*

Connector	Source	Target	Description
<u>Association</u> has a embedded image Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 0..* Type: VersionedDocumentImage	Reference from a document block to an image that should appear inside the block.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS018]</li> </ul>
<u>Association</u> has a embedded image Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocumentBlock	Role: Cardinality: 0..* Type: VersionedDocumentImage	Reference from a document block to an image that should appear inside the block.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS018]</li> </ul>
<u>Association</u> has document image version data Source -> Destination	Role: Cardinality: 1 Type: DocumentImage	Role: Cardinality: 1..* Type: VersionedDocumentImage	References the individual versions of a document image.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS018]</li> </ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: VersionedDocumentImage	Change History: <ul style="list-style-type: none"> <li>• new since [SRS018]</li> </ul>

#### *Attributes*

Attribute	Type	Notes
<b>image</b>	binary	The binary representation of the image that is represented by the VersionedDocumentImage.
<b>VersionId</b>	int «UniqueKey»	Version Id of the image. This version is independent of the VersionedDocument's version id.

---

## 21.12.25 VersionedDocumentTemplate

Type: Class

Stereotype: «MasterData,I18n,EntityObject,RevisionControlled»

This entity contains the data for a document template. There exists a separate entry for each template version.

### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: VersionedDocument	Role: Cardinality: 1..* Type: VersionedDocumentTemplate	
<u>Aggregation</u> has template data Destination -> Source	Role: Cardinality: 1..* Type: VersionedDocumentTemplate	Role: Cardinality: 1 Type: DocumentTemplate	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: VersionedDocumentTemplate	

### Attributes

Attribute	Type	Notes
VersionId	int «UniqueKey»	Version Id of the document template. This version is independent of the VersionedDocument's version id.
pdf		Binary attribute storing the PDF document that is used as template for the document. This is the actual template data.

## 21.13 Services

Type: Package

Package: SOE

## Services

Overview of the entities that belong to Services.

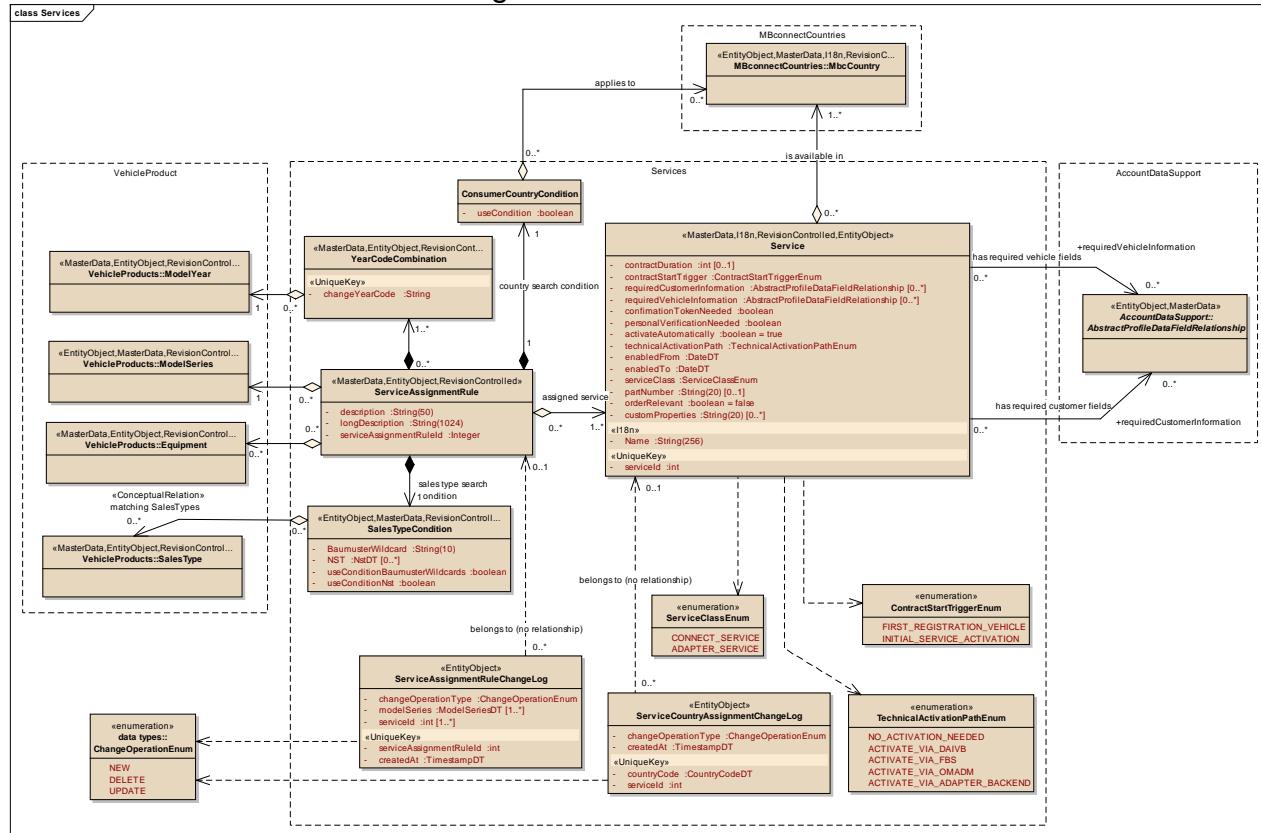


Figure 97: Services

### 21.13.1 ConsumerCountryCondition Class

Type: Class

Stereotype:

This entity stores the search condition for a ServiceAssignmentRule regarding the consumer country. If this condition is fulfilled, the rule services assigned to the rule are possible for a vehicle from the sales perspective.

Change History:

- new since [SRS032]

### Connections

Connector	Source	Target	Description
<u>Association</u> applies to Source -> Destination	Role: Cardinality: 0..* Type:	Role: Cardinality: 0..* Type: MbcCountry	Defines which countries the rule applies to.  Change History:

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
on	ConsumerCountryCondition		<ul style="list-style-type: none"> <li>• new since [SRS032]</li> </ul>
<u>Association</u> country search condition Source -> Destination	Role: Cardinality: 1 Type: ServiceAssignmentRule	Role: Cardinality: 1 Type: ConsumerCountryCondition	<p>Links the rule and the country condition.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS032]</li> </ul>

#### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>useCondition</b>	boolean	Tells whether the country condition shall be taken in account (true) or not (false) when the rule is evaluated.

### **21.13.2 ContractStartTriggerEnum**

*Type:* Enumeration

*Stereotype:* «enumeration»

Indicates what event triggers the activation of a contract.

#### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>FIRST_REGISTRATION_VEHICLE</b>	«enum»	Indicates that a contract duration starts from the first vehicle registration in SOE. The contract will be activated automatically upon the first vehicle registration.
<b>INITIAL_SERVICE_ACTIVATION</b>	«enum»	<p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS015]</li> </ul>

### **21.13.3 SalesTypeCondition**

*Type:* Class

*Stereotype:* «EntityObject,MasterData,RevisionControlled»

---

This entity stores the search condition for a ServiceAssignmentRule. If this condition is fulfilled, the rule services assigned to the rule are possible for a vehicle from the sales perspective.

#### Change History:

- new since [SRS030]

#### Connections

Connector	Source	Target	Description
<u>Association</u> matching SalesTypes Source -> Destination	Role: Cardinality: 0..* Type: SalesTypeCondition	Role: Cardinality: 0..* Type: SalesType	This relation is transient (not persistent) and emerges from applying the search condition in the vehicle master data by dynamically interpreting the SalesTypeCondition.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS030]</li> </ul>
<u>Association</u> sales type search condition Source -> Destination	Role: Cardinality: 1 Type: ServiceAssignmentRule	Role: Cardinality: 1 Type: SalesTypeCondition	Links the rule and the sales type condition.  Change History: <ul style="list-style-type: none"> <li>• new since [SRS030]</li> <li>• changed name to avoid misunderstandings [SRS032]</li> </ul>

#### Attributes

Attribute	Type	Notes
<b>BaumusterWildcard</b>	String(10)	Search condition for a matching Baumuster. This pattern may contain wildcards.
<b>NST [0..*]</b>	NstDT	Search condition for a National Sales Type.
<b>useConditionBaumusterWildcards</b>	boolean	Boolean flag that indicates if the search condition under BaumusterWildcard shall be used during a search (TRUE) or ignored (FALSE).
<b>useConditionNst</b>	boolean	Boolean flag that indicates if the search condition under Nst shall be used during a search (TRUE) or ignored (FALSE).

---

## 21.13.4 Service

Type: Class

Stereotype: «MasterData,I18n,RevisionControlled,EntityObject»

Dienst

Beispiele:

- Accident Management
- Breakdown Management
- Maintenance & Prädiktive Diagnose
- Live Traffic
- Vehicle Finder
- Vehicle Tracker / Geofencing
- Remote Lock/Unlock
- Remote Status
- Remote Standheizung

User BOM: *Dienstkomponente*

Change History:

- removed attribute "serviceOwnerReference" [SRS038]

### Connections

Connector	Source	Target	Description
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: NonTechnicalServiceAvailability	Role: service Cardinality: 1 Type: Service	Entweder implizit durch Zustimmung zur Nutzungsvereinbarung verfügbarer Dienst oder explizit durch einen Vertragsabschluss gebuchter Dienst.
<u>Association</u> Source -> Destination	Role: Cardinality: 0..* Type: Service	Role: marketingService Cardinality: 1 Type: Service	
<u>Association</u> assigned service Source -> Destination	Role: Cardinality: 0..* Type: ServiceAssignmentRule	Role: Cardinality: 1..* Type: Service	Defines which services are available in case the rule applies.
<u>Association</u> has required customer fields Source -> Destination	Role: Cardinality: 0..* Type: Service	Role: requiredCustomerInformation Cardinality: 0..* Type: AbstractProfileDataFiel	Mandatory fields for a service.

Connector	Source	Target	Description
		dRelationship	
<u>Association</u> has required vehicle fields Source -> Destination	Role: Cardinality: 0..* Type: Service	Role: requiredVehicleInformation Cardinality: 0..* Type: AbstractProfileDataFieldRelationship	<b>Mandatory fields for a service.</b>
<u>Association</u> is available in Source -> Destination	Role: Cardinality: 0..* Type: Service	Role: Cardinality: 1..* Type: MbcCountry	References the country in which a service is available.  Change History: <ul style="list-style-type: none"><li>new since [SRS011]</li></ul>
<u>Association</u> relates to Source -> Destination	Role: Cardinality: 0..* Type: Contract	Role: Cardinality: 1 Type: Service	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: Service	Change History: <ul style="list-style-type: none"><li>new since [SRS005]</li></ul>
<u>Association</u> user agreement covers service Source -> Destination	Role: Cardinality: 0..* Type: UserAgreementServiceAssignment	Role: Cardinality: 0..* Type: Service	Change History: <ul style="list-style-type: none"><li>[SRS022]: changed name</li></ul>

#### Attributes

Attribute	Type	Notes
<b>Name</b>	String(256) «I18n»	
<b>serviceId</b>	int «UniqueKey»	Functional Id of a service.
<b>contractDuration</b> [0..1]	int	Describes the contract duration in months, in case a contract for this service is established. This attribute is optional.
<b>contractStartTrigger</b>	ContractStartTriggerEnum	Indicates what event triggers the begin of a contract related to this service.
<b>requiredCustomerInfo</b> rmation [0..*]	AbstractProfileDataFieldRelationship	List with the mandatory fields for a service that relate to attributes of the customer profile.

Attribute	Type	Notes
		<p>Change History:</p> <ul style="list-style-type: none"> <li>• [SRS003] Changed Type from ProfileDataFieldIdEnum to AbstractProfileDataField</li> </ul>
<b>requiredVehicleInformation</b> [0..*]	AbstractProfileDataFieldRelationship	<p>List with the mandatory fields for a service that relate to attributes of a vehicle.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS003]</li> </ul>
<b>confirmationTokenNeeded</b>	boolean	<p>If "TRUE" a service needs a confirmed connection between the user and a vehicle through a verification token in. Such a service must be activated only after a successful verification.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS002]</li> </ul>
<b>personalVerificationNeeded</b>	boolean	<p>If "TRUE" a retailer has to verify the identity of the customer in order to use this service.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS002]</li> </ul>
<b>activateAutomatically</b>	boolean	<p>States if the service is activated automatically when the vehicle is registered to a customer.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS010]</li> </ul>
<b>technicalActivationPath</b>	TechnicalActivationPathEnum	<p>Determines the technical activation path, i.e. the system, that has to be contacted in order to activate or deactivate the service in the vehicle.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS023]</li> </ul>
<b>enabledFrom</b>	DateDT	The start date of a service (including this day). From this date the

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
		<p>service is available of the sales' point of view.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS025]</li> </ul>
<b>enabledTo</b>	DateDT	<p>The last date of a service (including this day). From this date the service is not available of the sales' point of view.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS025]</li> </ul>
<b>serviceClass</b>	ServiceClassEnum	<p>Describes the class of a service. E.g. ConnectMe-Service or Adapter-Service.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS039]</li> </ul>
<b>partNumber</b> [0..1]	String(20)	<p>The part number of the service.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS038]</li> </ul>
<b>orderRelevant</b>	boolean	<p>Tells whether the service must be paid/purchased before activation.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS038]</li> </ul>
<b>customProperties</b> [0..*]	String(20)	<p>Contains keys for service specific properties which are required to activate the service. These keys are checked against the dynamic customer fields in the CPD by the CPD.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS038]</li> </ul>

## 21.13.5 ServiceAssignmentRule

*Type:* Class

**Stereotype:** «MasterData, EntityObject, RevisionControlled»

A sales bundle rule combines a possible vehicle configuration (model series and optional codes, etc.) to a Service.

#### *Connections*

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<b>Association</b> Destination -> Source	Role: Cardinality: 1 Type: ModelSeries	Role: Cardinality: 0..* Type: ServiceAssignmentRule	Links the assigned model series. The rule applies if one of the model series is matched.  Change History: <ul style="list-style-type: none"><li>• [SRS004]: Moved relationship from SalesTypes to ModelSeries</li><li>• [SRS030]: Changed cardinality towards ModelSeries from 1..* to 1</li></ul>
<b>Aggregation</b> Destination -> Source	Role: Cardinality: 1..* Type: YearCodeCombination	Role: Cardinality: 0..* Type: ServiceAssignmentRule	Links the assigned year code combination. The rule applies if one of the model years is matched.  • [SRS040] : Source of Link is YearCodeCombination instead of Model Year to support ChangeYears
<b>Association</b> Destination -> Source	Role: Cardinality: 0..* Type: Equipment	Role: Cardinality: 0..* Type: ServiceAssignmentRule	Links the assigned equipment codes. The rule applies if all equipment codes are matched.
<b>Association</b> assigned service Source -> Destination	Role: Cardinality: 0..* Type: ServiceAssignmentRule	Role: Cardinality: 1..* Type: Service	Defines which services are available in case the rule applies.
<b>Association</b> country search condition Source -> Destination	Role: Cardinality: 1 Type: ServiceAssignmentRule	Role: Cardinality: 1 Type: ConsumerCountryCondition	Links the rule and the country condition.  Change History: <ul style="list-style-type: none"><li>• new since [SRS032]</li></ul>
<b>Association</b> sales	Role:	Role:	Links the rule and the sales type condi-

---

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
type search condition Source -> Destination	Cardinality: 1 Type: ServiceAssignmentRule	Cardinality: 1 Type: SalesTypeCondition	tion.  Change History: <ul style="list-style-type: none"><li>• new since [SRS030]</li><li>• changed name to avoid misunderstandings [SRS032]</li></ul>
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ServiceAssignmentRule	Change History: <ul style="list-style-type: none"><li>• new since [SRS005]</li></ul>

**Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>description</b>	String(50)	A name for a rule. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
<b>longDescription</b>	String(1024)	A comment on the rule that explains or summarizes the rule in greater detail. The name is entered by a user and only used internally in SOE to enhance readability. It is neither internationalized nor used for "marketing" purposes.
<b>serviceAssignmentRuleId</b>	Integer	The service assignment rule ID is a functional ID that helps identify the service assignment and is therefore also displayed in the dialog.  Change history: changed datatype from int to string in [SRS035]

## 21.13.6 ServiceAssignmentRuleChangeLog

*Type:* Class

*Stereotype:* «EntityObject»

Logs changes of ServiceAssignmentRules.

Change History:

- 
- new since [SRS025]

**Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>changeOperationType</b>	ChangeOperationEnum	The change operation that belongs to the change log entry.
<b>serviceAssignmentRuleId</b>	int «UniqueKey»	The id of the modified service assignment rule. This id is essential if a ServiceAssignmentRule has been deleted.
<b>modelSeries [1..*]</b>	ModelSeriesDT	Model series that is affected by the ServiceAssignmentRule change.
<b>serviceId [1..*]</b>	int	Service ids that are affected by the ServiceAssignmentRule change.
<b>createdAt</b>	TimestampDT «UniqueKey»	Timestamp when a change log entry was created.

### 21.13.7 ServiceClassEnum

*Type:* **Enumeration**

*Stereotype:* «enumeration»

Describes the class of a service. E.g. ConnectMe-Service or Adapter-Service.

Change History:

- new since [SRS039]

**Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>CONNECT_SERVICE</b>	«enum»	"REAL" connect me service.
<b>ADAPTER_SERVICE</b>	«enum»	Service provided via OBD2-Adatper.

### 21.13.8 ServiceCountryAssignmentChangeLog

*Type:* **Class**

---

*Stereotype:* «EntityObject»

Logs changes of Service.

Change History:

- new since [SRS025]

**Attributes**

Attribute	Type	Notes
<b>changeOperationType</b>	ChangeOperationEnum	The change operation that belongs to the change log entry. Only "NEW" and "DELETED" are possible here.
<b>createdAt</b>	TimestampDT	Timestamp when a change log entry was created.
<b>countryCode</b>	CountryCodeDT «UniqueKey»	country that is affected by the Service change.
<b>serviceId</b>	int «UniqueKey»	Functional Id of a service.

### **21.13.9 TechnicalActivationPathEnum**

*Type:* **Enumeration**

*Stereotype:* «enumeration»

Enumeration of the allowed values for the activation path of a service.

Change History:

- new since [SRS023]

**Attributes**

Attribute	Type	Notes
<b>NO_ACTIVATION_NE EDED</b>	«enum»	When a service does not need to be activated through any system, i.e. it is already activated by other means outside MBconnect.
<b>ACTIVA TE_VIA_DAIVB</b>	«enum»	When a service must be activated through DaiVB.
<b>ACTIVATE_VIA_FBS</b>	«enum»	Use FBS backend system to activate this service.

Attribute	Type	Notes
		Change History: • new since [SRS038]
<b>ACTIVA-TE_VIA_OMADM</b>	«enum»	Use OMADM backend system to activate this service.  Change History: • new since [SRS038]
<b>ACTIVA-TE_VIA_ADAPTER_B ACKEND</b>	«enum»	Use adapter backend system to activate this service.  Change History: • new since [SRS039]

### 21.13.10 YearCodeCombination

Type: Class

Stereotype: «MasterData, EntityObject, RevisionControlled»

Constitutes ModelYear and ChangeYear combinations

Change history:

- New since [SRS040]

#### Connections

Connector	Source	Target	Description
<u>Aggregation</u> Destination -> Source	Role: Cardinality: 1 Type: ModelYear	Role: Cardinality: 0..* Type: YearCodeCombination	Links a Model Year.  • new since [SRS040]
<u>Aggregation</u> Destination -> Source	Role: Cardinality: 1..* Type: YearCodeCombination	Role: Cardinality: 0..* Type: ServiceAssignmentRule	Links the assigned year code combination. The rule applies if one of the model years is matched.  • [SRS040] : Source of Link is YearCodeCombination instead of Model Year to support ChangeYears

---

Connector	Source	Target	Description

**Attributes**

Attribute	Type	Notes
changeYearCode	String «UniqueKey»	

## 21.14      VehicleProducts

*Type:*

**Package**

*Package:*

SOE

### **VehicleProducts**

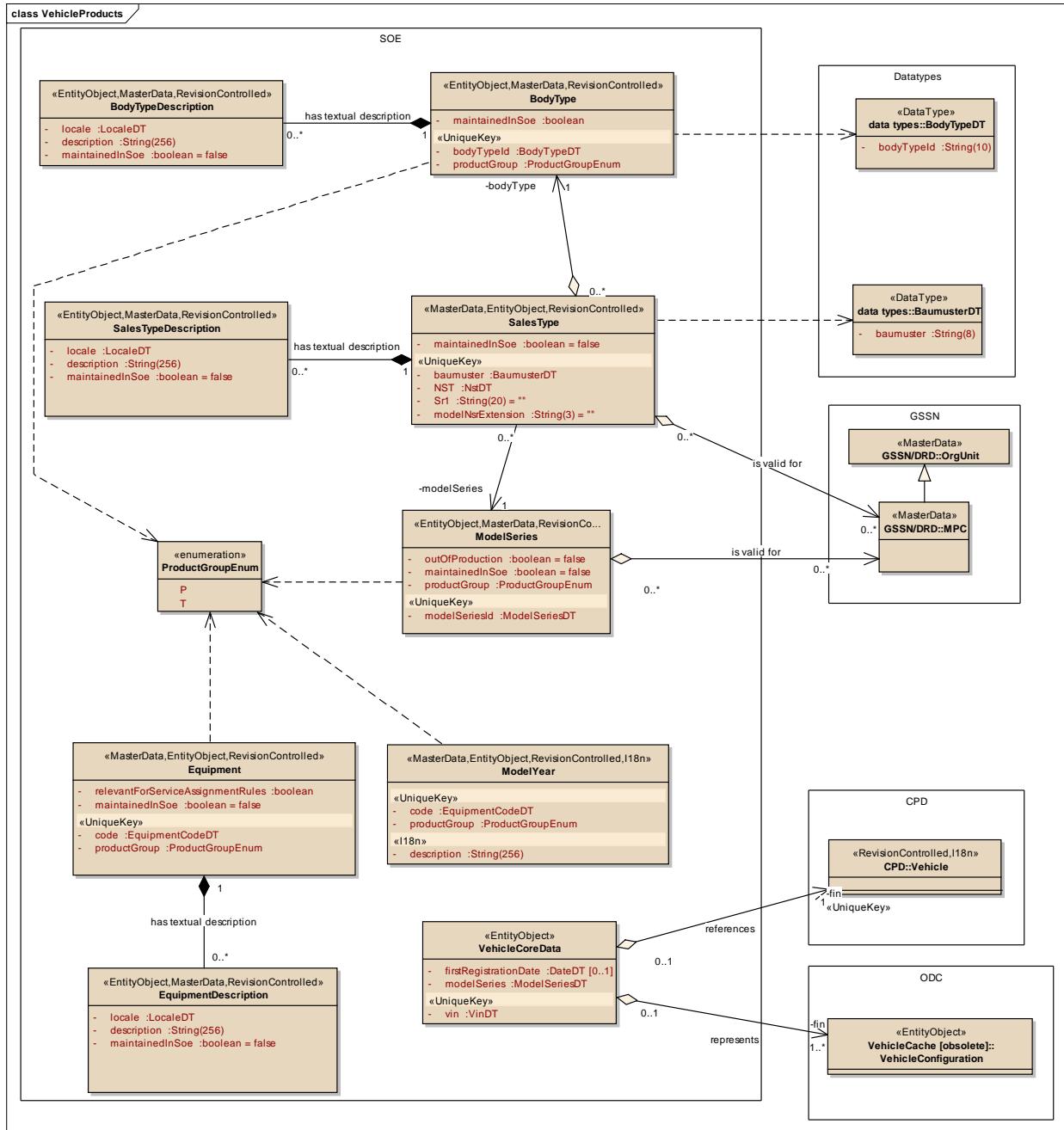


Figure 98: VehicleProducts of the entities that belong to Vehicle Products.

## 21.14.1 BodyType

Type: **Class**

Stereotype: «EntityObject,MasterData,RevisionControlled»

Encapsulates the base identification for a body type (German: "Aufbauart").

Change History:

- new since [SRS030]

- 
- [SRS045] modified business key from BodyTypeld to BodyTypeld+ProductGroup

#### *Connections*

Connector	Source	Target	Description
<u>Aggregation</u> Destination -> Source	Role: bodyType Cardinality: 1 Type: BodyType	Role: Cardinality: 0..* Type: SalesType	Change History: • new since [SRS030] • fixed direction, name and cardinality [26.02.2015 DW]
<u>Aggregation</u> has textual description Source -> Destination	Role: Cardinality: 1 Type: BodyType	Role: Cardinality: 0..* Type: BodyTypeDescription	Change History: • new since [SRS030]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: BodyType	Change History: • new since [SRS030]

#### *Attributes*

Attribute	Type	Notes
<b>bodyTypeld</b>	BodyTypeDT «UniqueKey»	The id of the body type.
<b>maintainedInSoe</b>	boolean	This flag is set to true if the body type is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.  Change History: - added as a fix of the BOM [26.02.2015 DW]
<b>productGroup</b>	ProductGroupEnum «UniqueKey»	The related product group.  Change History: • new since [SRS045]

## 21.14.2 BodyTypeDescription

*Type:* Class

*Stereotype:* «EntityObject,MasterData,RevisionControlled»

---

This entity stores a localized description text for a body type.

Change History:

- new since [SRS030]

#### Connections

Connector	Source	Target	Description
<u>Aggregation</u> has textual description Source -> Destination	Role: Cardinality: 1 Type: BodyType	Role: Cardinality: 0..* Type: BodyTypeDescription	Change History: • new since [SRS030]

#### Attributes

Attribute	Type	Notes
locale	LocaleDT	The locale for which a description is valid.
description	String(256)	The (localized) description of a sales type.
maintainedInSoe	boolean	This flag is set to true if the sales type description is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.  Change History • new since [SRS004]

### 21.14.3 Equipment

Type: Class

Stereotype: «MasterData, EntityObject, RevisionControlled»

Code for equipments of a vehicle.

Change History:

- [SRS004]: moved description to separate entity
- [SRS045]: modified business key from CODE to CODE+ProductGroup

#### Connections

Connector	Source	Target	Description
<u>Association</u>	Role:	Role:	Links the assigned

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
Destination -> Source	Cardinality: 0..* Type: Equipment	Cardinality: 0..* Type: ServiceAssignmentRule	equipment codes. The rule applies if all equipment codes are matched.  Change History: changed
<u>Aggregation</u> has textual description Source -> Destination	Role: Cardinality: 1 Type: Equipment	Role: Cardinality: 0..* Type: EquipmentDescription	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: Equipment	Change History: • new since [SRS005]

#### **Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>code</b>	EquipmentCodeDT «UniqueKey»	Special equipment code.
<b>relevantForServiceAssignmentRules</b>	boolean	The flag is set to true if the equipment code is relevant for maintaining the service assignment rules. Only equipment codes with this flag set to true are available on DLG_ServiceAssignmentRuleDetails. The flag is false by default.  Change History: • new since [SRS004]
<b>maintainedInSoe</b>	boolean	This flag is set to true if the equipment description is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.  Change History: • new since [SRS004]
<b>productGroup</b>	ProductGroupEnum «UniqueKey»	The related product group.  Change History: • new since [SRS045]

#### **21.14.4 EquipmentDescription**

**Type:** Class

---

*Stereotype:* «EntityObject,MasterData,RevisionControlled»

This entity stores a localized equipment description.

Change History:

- new since [SRS004]

#### Connections

Connector	Source	Target	Description
<u>Aggregation</u> has textual description Source -> Destination	Role: Cardinality: 1 Type: Equipment	Role: Cardinality: 0..* Type: EquipmentDescription	
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: EquipmentDescription	

#### Attributes

Attribute	Type	Notes
<b>locale</b>	LocaleDT	The locale for which a description is valid.
<b>description</b>	String(256)	The (localized) description of a equipment code.
<b>maintainedInSoe</b>	boolean	This flag is set to true if the equipment description is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.  Change History • new since [SRS004]

## 21.14.5 ModelSeries

*Type:* Class

*Stereotype:* «EntityObject,MasterData,RevisionControlled»

The entity ModelSeries is used to store attributes of a model series.

## Change History

- new since [SRS004]

### *Connections*

Connector	Source	Target	Description
<b>Association</b> Destination -> Source	Role: Cardinality: 1 Type: ModelSeries	Role: Cardinality: 0..* Type: ServiceAssignmentRule	Links the assigned model series. The rule applies if one of the model series is matched.  Change History: <ul style="list-style-type: none"> <li>• [SRS004]: Moved relationship from SalesTypes to ModelSeries</li> <li>• [SRS030]: Changed cardinality towards ModelSeries from 1..* to 1</li> </ul>
<b>Association</b> Destination -> Source	Role: modelSeries Cardinality: 1 Type: ModelSeries	Role: Cardinality: 0..* Type: SalesType	Change History: <ul style="list-style-type: none"> <li>• new since [SRS004]</li> <li>• [SRS030]: changed cardinality on the side of ModelSeries from 1 to 0..1</li> <li>• fixed direction, name and cardinality [26.02.2015 DW]</li> </ul>
<b>Association</b> is valid for Source -> Destination	Role: Cardinality: 0..* Type: ModelSeries	Role: Cardinality: 0..* Type: MPC	Change History: <ul style="list-style-type: none"> <li>• new since [SRS004]</li> </ul>
<b>Association</b> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: ModelSeries	Change History: <ul style="list-style-type: none"> <li>• new since [SRS005]</li> </ul>

### *Attributes*

Attribute	Type	Notes
<b>modelSeriesId</b>	ModelSeriesDT «UniqueKey»	The id of the model series e.g. "203"
<b>outOfProduction</b>	boolean	This flag is set to true if the sales type is out of production. The flag is false by default.
<b>maintainedInSoe</b>	boolean	This flag is set to true if the equip-

Attribute	Type	Notes
		ment description is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.
<b>productGroup</b>	ProductGroupEnum	The related product group.  Change History: <ul style="list-style-type: none"><li>• new since [SRS045]</li></ul>

## 21.14.6 ModelYear

Type: Class

Stereotype: «MasterData,EntityObject,RevisionControlled,I18n»

Model Year code for vehicles.

Through the model year code it is possible to restrict sales bundles to certain model years.

Change history:

- Removed I18n [SRS040]
- [SRS045]: modified business key from CODE to CODE+ProductGroup

### Connections

Connector	Source	Target	Description
<u>Aggregation</u> Destination -> Source	Role: Cardinality: 1 Type: ModelYear	Role: Cardinality: 0..* Type: YearCodeCombination	Links a Model Year. <ul style="list-style-type: none"><li>• new since [SRS040]</li></ul>

### Attributes

Attribute	Type	Notes
<b>code</b>	EquipmentCodeDT «UniqueKey»	Model year code with max. 5 digits.
<b>description</b>	String(256) «I18n»	
<b>productGroup</b>	ProductGroupEnum «UniqueKey»	The related product group.  Change History: <ul style="list-style-type: none"><li>• new since [SRS045]</li></ul>

---

## 21.14.7 ProductGroupEnum [new]

Type: Enumeration

Stereotype: «enumeration»

Contains the allowed product groups.

Change History:

- new since [SRS045]

### Attributes

Attribute	Type	Notes
P	«enum»	PKW
T	«enum»	transporter

## 21.14.8 SalesType

Type: Class

Stereotype: «MasterData,EntityObject,RevisionControlled»

Encapsulates the base identifier for a sales type. A sales type consists of a Baumuster together with an optional NST.

Change History:

- [SRS004]: moved description to separate entity
- [SRS045]: modified business key from BM+NST to BM+NST+SR1+ModelNsRExtension

### Connections

Connector	Source	Target	Description
<u>Association</u> Destination -> Source	Role: modelSeries Cardinality: 1 Type: ModelSeries	Role: Cardinality: 0..* Type: SalesType	Change History: <ul style="list-style-type: none"><li>• new since [SRS004]</li><li>• [SRS030]: changed cardinality on the side of ModelSeries from 1 to 0..1</li><li>• fixed direction, name and cardinality [26.02.2015]</li></ul>

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
			DW]
<u>Aggregation</u> Destination -> Source	Role: bodyType Cardinality: 1 Type: BodyType	Role: Cardinality: 0..* Type: SalesType	Change History: • new since [SRS030] • fixed direction, name and cardinality [26.02.2015 DW]
<u>Aggregation</u> has textual description Source -> Destination	Role: Cardinality: 1 Type: SalesType	Role: Cardinality: 0..* Type: SalesTypeDescription	Change History: • new since [SRS030]
<u>Association</u> is valid for Source -> Destination	Role: Cardinality: 0..* Type: SalesType	Role: Cardinality: 0..* Type: MPC	Change History: • new since [SRS030]
<u>Association</u> matching SalesTypes Source -> Destination	Role: Cardinality: 0..* Type: SalesTypeCondition	Role: Cardinality: 0..* Type: SalesType	This relation is transient (not persistent) and emerges from applying the search condition in the vehicle master data by dynamically interpreting the SalesTypeCondition.  Change History: • new since [SRS030]
<u>Association</u> takes part in Unspecified	Role: Cardinality: 0..1 Type: ChangeOperation	Role: Cardinality: 1 Type: SalesType	Change History: • new since [SRS030]

#### Attributes

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>baumuster</b>	BaumusterDT «UniqueKey»	Baumuster 7.
<b>NST</b>	NstDT «UniqueKey»	National Sales Type. Can be empty.  Der NST erweitert den Baumusterbegriff um eine marktabhängige Fahrzeugverkaufsbezeichnung. Die Kombination von BM+NST ermöglicht primär 2 Aspekte: 1. Zusätzlich zum zentralen Serienumfang können marktspezifische Zusätze (je nach

Attribute	Type	Notes
		<p>Markt+NST) als Serie dargestellt werden.</p> <p>2. Die NSTs ermöglichen eine spezifischere Baumusterbeschreibung (wie z.B. Blue Efficiency) zur Steigerung von Transparenz und Attraktivität im Vertriebsprozess.</p>
<b>maintainedInSoe</b>	boolean	<p>This flag is set to true if the sales type is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS030]</li> </ul>
<b>Sr1</b>	String(20) «UniqueKey»	<p>sr1 for Vans.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS045]</li> </ul>
<b>modelNsrExtension</b>	String(3) «UniqueKey»	<p>The nsr extension for Vans.</p> <p>Change History:</p> <ul style="list-style-type: none"> <li>• new since [SRS045]</li> </ul>

## 21.14.9 SalesTypeDescription

Type: **Class**

Stereotype: «EntityObject,MasterData,RevisionControlled»

This entity stores a localized description text for a sales type.

Change History:

- new since [SRS030]

### Connections

Connector	Source	Target	Description
<u>Aggregation</u> has textual description Source -> Destinati-	Role: Cardinality: 1 Type: SalesType	Role: Cardinality: 0..* Type:	Change History: <ul style="list-style-type: none"> <li>• new since [SRS030]</li> </ul>

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
on		SalesTypeDescription	

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>locale</b>	LocaleDT	The locale for which a description is valid.
<b>description</b>	String(256)	The (localized) description of a sales type.
<b>maintainedInSoe</b>	boolean	This flag is set to true if the sales type description is not available in AMDS and is exclusively maintained in SOE. The flag is false by default.  Change History • new since [SRS004]

### 21.14.10 VehicleCoreData

Type: **Class**

Stereotype: «EntityObject»

This entity stores the mapping FIN/VIN and the first registration date of registered vehicles.

***Connections***

<b>Connector</b>	<b>Source</b>	<b>Target</b>	<b>Description</b>
<b>Association</b> references Source -> Destination	Role: Cardinality: 0..1 Type: VehicleCoreData	Role: fin Cardinality: 1 Type: Vehicle	
<b>Association</b> represents Source -> Destination	Role: Cardinality: 0..1 Type: VehicleCoreData	Role: fin Cardinality: 1..* Type: VehicleConfiguration	Reference to a Vechicle Configuration that is potentially stored in MYMVDC.

***Attributes***

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
<b>firstRegistrationDate</b> [0..1]	DateDT	Date of the vehicle's registration.

Attribute	Type	Notes
<b>vin</b>	VinDT «UniqueKey»	VIN (in contrast to the FIN!) of the vehicle. This attribute stores the VIN in case we get that information in addition to a FIN.
<b>modelSeries</b>	ModelSeriesDT	The vehicle's model series.  Change History: • new since [SRS025]

## 21.15 data types

Type: Package

Package: SOE

This section describes the data types used throughout this data model. It is used to uniformly describe common business types and to give a semantics behind a mere type instead of just writing „String“.

### Business Data Types

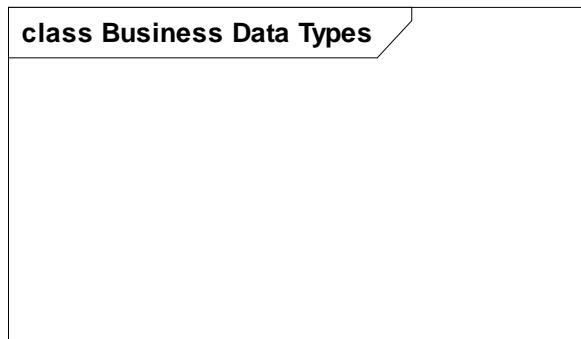


Figure 99: data types

### 21.15.1 BaumusterDT

Type: Class

Stereotype: «DataType»

Data type for a Baumuster.

Change History:

- [SRS045]: renamed baumuster7 to baumuster

---

#### Attributes

Attribute	Type	Notes
baumuster	String(8)	Baumuster (without any separators such as ".").  Change History: <ul style="list-style-type: none"><li>• [SRS045]: renamed baumuster7 to baumuster</li><li>• [SRS045]: modified length of baumuster from 7 to 8</li></ul>

## 21.15.2 BodyTypeDT

Type: Class

Stereotype: «DataType»

Data type for a Body Type (German: "Aufbauart").

Change History:

- new since [SRS030]

#### Attributes

Attribute	Type	Notes
bodyTypeld	String(10)	The body type of a vehicle.  Change History: <ul style="list-style-type: none"><li>• [SRS045]: modified length from 1 to 10</li></ul>

## 21.15.3 CountryCodeDT

Type: Class

Stereotype: «DataType»

Data type for country codes.

#### Attributes

Attribute	Type	Notes
countyCode	String(2)	two letter country code as specified by ISO 3166-1.

---

## **21.15.4 DateDT**

Type: **Class**

Stereotype: «DataType»

Data type for a date, e.g. only year, months and date, but without information of hours, minutes, etc.

## **21.15.5 EquipmentCodeDT**

Type: **Class**

Stereotype: «DataType»

Data type for a equipment code of a vehicle.

***Attributes***

Attribute	Type	Notes
equipmentCode	String(5)	String representing an equipment code.

## **21.15.6 FinDT**

Type: **Class**

Stereotype: «DataType»

Data type of a FIN of a vehicle.

***Attributes***

Attribute	Type	Notes
fin [17]	String	a FIN

---

## 21.15.7    **GemsOutletIdDT**

Type:            Class

Stereotype:    «DataType»

ID of the org unit in GEMS/GSSN. For the communication about MB organizational units, the GEMS OutletId (=GSSN “dcxGSSNOutletOutletId”, MBC POS “OutletOutletId”) is used within SOE.

### Attributes

Attribute	Type	Notes
<b>GemsOutletID</b>	String(9)	ID of the org unit in GEMS/GSSN. For the communication about MB organizational units, the GEMS OutletId (=GSSN “dcxGSSNOutletOutletId”, MBC POS “OutletOutletId”) is used within SOE.

## 21.15.8    **LocaleDT**

Type:            Class

Stereotype:    «DataType»

Locale for a field. E.g. "EN\_US", "DE\_DE", etc.

(as defined in request for Comments: 5646/BCP 47 for language and region)

### Attributes

Attribute	Type	Notes
<b>locale</b>	Strng(5)	

## 21.15.9    **ModelSeriesDT**

Type:            Class

Stereotype:    «DataType»

Model series as 3 alphanumeric characters. Eg. "203", "205", "A23"...

Change History:

- 
- new since [SRS004]

**Attributes**

Attribute	Type	Notes
<b>modelSeriesId</b>	String(3)	Id of the model series

### **21.15.10 NstDT**

*Type:*      **Class**

*Stereotype:* «DataType»

Data type for "National Sales Type".

**Attributes**

Attribute	Type	Notes
<b>NST</b>	String(3)	String with length of 3.

### **21.15.11 TimestampDT**

*Type:*      **Class**

*Stereotype:* «DataType»

Data type for a timestamp, i.e. a date incl. at least hours, minutes and seconds.

### **21.15.12 VinDT**

*Type:*      **Class**

*Stereotype:* «DataType»

Data type of a VIN of a vehicle.

---

**Attributes**

<b>Attribute</b>	<b>Type</b>	<b>Notes</b>
vin [17]	Sring	a VIN

## 22 User / Organization / Entitlements

Only users that manage static data access SOE directly through its user interface. All other access paths are through external interfaces from third party systems.

Authorization for access from third party systems takes place through technical users (one per external system). A SOE internal system or user specific authentication is not necessary and is not implemented apart from the technical user.

For direct access through SOE's user interface only two roles are used for authorization. A fine granular authorization does not happen.

### 22.1 Business Roles

Business Role	Description	Comment
MBconnect Admin - Product masterdata and rules	This role is for users that have the right to administer the product master data and rules for MBconnect.	
MBconnect Admin - Documents	This role is for users that have the right to administer the document master data for MBconnect.	
MBC.MBC_RETAIL	The retailer who is using POS.	
MBC.MBC_SUPPORT	The CAC support who is using POS.	
MBC.VEHICLE_MASTERUSER	The master user who is using MBconnect services.	
MBC.VEHICLE_SUBUSER	The sub user who is using MBconnect services.	
MBC.DFS	The user working for Daimler Financial Services and is entitled to separate a vehicle belonging to a defaulting debtor and to track the vehicle.	The role of the users that have the right to trigger the tracking of the vehicle. This is done by requesting the activation of specific services (e.g. Skip Loss).

Table 414: Business Roles

### 22.2 Application Roles

Application Role	Org Levels	Description	Comment
SOE_VEHICLE_MDM	HQ	MBconnect Admin for Product master data and service rules. Grants access to SOE and allows a user to administer master data regarding product master data and rules.	
SOE_DOC_MDM	HQ	MBconnect Admin for Documents. Grants access to SOE and allows a user to maintain document master data.	

Table 415: Application Roles

## 22.3 Assignment of Application Roles to Business Roles

Business Role	Application Role
MBconnect Admin - Product masterdata and rules	SOE_VEHICLE_MDM
MBconnect Admin - Documents	SOE_DOC_MDM

Table 416: Assignment Business Roles Entitlements

## 22.4 UI Entitlements

As access from third party systems to SOE are always from trusted systems inside Daimler, no special treatment for authentication or authorization is therefore provided for data access.

The following table shows the right (Access Right) of each dialog:

ResourceName (ResourceType: DLG)	SOE_DOC_MDM (Entitlement Name)		SOE_VEHICLE_MDM (Entitlement Name)	
	MDM (Ent. Source)	REGION (Ent. Source)	MDM (Ent. Source)	REGION (Ent. Source)
DLG_Login	WRITE	WRITE	WRITE	WRITE
DLG_Admin	WRITE	WRITE	WRITE	WRITE
DLG_InformOfMasterDataChange	NONE	WRITE	NONE	WRITE
DLG_ReplicationImportOverview	NONE	READ	NONE	READ
DLG_ReplicationExportOverview	READ	NONE	READ	NONE
DLG_ModelSeriesOverview	NONE	NONE	WRITE	READ
DLG_ModelSeriesDetail	NONE	NONE	WRITE	READ
DLG_SalesTypesOverview	NONE	NONE	WRITE	READ
DLG_SalesTypesDetail	NONE	NONE	WRITE	READ
DLG_BodyTypesOverview	NONE	NONE	WRITE	READ
DLG_BodyTypesDetail	NONE	NONE	WRITE	READ
DLG_EquipmentCodesOverview	NONE	NONE	WRITE	READ
DLG_EquipmentCodesDetail	NONE	NONE	WRITE	READ
DLG_ServiceAssignmentRuleOverview	NONE	NONE	WRITE	WRITE
DLG_ServiceAssignmentRuleDetail	NONE	NONE	WRITE	WRITE
DLG_GenericMasterDataOverview	NONE	NONE	WRITE	READ
DLG_GenericMasterDataDetail	NONE	NONE	WRITE	READ
DLG_ServiceAssignmentRulesTestSimulation	NONE	NONE	WRITE	WRITE
DLG_ServiceAssignmentRulesTestSimulation : VIN/FIN	NONE	NONE	NONE	WRITE
DLG_ServiceAssignmentRulesTestResults	NONE	NONE	WRITE	WRITE
DLG_ServicesOverview	NONE	NONE	WRITE	READ
DLG_ServiceDetail	NONE	NONE	WRITE	READ
DLG_ServiceMasterOverview	NONE	NONE	WRITE	READ

ResourceName (ResourceType: DLG)	SOE_DOC_MDM (Entitlement Name)		SOE_VEHICLE_MDM (Entitlement Name)	
	MDM (Ent. Source)	REGION (Ent. Source)	MDM (Ent. Source)	REGION (Ent. Source)
DLG_ServiceMasterDetail	NONE	NONE	WRITE	READ
DLG_ServiceCategoriesOverview	NONE	NONE	WRITE	READ
DLG_ServiceCategoryDetail	NONE	NONE	WRITE	READ
DLG_DocumentOverview	WRITE	READ	NONE	NONE
DLG_DocumentDefinition	WRITE	READ	NONE	NONE
DLG_UserAgreementServiceAssignment	WRITE	READ	NONE	NONE
DLG_UserAgreementServiceAssignmentOverview	WRITE	READ	NONE	NONE
DLG_PDFOverview	WRITE	READ	NONE	NONE
DLG_PDFDetail	WRITE	READ	NONE	NONE
DLG_DocumentBlockOverview	WRITE	READ	NONE	NONE
DLG_DocumentBlockDetail	WRITE	READ	NONE	NONE
DLG_DocumentTriggerAssignment	WRITE	READ	NONE	NONE
DLG_CustomTagOverview	WRITE	READ	NONE	NONE
DLG_CustomTagDetail	WRITE	READ	NONE	NONE
DLG_DocumentPreview	WRITE	READ	NONE	NONE
DLG_ImageOverview	WRITE	READ	NONE	NONE
DLG_ImageDetail	WRITE	READ	NONE	NONE
DLG_PrintApproval	NONE	WRITE	NONE	NONE
DLG_CountriesOverview	NONE	NONE	WRITE	READ
DLG_CountryDetail	NONE	NONE	WRITE	READ
DLG_ChangeSessionMenu	WRITE	NONE	WRITE	NONE
DLG_ManageChangeSession	WRITE	NONE	WRITE	NONE
DLG_TakeOverChangeSession	WRITE	NONE	WRITE	NONE

Table 417: UI Entitlements

## 22.5 Functional Entitlements

The following table shows the rights (Access Right) of each batch:

ResourceName (ResourceType: Batch)	MDM	REGION
AF_ImportSalesTypesAndGroupsBatch	WRITE	NONE
AF_ImportEquipmentCodesBatch	WRITE	NONE
AF_GenerateDocumentsPreviewArchive	WRITE	NONE
AF_SendNewLegalDocumentsBatch	NONE	WRITE
AF_SelectNewLegalDocumentsBatch	NONE	WRITE
AF_MigrateUserToNewUserAgreementBatch	NONE	WRITE
AF_UpdateServiceAvailabilityForUsers	NONE	WRITE
AF_UpdateReplicationsMasterDataBatch	NONE	WRITE
AF_UpdateSentLettersStatusBatch	NONE	WRITE

---

AF_ReportAboutMonthlyProcessedLetters	NONE	WRITE
---------------------------------------	------	-------

Table 418: Function Entitlements

---

## **23 Rollout Aspects**

### **23.1 Language maintenance**

It has to be ensured that the available languages maintained in the application configuration match the languages available in the neighboring systems MBC POS and MyMercedes.

Further the cluster CEX will provide the localized, permitted enumeration values with unique IDs for the specific customer data fields. Every unique ID – in case it will be used in different countries – has to have the same semantic for every localized value. The affected customer data fields will also be determined by cluster CEX during rollout.

### **23.2 Dictionary**

To ensure the homogenous use of terms and labels within the SOE specification and its application dialogs, a dictionary has been created. The dictionary is maintained in form of an Excel and can be found in the references chapter.

# 24 Error Message Response Code Mapping

## 24.1 Response Codes for Interface Errors

Table 419 contains the mapping between the error messages of interfaces and the corresponding response codes. The response codes are returned to the calling system (e.g. POS), if an error in the interface occurs.

The master of this table can be found in the MBconnect Confluence as attachment on this site (<https://s415vmmt060.detss.corpintra.net/confluence/display/MBC/SOE%3A+Spezifisch+e+Response+Codes>).

Explanation of column “Response Code Type”:

- “Common”: Used by SOE and other MBconnect systems like MyMercedes or CPD.
- “SOE”: Only used by SOE.

Message ID (SOE internal)	Response Code Type	Response Code (external/interface)	Identical to Response Code	Error Title
VEHPRO_005	Common	1		Vehicle/FIN not found
ACCDAS_002	Common	2		Customer (i.e. Userid) not found
SERMAN_008	Common	3		Invalid service ID
VEHPRO_006	Common	99		Error retrieving data from UVS
ACCDAS_004	Common	-	99	Error connecting to system CPD
GETDOC_003	SOE	100		User Agreement ID does not exist
VEHPRO_011	SOE	101		No country number found
GENDOC_006	SOE	102		Cannot find CustomTag
CONMAN_008	SOE	104		Inconsistent MasterData
ACCDAS_005	SOE	106		Unsupported Country
SERMAN_010	Common	108	3	Invalid market ID
CERMAN_019	Common		99	Error connecting to system CCM
GMDM_003	Common	109		Invalid Key
SERMAN_023	SOE	110		Invalid service master ID
CONMAN_009	SOE	111		Invalid combination of user ID and FIN
LICENSE_001	SOE	112		Invalid license ID
LICENSE_002	SOE	113		Invalid combination of user ID and license ID

Table 419: Mapping between Error Messages and Response Codes

---

## **25 Open Issues**

**No table of figures entries found.**

---

## 26 References

Reference	Name of referenced object
AS07	Target Business Process Model
SOE Dictionary	SOE Dictionary.xlsx

Table 420: References

---

# 27 Indexes

## 27.1 Figures

Figure 1: Overview SOE's functional interfaces and the related third party systems that are connected to them .....	20
Figure 2: Internal view of SOE showing the functional architecture and the decomposition into components.....	23
Figure 3: Formatting of addresses.....	25
Figure 4: Import and storage of vehicle master data in SOE .....	26
Figure 5: Maintenance of service assignment rules in SOE .....	28
Figure 6: Interplay of vehicle, user and service .....	31
Figure 7: The four MBconnect regions.....	37
Figure 8: Replication of master data .....	39
Figure 9: releasing a change session .....	42
Figure 10: Discard modifications inside change session .....	43
Figure 11: Taking over a change session .....	44
Figure 12: Retrieve information about what modifications have been done.....	45
Figure 13: creating a new model series .....	48
Figure 14: modifying existing equipment .....	49
Figure 15: deleting existing equipment .....	50
Figure 16: Example layout for a cover letter .....	54
<b>Figure 17:</b> Creating a new version of a document based on previous latest version .....	55
<b>Figure 18:</b> Creating a new version of a document block based on previous version .....	56
<b>Figure 19:</b> Creating a new version of a document template based on previous version .....	57
Figure 20: Document XML Definition Structure .....	57
Figure 21: Generation of the cover letter associated to the changed contract information .....	59
Figure 22: Document Definition Overview .....	62
Figure 23: Handling of profile data within SOE .....	72
Figure 24: Structure of profile data fields as provided to other systems .....	74
Figure 25: Illustration of basic dialog elements (marked in red).....	78
Figure 26: Translation table as used in all dialogs .....	79
Figure 27: DLG_Main .....	80
Figure 28: Exemplary dialog (DLG_DocumentDefinition) opened in read-only mode....	81
Figure 29: Dialog flow of SOE .....	84
Figure 30: Outlook of DLG_Info .....	85
Figure 31 DLG_ModelSeriesOverview .....	86
Figure 32: DLG_ModelSeriesDetail .....	88
Figure 33: DLG_SalesTypesOverview.....	91
Figure 34: DLG_SalesTypeDetail for passenger cars.....	94
Figure 35: DLG_SalesTypeDetail for transporters .....	94
Figure 36: DLG_BodyTypesOverview .....	97
Figure 37: DLG_BodyTypesDetail .....	99
Figure 38: DLG_EquipmentCodesOverview .....	101
Figure 39: DLG_EquipmentCodeDetail .....	103
Figure 40: DLG_ServiceCategoryOverview .....	147
Figure 41: DLG_ServiceCategoryDetail.....	149

---

Figure 42: DLG_ServiceAssignmentRulesOverview.....	151
Figure 43: DLG_ServiceAssignmentRuleDetail for passenger cars.....	154
Figure 44 DLG_TestServiceAssignmentRules .....	161
Figure 45 DLG_ServiceAssignmentRulesTestResults.....	165
Figure 46: DLG_ServicesOverview.....	234
Figure 47: DLG_ServiceDetail .....	237
Figure 48: DLG_ServiceMasterOverview .....	241
Figure 49: DLG_ServiceMasterDetail (Master Data) .....	243
Figure 50: DLG_ServiceMasterDetail (Authorization Data) .....	244
Figure 51: DLG_DocumentOverview .....	249
Figure 52: DLG_DocumentDefinition-UserAgreement.....	252
Figure 53: DLG_DocumentDefinition for non user agreements .....	252
Figure 54: DLG_UserAgreementServiceAssignmentOverview.....	257
Figure 55: DLG_UserAgreementServiceAssignment.....	258
Figure 56: DLG_PDFOverview .....	273
Figure 57: DLG_PDFDetail .....	275
Figure 58: DLG_DocumentBlockOverview .....	277
Figure 59: DLG_DocumentBlockDetail .....	279
Figure 60: DLG_DocumentTriggerAssignment .....	283
Figure 61: DLG_CustomTagOverview .....	285
Figure 62: DLG_CustomTagDetail.....	287
Figure 63: DLG_DocumentPreview .....	289
Figure 64: DLG_ImageOverview .....	291
Figure 65: DLG_ImageDetail .....	292
Figure 66: Batches.....	345
Figure 67: Component "Contract management" - Batches .....	387
Figure 68: DLG_PrintApproval.....	400
Figure 69: Email for functional support SOE.....	421
Figure 70: DLG_CountriesOverview .....	447
Figure 71: DLG_CountryDetail .....	449
Figure 72 Attributes and Relationships provided by the Dealer Runtime Directory (DRD) for a OrgUnit .....	463
Figure 73: DLG_GenericMasterDataOverview .....	466
Figure 74: DLG_GenericMasterDataDetail .....	468
Figure 75: DLG_ChangeSessionMenu (no opened change session) .....	473
Figure 76: DLG_ChangeSessionMenu (change session opened) .....	473
Figure 77: DLG_ManageChangeSession .....	475
Figure 78: DLG_ManageChangeSession – Delete Change Session PopUp.....	475
Figure 79: DLG_TakeOverChangeSession .....	477
Figure 80: DLG_ReplicationExportOverview .....	487
Figure 81: DLG_ReplicationImportOverview .....	494
Figure 82: Overview.....	521
Figure 83: Overview.....	522
Figure 84: GenericMasterData .....	523
Figure 85: MasterDataReplicationExport .....	524
Figure 86: MasterDataReplicationImport .....	527
Figure 87: MasterDataReplicationImport .....	529
Figure 88: ChangeSessionManagement .....	532
Figure 89: OutboundMessaging .....	538

---

Figure 90: MBconnectCountries .....	542
Figure 91: AccountDataSupport .....	544
Figure 92: AuthorizationSupport .....	561
Figure 93: ContractManagement .....	566
Figure 94: ServiceManagement.....	573
Figure 95: Documents .....	575
Figure 96: Documents .....	576
Figure 97: Services .....	602
Figure 98: VehicleProducts of the entities that belong to Vehicle Products. ....	615
Figure 99: data types .....	626

## 27.2 Tables

Table 1: Reading recommendations .....	16
Table 2: Changes towards conceptual document.....	19
Table 3: Exemplary selection of service assignment rules .....	29
Table 4: Example for default/fallback locales.....	33
Table 5: Example retrieving translations .....	33
Table 6: Overview master data changes relevant to adjacent systems .....	40
Table 7: Properties of Application Configuration.....	52
Table 8: Document Trigger and Channel association .....	61
Table 9: List of available data fields.....	67
Table 10: Mapping of CPD parameters to the SOE data model .....	73
Table 11: MBconnect servicemaster roles.....	76
Table 12: MBconnect servicemaster rights .....	76
Table 13: States of the common elements displayed by the “overview”-dialogs.....	81
Table 14: Content of DLG_Info .....	84
Table 15: Buttons and functions (DLG_ModelSeriesOverview) .....	87
Table 16 Form fields and front-end data objects (DLG_ModelSeriesOverview) .....	87
Table 17 Dialog Element States (DLG_ModelSeriesOverview) .....	88
Table 18: Buttons and functions (DLG_ModelSeriesDetail).....	89
Table 19: Form fields and front-end data objects .....	90
Table 20: Dialog field validations (DLG_ModelSeriesDetail) .....	90
Table 21: Dialog Elements State (DLG_ModelSeriesDetail).....	91
Table 22: Buttons and functions (DLG_SalesTypesOverview) .....	92
Table 23: Form fields and front-end data objects (DLG_SalesTypesOverview) .....	92
Table 24 Dialog Element States (DLG_SalesTypesOverview) .....	93
Table 25: Buttons and functions (DLG_SalesTypeDetail).....	95
Table 26: Form fields and front-end data objects (DLG_SalesTypeDetail).....	95
Table 27: Dialog field validations (DLG_SalesTypeDetail) .....	96
Table 28 Dialog Elements State (DLG_SalesTypeDetail).....	96
Table 29: Buttons and functions (DLG_BodyTypesOverview) .....	98
Table 30: Form fields and front-end data objects (DLG_BodyTypesOverview) .....	98
Table 31 Dialog Element States (DLG_BodyTypeOverview) .....	98
Table 32: Buttons and functions (DLG_BodyTypesDetail).....	99
Table 33: Form fields and front-end data objects (DLG_BodyTypesDetail) .....	100
Table 34: Dialog field validations (DLG_BodyTypesDetail) .....	100
Table 35 Dialog Elements States .....	100
Table 36: Buttons and functions .....	102

---

Table 37: Form fields and front-end data objects .....	102
Table 38 Dialog Element States (DLG_EquipmentCodesOverview) .....	102
Table 39: Buttons and functions .....	104
Table 40: Form fields and front-end data objects .....	104
Table 41: Dialog field validations .....	104
Table 42 Dialog Elements State (DLG_EquipmentCodesDetail) .....	105
Table 43: External interface input (IF_AMDS_GetSalesTypeGroups) .....	105
Table 44: External interface output (IF_AMDS_GetSalesTypeGroups) .....	106
Table 45: External interface input (IF_AMDS_GetSalesTypes) .....	107
Table 46: External interface output (IF_AMDS_GetSalesTypes) .....	107
Table 47: External interface input (IF_AMDS_getCodes) .....	108
Table 48: External interface output (IF_AMDS_getCodes) .....	108
Table 49: IF_ODC_GetVehicleData input.....	109
Table 50: Colour coding of the output table .....	109
Table 51: IF_ODC_GetVehicleData output.....	111
Table 52: IF_ODC_GetVehicleDataWithoutLocale input .....	112
Table 53: Colour coding of the output table .....	112
Table 54: IF_ODC_GetVehicleDataWithoutLocale output .....	113
Table 55: AF_GetModelSeries output.....	116
Table 56: AF_GetSalesTypes output.....	116
Table 57: AF_GetBodyTypes output.....	117
Table 58: AF_GetEquipment output .....	118
Table 59: AF_UpdateModelSeries input .....	119
Table 60: AF_UpdateSalesTypes input .....	120
Table 61: AF_UpdateBodyTypes input .....	121
Table 62: AF_UpdateEquipment input.....	122
Table 63: Application function input .....	123
Table 64: AF_DeleteModelSeries Input .....	124
Table 65: AF_SaveSalesType input .....	125
Table 66: AF_DeleteSalesTypes Input .....	126
Table 67: Application function input .....	127
Table 68: AF_DeleteBodyTypes Input .....	128
Table 69: Application function input .....	130
Table 70: AF_DeleteEquipmentCodes Input .....	131
Table 71: AF_FetchVehicleData Input .....	138
Table 72: AF_FetchVehicleData Output .....	138
Table 73: AF_FetchVehicleDataWithoutLocale Input .....	139
Table 74: AF_FetchVehicleDataWithoutLocale Output.....	139
Table 75: AF_UpdateFirstRegistrationDate Input .....	140
Table 76: AF_ResolveModelSeriesForSalesType Input .....	141
Table 77: AF_ResolveModelSeriesForSalesType Output .....	141
Table 78: AF_RetrieveModelSeriesForFIN Input .....	141
Table 79: AF_RetrieveModelSeriesForFIN Output .....	141
Table 80: AF_FilterAvailableModelSeries Input .....	142
Table 81: AF_FilterAvailableModelSeries Output .....	142
Table 82: AF_GetSalesTypesForCondition Input .....	143
Table 83: AF_GetSalesTypesForCondition Output.....	143
Table 84: AF_GetSalesTypesForModelSeriesId Input.....	143
Table 85: AF_GetSalesTypesForModelSeriesId Output .....	143

---

Table 86: AF_GetModelYear Input .....	144
Table 87: AF_GetModelYear Input .....	144
Table 88: Batches.....	144
Table 89: Messages of the component Vehicle Products .....	145
Table 90: Buttons and functions (DLG_ServiceCategoryOverview) .....	148
Table 91: Form fields and front-end data objects (DLG_ServiceCategoryOverview) ..	148
Table 92: Buttons and functions (DLG_ServiceCategoryDetail) .....	149
Table 93: Form fields and front-end data objects (DLG_ServiceCategoryDetail) .....	150
Table 94: Dialog field validations (DLG_ServiceCategoryDetail) .....	150
Table 95: Buttons and functions .....	151
Table 96: Form fields and front-end data objects .....	152
Table 97 DLG_ServiceAssignmentRulesOverview.....	153
Table 98: Buttons and functions .....	156
Table 99: Form fields and front-end data objects .....	159
Table 100: Dialog field validations .....	159
Table 101: Dialog Elements States.....	160
Table 102: Buttons and functions .....	161
Table 103: Form fields and front-end data objects .....	163
Table 104: Dialog field validations .....	163
Table 105: Dialog Elements State .....	164
Table 106: Buttons and functions .....	165
Table 107: Form fields and front-end data objects .....	167
Table 108: Dialog Elements State .....	168
Table 109: IF_SOE_CheckVehicle Input .....	170
Table 110: IF_SOE_CheckVehicle Output .....	171
Table 111: Table of exceptions (IF_SOE_CheckVehicle).....	171
Table 112: IF_SOE_GetRequiredFieldsUsedByServices input .....	172
Table 113: IF_SOE_GetRequiredFieldsUsedByServices output .....	173
Table 114: IF_SOE.GetServiceCategories input .....	173
Table 115: IF_SOE.GetServiceCategories output .....	174
Table 116: IF_SOE.GetServiceMasterData input .....	174
Table 117: IF_SOE.GetServiceMasterData output .....	176
Table 118: IF_SOE.GetServiceRelevantBodyTypesAndSalesTypes Input .....	177
Table 119: IF_SOE.GetServiceRelevantBodyTypesAndSalesTypes Output.....	178
Table 120: IF_SOE.GetServiceRelevantCodes Input .....	179
Table 121: IF_SOE.GetServiceRelevantCodes Output .....	179
Table 122: IF_SOE.GetServiceRelevantModelSeries Input .....	180
Table 123: IF_SOE.GetServiceRelevantModelSeries Output.....	180
Table 124: IF_SOE.GetServiceRightsByRole input .....	181
Table 125: IF_SOE.GetServiceRightsByRole output.....	181
Table 126: IF_CCM.GetServiceListRequest Input.....	182
Table 127: IF_CCM.GetServiceListRequest Output .....	182
Table 128: Table of exceptions (IF_CCM.GetServiceListRequest) .....	182
Table 129: AF_DeleteServiceAssignmentRuleChangeLog Input .....	188
Table 130: AF_DeleteServiceCategories Input.....	189
Table 131: AF_DeleteServiceCountryAssignmentChangeLog Input .....	189
Table 132: AF_DeleteServiceMasters input .....	190
Table 133: Table of exceptions (AF_DeleteServiceMasters) .....	190
Table 134 AF_GetAllServices Input.....	191

---

Table 135: AF_GetAllServices Output .....	191
Table 136: AF_GetAvailableServicesByConfiguration Input.....	192
Table 137: AF_GetAvailableServiceByConfiguration Output .....	192
Table 138: AF_GetMBcCountriesForServices Input .....	193
Table 139: AF_GetMBcCountriesForServices Output .....	193
Table 140: AF_GetModelSeriesForService Input .....	193
Table 141: AF_GetModelSeriesForService Output.....	193
Table 142: AF_GetServices output.....	196
Table 143: AF.GetServiceAssignmentRules output.....	197
Table 144: AF.GetServiceAssignmentRulesForService Input .....	197
Table 145: AF.GetServiceAssignmentRulesForService Output.....	197
Table 146: AF.GetServiceMasters output.....	199
Table 147: Exemplary selection of service assignment rules .....	203
Table 148: Selected service assignment rules .....	204
Table 149: AF_GetVehicleServiceInformation Input .....	208
Table 150: AF_GetVehicleServiceInformationByFin Output .....	209
Table 151: Example for overlapping/duplicate rules .....	210
Table 152: Application function input.....	210
Table 153: AF_SaveServiceCategory input.....	211
Table 154: AF_SaveServiceDetail linput .....	212
Table 155: AF_SaveServiceMaster input .....	212
Table 156: AF_RetrieveCountryCodeForVehicleProductMasterData Input .....	213
Table 157: AF_RetrieveCountryCodeForVehicleProductMasterData Output .....	213
Table 158: Application function input .....	214
Table 159: Application function output.....	215
Table 160: AF_RetrieveOldestServiceAssignmentRuleChangeLog Input .....	215
Table 161: Exemplary selection of service assignment rules .....	217
Table 162: Matched and not matched service assignment rules .....	217
Table 163: AF_RetrieveServicesAndMatchedRules Input.....	219
Table 164: AF_RetrieveServicesAndMatchedRules Output .....	219
Table 165: AF_RetrieveServiceCountryAssignmentChangeLog Input .....	220
Table 166: AF_RetrieveServiceCountryAssignmentChangeLog Output.....	220
Table 167: AF_TestServiceAssignmentRules Input.....	223
Table 168: AF_TestServiceAssignmentRules Output.....	224
Table 169: AF_UpdateServices input .....	225
Table 170: AF_UpdateServiceAssignmentRules input .....	228
Table 171: AF_UpdateServiceAssignmentRuleChangeLog Input .....	229
Table 172: AF_UpdateServiceCategories input.....	229
Table 173: AF_UpdateServiceCountryAssignmentChangeLog Input .....	230
Table 174: AF_UpdateServiceMasterss input .....	232
Table 175: Messages of the component Services .....	233
Table 176: Buttons and functions (DLG_ServicesOverview) .....	235
Table 177: Form fields and front-end data objects .....	235
Table 178: Buttons and functions (DLG_ServiceDetail).....	237
Table 179: Form fields and front-end data objects (DLG_ServiceDetail).....	239
Table 180: Dialog field validation (DLG_ServiceDetail) .....	240
Table 181: Dialog Elements States (DLG_ServiceDetail).....	240
Table 182: Buttons and functions (DLG_ServiceMasterOverview) .....	241
Table 183: Form fields and front-end data objects (DLG_ServiceMasterOverview) ....	242

---

Table 184: Buttons and functions (DLG_ServiceMasterDetail).....	245
Table 185: Form fields and front-end data objects (DLG_ServiceMasterDetail) .....	247
Table 186: Dialog Elements State (DLG_ServiceMasterDetail).....	248
Table 187: Form fields and front-end data objects .....	249
Table 188: Form fields and front-end data objects .....	255
Table 189: Input parameters.....	260
Table 190: Output parameters .....	260
Table 191: AF_GetUserAgreementServiceAssignments output .....	261
Table 192: AF_UpdateUserAgreementServiceAssignments input .....	262
<b>Table 193: AF_SaveService Input.....</b>	<b>262</b>
Table 194: Application function input .....	263
Table 195: AF_GetServicesByUserAgreements Input.....	268
Table 196: AF_GetServicesByUserAgreements Output .....	268
Table 197 Error Messages of component “Service Management”.....	271
Table 198: Buttons and functions .....	291
Table 199: Form fields and front-end data objects .....	292
Table 200: Buttons and functions .....	293
Table 201: Form fields and front-end data objects .....	294
Table 202: External interface input .....	295
Table 203: External interface output .....	295
Table 204: IF_SOE_GetFormattedAddress Input.....	297
Table 205: IF_SOE_GetFormattedAddress Output .....	297
Table 206: IF_SOE_GetUserAgreementMasterData Input.....	298
Table 207: IF_SOE_GetUserAgreementMasterData Output .....	298
Table 208: IF_SOE_GetTermsOfUseMasterDataForCountry Input.....	299
Table 209: IF_SOE_GetTermsOfUseMasterDataForCountry Output .....	299
Table 210: AF_GetDocumentTemplates output .....	303
Table 211: AF_GetDocuments output .....	303
Table 212: AF_GetDocumentBlocks output.....	304
Table 213: AF_GetTrigger2DocAssignments output .....	305
Table 214: AF_GetCustomTags output .....	305
Table 215: AF_GetDocumentImage output .....	306
Table 216: AF_UpdateDocumentTemplates input.....	306
Table 217: AF_UpdateDocuments input.....	308
Table 218: AF_UpdateDocumentBlocks input .....	308
Table 219: AF_UpdateTrigger2DocAssignments input.....	309
Table 220: AF_UpdateCustomTags input.....	310
Table 221: AF_UpdateDocumentImages input .....	311
Table 222: DocumentTriggerEnum and PerformedActionEnum mapping .....	312
Table 223: AF_LogChangedServiceAvailability Input .....	317
Table 224: AF_GeneratePDF - Input.....	319
Table 225: AF_GeneratePDF - Output .....	319
Table 226: AF_GetLegalDocuments Input.....	325
Table 227: AF_GetLegalDocuments Output.....	325
Table 228: AF_GetDocumentsForDocTrigger Input .....	329
Table 229: AF_GetDocumentsForDocTrigger Output.....	330
Table 230: IF_SOE_GetFormattedAddress Input .....	332
Table 231: IF_SOE_GetFormattedAddress Output .....	332
Table 232: AF_GetServicesByUserAgreements Input.....	335

---

Table 233: AF_GetServicesByUserAgreements Output .....	335
Table 234: AF_CollectLegalTitles input .....	342
Table 235: AF_CollectLegalTitles output .....	342
Table 236: IF_CPD_updateMbcServiceAvailability Input.....	349
Table 237: IF_SOE_InformOfAccountDeletion Input .....	351
Table 238: IF_SOE_InformOfCustomerDataChange Input.....	353
Table 239: IF_SOE_InformOfCustomerDataChange Output.....	354
Table 240: IF_SOE_InformOfCustomerDataChange exceptions .....	354
Table 241: IF_SOE_InformOfInitialServiceActivation Input .....	355
Table 242: IF_SOE_InformOfVehicleRegistration Input .....	356
Table 243: IF_SOE_InformOfVehicleRegistration Output.....	357
Table 244: IF_SOE_InformOfVehicleSeparation Input .....	358
Table 245: IF_SOE_InformOfVehicleSeparation output .....	359
Table 246: IF_SOE_InformOfVehicleSeparation exceptions .....	360
Table 247: IF_SOE_SetUserAgreementStateOfConsent input .....	361
Table 248: IF_SOE_SetUserAgreementStateOfConsent output .....	361
Table 249: IF_SOE_SetUserAgreementStateOfConsent exceptions .....	362
Table 250: AF_DetermineSummaryForUser Input .....	365
Table 251: AF_DetermineSummaryForUser Output.....	366
Table 252: AF_InformOfMasterDataChangesAffectedByServiceEnablement Input .....	370
Table 253: <b>Error! Reference source not found.</b> input.....	372
Table 254: <b>Error! Reference source not found.</b> output .....	373
Table 255: Exceptions of <b>Error! Reference source not found.</b> .....	374
Table 256 AF_SeparateVehicleFromUser – Input .....	375
Table 257 AF_SeparateVehicleFromUser – Exceptions.....	375
Table 258: Table 257 AF_SeparateVehicleFromUser – Exceptions.....	375
Table 259: AF_ input .....	376
Table 260: AF_ output .....	377
Table 261: <b>Error! Not a valid bookmark self-reference.</b> input .....	378
Table 262: <b>Error! Not a valid bookmark self-reference.</b> output .....	378
Table 263: Exceptions of <b>Error! Not a valid bookmark self-reference.</b> .....	378
Table 264: AF_UpdateAvailabilititesAndLicensesForUser input .....	381
Table 265: AF_UpdateAvailabilititesAndLicensesForUser exceptions.....	382
Table 266: Component "Contract management" - error messages .....	387
Table 267: IF_CPD_InformOfLicenseModification Input.....	388
Table 268: IF_CPD_InformOfLicenseRevocation Input.....	389
Table 269: IF_SOE_GetActiveLicensesByID Input.....	389
Table 270: IF_SOE_GetActiveLicensesByID Output .....	389
Table 271: IF_SOE_ModifyLicenses Input .....	390
Table 272: IF_SOE_InformOfServiceLicenseExpiration Input .....	391
Table 273: AF_CreateLicenses Input .....	393
Table 274:AF_CreateLicensesAndLicenseOrders Output .....	393
Table 275: AF_GetLicensesForUser Input .....	394
Table 276: AF_GetLicensesForUser Output.....	394
Table 277: AF_ReassignLicenses Input .....	396
Table 278: AF_SeparateLicenseOrdersFromUser Input.....	397
Table 279: AF_GetLicenses Input .....	397
Table 280: AF_UpdateLicensesForUserAndVehicle input.....	399
Table 281: Messages of the component Licenses.....	399

---

Table 282: Buttons and functions (new dialogue) .....	400
Table 283: Form fields and front-end data objects (new dialogue) .....	401
Table 284: IF_SEO_GetAvailableUserAgreementsForUser Input .....	402
Table 285: IF_SEO_GetAvailableUserAgreementsForUser Output .....	402
Table 286: IF_SOE_GetAvailableUserAgreementsForVehicle Input .....	403
Table 287: IF_SOE_GetAvailableUserAgreementsForVehicle Output .....	404
Table 288: IF_SOE_GetAvailableUserAgreementsForVehicle exceptions .....	404
Table 289: IF_SOE_GetContractData Input .....	405
Table 290: IF_SOE_GetContractData Output .....	405
Table 291: IF_GetLegalDocuments Input .....	406
Table 292: IF_GetLegalDocuments Output .....	407
Table 293: Table of exceptions .....	407
Table 294: IF_SOE_GetRequiredUserAgreementForService Input .....	407
Table 295: IF_SOE_GetRequiredUserAgreementForService Output .....	408
Table 296: Exceptions .....	408
Table 297: IF_SOE_GetServicesCoveredByUserAgreement Input .....	408
Table 298: IF_SOE_GetServicesCoveredByUserAgreement Output .....	409
Table 299: IF_SOE_GetServicesCoveredByUserAgreement exceptions .....	409
Table 300: AF_GetAllUserAgreementConsentsAndAssociations input .....	410
Table 301: AF_GetAllUserAgreementConsentsAndAssociations output .....	410
Table 302: AF_DeleteUserAgreementConsents input .....	411
Table 303: AF_DeleteUserAgreementConsents output .....	411
Table 304: AF_GetAllUserAgreementsWithStateForUser input .....	412
Table 305: AF_GetAllUserAgreementsWithStateForUser output .....	412
Table 306: AF.GetUserAgreementConsents input .....	419
Table 307: AF.GetUserAgreementConsents output .....	419
Table 308: AF_SetUserAgreementConsent input .....	424
Table 309: AF_DeleteAvailabilitiesForUserAndVehicle input .....	426
Table 310: AF_DeleteAvailabilitiesForUserAndVehicle output .....	427
Table 311: AF_GetAvailabilitiesForUser input .....	427
Table 312: AF_GetAvailabilitiesForUser output .....	427
Table 313: AF_UpdateAvailabilitiesForUserAndVehicle input .....	428
Table 314: AF_UpdateAvailabilitiesForBusinessPartnerAndVehicle input .....	429
Table 315: AF_DeleteAvailabilitiesForBusinessPartnerAndVehicle input .....	429
Table 316: AF_GetOrCreateBusinessPartnerProxy input .....	431
Table 317: AF_GetOrCreateBusinessPartnerProxy output .....	431
Table 318: AF_GetOrCreateBusinessPartnerProxy input .....	432
Table 319: AF_GetOrCreateBusinessPartnerProxy output .....	432
Table 320: AF_AssignUserProxyToVehicleProxy input .....	433
Table 321: AF_AssignUserProxyToVehicleProxy output .....	433
Table 322: AF_DeleteUserProxy input .....	434
Table 323: AF_DetachUserProxyFromVehicleProxy input .....	434
Table 324: AF_DetachUserProxyFromVehicleProxy exceptions .....	435
Table 325: AF_DetachBusinessPartnerProxyFromVehicleProxy input .....	435
Table 326: AF_GetOrCreateOrUpdateUserProxy input .....	436
Table 327: AF_GetOrCreateOrUpdateUserProxy output .....	436
Table 328: AF_GetOrCreateOrUpdateUserProxy exceptions .....	436
Table 329: AF_GetVehicleProxy input .....	436
Table 330: AF_GetVehicleProxy output .....	436

---

Table 331: AF_GetVehicleProxiesForUser input .....	437
Table 332: AF_GetVehicleProxiesForUser output.....	437
Table 333: AF_GetVehicleProxiesAndAssociatedUserForModelSeries output .....	437
Table 334: Component "Context Proxy" – error messages .....	438
Table 335 IF_SOE_GetProfileFieldsUsedByCountry Input.....	439
Table 336 IF_SOE_GetProfileFieldsUsedByCountry Output .....	441
Table 337: IF_CPD_getUserProfile Input .....	441
Table 338: IF_CPD_getUserProfile Output.....	441
Table 339: Output parameters.....	443
Table 340: AF_GetUserProfile input .....	443
Table 341: AF_GetUserProfile output.....	443
Table 342: Messages of Component "AccountDataSupport".....	446
Table 343: Buttons and functions .....	448
Table 344: Form fields and front-end data objects .....	448
Table 345: Buttons and functions .....	449
Table 346: Form fields and front-end data objects .....	450
Table 347: Dialog element states .....	450
Table 348: External interface input .....	451
Table 349: External interface output.....	451
Table 350: AF_GetCountries output .....	452
Table 351: AF_UpdateCountries input.....	453
Table 352: AF_IsMCconnectCountry input .....	453
Table 353: AF_IsMCconnectCountry output.....	453
Table 354: AF_GetMBconnectCountries output .....	454
Table 355: AF_GetOutletOutletIdsForAllMbcCountries output .....	454
Table 356: Table of error messages .....	455
Table 357: AF.GetServiceAuthorizations output.....	457
Table 358: AF_UpdateServiceAuthorizations input .....	458
Table 359: AF_GetRightsForResource input.....	458
Table 360: AF_GetRightsForResource output.....	458
Table 361: AF_SetRightsForResource input .....	459
Table 362: AF_SetRightsForResource exceptions .....	459
Table 363: AF_GetRightsByServiceMasterAndRole input .....	460
Table 364: AF_GetRightsByServiceMasterAndRole output.....	460
Table 365: Component "Authorization Support" error messages.....	461
Table 366: AF_GetMpcByCountry input .....	464
Table 367: AF_GetMpcByCountry input .....	464
Table 368: Buttons and functions (DLG_GenericMasterDataOverview) .....	466
Table 369: Form fields and front-end data objects (DLG_GenericMasterDataOverview) .....	467
Table 370: Dialog Elements State (DLG_GenericMasterDataOverview) .....	467
Table 371: Buttons and functions (DLG_GenericMasterDataDetail) .....	468
Table 372: Form fields and front-end data objects (DLG_GenericMasterDataDetail) .....	468
Table 373: Dialogue field validations (DLG_GenericMasterDataDetail) .....	469
Table 374: Dialog Elements State (DLG_GenericMasterDataDetail).....	469
Table 375: IF_SOE_GetGenericMasterDataForKey input .....	469
Table 376: IF_SOE_GetGenericMasterDataForKey output.....	469
Table 377: IF_SOE_GetGenericMasterDataForKey exceptions.....	469
Table 378: AF_SaveGenericMasterData input .....	471

---

Table 379: AF_GetGenericMasterData output .....	471
Table 380: AF_UpdateGenericMasterData input.....	472
Table 381: Messages of the component Generic Master Data Management .....	472
Table 382: Buttons and functions (DLG_ChangeSessionMenu) .....	473
Table 383: Form fields and front-end data objects (DLG_ChangeSessionMenu).....	473
Table 384: Dialog Elements State (DLG_ChangeSessionMenu) .....	474
Table 385: Buttons and functions (DLG_ManageChangeSession) .....	475
Table 386: Form fields and front-end data objects (DLG_ManageChangeSession)....	476
Table 387: Buttons and functions (DLG_TakeOverChangeSession).....	477
Table 388: Form fields and front-end data objects (DLG_TakeOverChangeSession).477	477
Table 389: Application function input (AF_IsChangeSessionOpenedForUser) .....	479
Table 390: Application function output (AF_IsChangeSessionOpenedForUser) .....	479
Table 391: Application function (IIF_GetChangeSessionForUser) .....	479
Table 392: Application function output (IIF_GetChangeSessionForUser) .....	480
Table 393: Application function input (IIF_OpenNewChangeSession) .....	480
Table 394: Application function output (IIF_OpenNewChangeSession) .....	480
Table 395: Trigger SOE Regions with changed master data .....	485
Table 396: Error messages of component Component “Users and Organizations” .....	486
Table 397: Buttons and functions (new dialogue) .....	488
Table 398: Form fields and front-end data objects (new dialogue) .....	488
Table 399: IF_SOE_UpdateReplicationStatus input.....	489
Table 400: AF_TriggerReplications input.....	491
Table 401: Receiving all replication relevant change master data entities.....	491
Table 402: Batches.....	493
Table 403: Buttons and functions (new dialogue) .....	494
Table 404: Form fields and front-end data objects (new dialogue) .....	495
Table 405: IF_SOE_UpdateMasterData input .....	506
Table 406: IF_SOE_UpdateReplicationStatus input .....	506
Table 407: AF_InformOfMasterDataChange input .....	510
Table 408: Order and way how to update master data .....	511
Table 409: Informing adjacent systems about master data changes .....	511
Table 410: AF_UpdateReplicationMasterDataBatch input.....	512
Table 411: Batches.....	512
Table 412: IF_Printserver_SendLetter Input.....	515
Table 413: Batches (new batches) .....	520
Table 414: Business Roles .....	632
Table 415: Application Roles .....	632
Table 416: Assignment Business Roles Entitlements .....	633
Table 417: UI Entitlements .....	634
Table 418: Function Entitlements .....	635
Table 419: Mapping between Error Messages and Response Codes .....	637
Table 420: References .....	639