energy shop

www.newenergy-bremen.de

Inhaltsverzeichnis

Home

- Überblick 24
  - Systemanforderungen 24
  - Funktionsweise 25

Erste Schritte

- Erste Schritte 26
  - Empfohlene Einrichtungsreihenfolge 26
    - Phase 1: Grundlagen (Pflicht) 26
    - Phase 2: Kundenverwaltung (empfohlen) 26
    - Phase 3: Optionale Erweiterungen 26
  - Feature-Matrix 27
    - Kernfunktionen (immer enthalten) 27
    - Funktionen mit zusätzlichen Modulen 27
  - Hilfreiche Werkzeuge 28

Checkliste

- Checkliste für Installation und Einrichtung 30

Installation

- Installation 32
  - WinLine 32
    - Einführung zur Bestellungsynchronisierung 32
    - Grundsätzliches 32
    - Web-Vorlage 33
      - Setzen der Vorlage als WebService-Vorlage 35
      - Anpassen der WebService-Vorlage 36
    - Config 38
      - Sammelkonto 38
      - Vorlagen 38
        - Adressen 38
      - Belegstufe 39
      - Hauptkonto für Belege 39
      - Mapping - order.yml 41
- OrderSync Einrichtung mit Action Server 41
  - Makro für Orders einlesen 41
  - XML-Vorlagen 41
  - Bestellungen als Batch oder in Einzeldateien 41
  - Action Server als Dienst starten/einrichten (siehe auch ggf. Whitepaper “Action Server als Dienst”) 42
  - Action Server Definition 42
  - Action Server Dienst regelmäßig neustarten 43
    - Fehlerüberwachung 43

- OrderSync mit EXIM 43
  - Konfiguration in der Schnittstelle 44
    - Config 44
  - Tabellen durch WinLine anlegen lassen 44
  - BatchBeleg 44
- Bestellimport Plugins 45
  - Rechnungskauf 45
- Weitere Einrichtungen innerhalb der WinLine 46
  - Beschlagwortung 46
  - LibreOffice Installation - Beschreibungstexte konvertieren (optional) 46
  - Konfiguration und Mapping 47
  - Management Interface als Windows-Service 47
  - Aufgabenplaner von Windows 48
  - Sync Parameter 49
    - Benutzung der Parameter 49
  - Shopware 51
    - Grundeinstellungen 51
      - Shopware 6 API-Integration 51

Update

- Update 53
  - Migration Management Interface (Flask → Django 6) 53
    - management.ini anpassen 53
    - [SHOPS]-Sektion aktualisieren 53
    - Startbefehl 54
    - Installation als Windows-Service 54
    - Unveränderte Sektionen 54

- Beispiel: management.ini nach Migration 54
- Cache update to version 0.1 55

Konfiguration

- Konfiguration 57
  - Shop-Verbindung 57
  - Shopware 6 58
  - WinLine WebService 58
  - WinLine Mandant & Datenbank 58
  - SQL-Verbindung 60
  - Shop-Felder 61
  - Bestellungsimport 61
    - Lieferung 62
    - Bezahlung 62
  - Shop-Eigenschaften 62
  - E-Mail-Benachrichtigungen 62
  - Archiv 63
  - Bildkonvertierung 63
  - Artikeleigenschaften 64
  - Cross-Selling 64
  - Dynamisches Cross-Selling 64
  - Mehrsprachigkeit 65
  - Mapping 65
  - Sonstige Einstellungen 65
  - Feature-Flags 66

Funktionen

Artikel 69

- Webartikel 69
  - Überblick 69
  - Workflow 69
  - Verwendung 70
    - Differenzieller Sync 70
    - Vollsync 70
    - Einzelartikel-Sync 70
  - WinLine 70
    - Einrichtung 70
    - Verwendung 70
  - Schnittstelle 71
  - Steuersätze 71

Artikelgrafik 73

- Artikelgrafik & Downloads 73
  - Überblick 73
  - Workflow 73
    - Bilder synchronisieren 73
    - Downloads synchronisieren 73
  - Shopware 74
  - WinLine - Archiv II 74
    - WL Schlagwort anlegen 76
  - Schnittstelle 76
    - Maximale Größe setzen 77

- WinLine - Artikel Bild 77
- Bilder löschen 79

Crossselling 81

- Cross-Selling 81
  - Überblick 81
  - Workflow 81
    - Manuelles Crossselling 81
    - Dynamisches Crossselling 81
  - Cross-Selling Anzeige in Shopware 82
  - Cross-Selling per Ähnliche Artikel 82
    - Zur Einrichtung in der WinLine 82
    - Den Crossselling-Artikel als Ersatzartikel zuweisen 83
    - Prioritäten–Steuerung „Ähnliche Artikel“ und Zubehör-Artikel“ 83
  - Dynamisches Crossselling / Zusatzfelder 84

Disposition 86

- Lieferzeiten + Disposition 86
  - Überblick 86
  - Workflow 86
  - WinLine verfügbarer Lagerbestand 86
  - WinLine 87
  - Schnittstelle 87
- Bestände mit und ohne Handelsstücklisten 88
  - Schnittstelle 88
    - Mapping 88

Eigenschaften 91

- Überblick 91
- Workflow 91

- WinLine 92
  - Einrichtung 92
  - Verwendung 93
- Shop 93
  - Einrichtung 93
- Schnittstelle 93
  - Einrichtung 93

Feldzuordnung 95

- Überblick 95
- Workflow 95
- Umsetzungstabellen 96

Gutscheine 98

- Überblick 98
- Workflow 98
- Anlage in Shopware 98
- Bestellung in Shopware 99
- Gutschein-Artikel in der WinLine 100

Handelsstücklisten 101

- Überblick 101
- Workflow 101
- WinLine 101
  - Einrichtung 101
  - Verwendung 101
- Schnittstelle 102
  - Einrichtung 102

Kategorien 104

- Überblick 104
- Workflow 104
  - Variante 1: Artikeluntergruppen 104
  - Variante 2: Mehrfachauswahl aus Eigenschaft 104
- Aus Artikeluntergruppen 105
  - Filter über Eigenschaften Plugin 105
- Mehrfachauswahl aus Eigenschaft 105
- Artikeluntergruppen in Eigenschaften übertragen 106
  - Makro Autoanlage für Artikel-Kategorien 106
  - Kategorien Artikel zuweisen 106

Konfigurator 108

- Überblick 108
- Workflow 108
- Shopware 108
  - Einrichtung 108
- Screenshots 109

Kundenimport 113

- Kundenimport Shopware to WinLine 113
  - Überblick 113
    - Plugin zur Filterung von Kundenimporten 113
    - Es gibt vier Möglichkeiten mit den Kundendaten aus Shopware umzugehen. 114
  - 1. Import mit Kundennummer aus SW 114
  - 2. Import ohne Kundennummer 115
  - 3. Sammelkonto 115
  - 4. Sammelkonto fallback 116
  - Umsetzungstabellen für Kundenimport 116

- Existierende Kunden 117
- Kundenimport Per Mapping 118
- Kunden als Kontakt/Ansprechpartner 118
- Kunden Datenbankimport 118

Kundensynchronisation 120

- Kundensync 120
  - Überblick 120
  - Typischer Ablauf 120
    - Einstellungen Config 120
    - Einstellungen Interface (Button): 121
    - Export/Import 121
    - KdNr.-Vergabe in Shopware 122
    - WL-KundenNr. -> Shopware 123
  - WL-Kontakt -> Shopware 123

Losgröße 126

- Überblick 126
- WinLine 126
  - Einrichtung 126
  - Verwendung 126
- Shop 127
- Schnittstelle 128

Managementinterface 130

- Shop Management Interface 130
  - Rahmendaten 130
    - management.ini 130
      - Beispiel: Einen Shop hinzufügen 130
      - Beispiel: Mehrere Shops 130

- Beispiel: Shop-Einstellungen 131
- Beispiel: Sync-Bezeichnungen 133
- Beispiel: Featured-Zeile + Restbereich 134
- Beispiel: Drei Gruppen mit Wildcard in der Mitte 134
- Vollständige Beispiel management.ini 135
- 2. Geplante Syncs (Scheduler) 137
- Scheduler verwalten 137
- Standard-Zeitplan laden 138
- 3. Einzelsync 139
- Warteschlange 139
- Sync-Fortschritt 139
- Sync-Fehler-Management 139
- Scheduler-Events-Dashboard 140
- Konfigurationseditor 140
- Log-Ansicht 141

Mehrsprachigkeit 142

- Überblick 142
- Workflow 142
- Bezeichnung und Beschreibung 143
- Kategorie und Eigenschaften 143
- Eigenschaften 144
- Shopware 144
- Config 144
- Mapping 145

Mindestbestellmenge 147

- Mindestbestellmenge / Maximalbestellmenge 147
- Überblick 147

- WinLine 147
  - Einrichtung 147
  - Verwendung 147
- Shop 148
- Schnittstelle 148

Preise 150

- Überblick 150
- Workflow 150
- Verwendung 150
- Preisliste 151
- Generelles zur Preisübermittlung 151
  - Kundenindividuelle Preise in Shopware 6 151
  - Rabattmatrix 151
    - Rabattmatrix zusammenfassen 151
- Kundengruppen Preise 152
- Brutto/Netto-Flag Preisliste 152
- Pseudo- /rabattierter Preis 152
- Preise mit überschneidender Gültigkeit 153
- Preisduplikate entfernen 153

Statusupdate & Tracking 155

- Statusupdates 155
  - Überblick 155
  - Workflow 155
    - Bestellimport 155
    - Sendungsverfolgung 155
  - Tracking 156

- Statuswechsel 156
  - Statusarten in Shopware 6 157
  - Beispielkonfiguration 157
  - Weitere Belegarten 157

Umsetzungstabellen 159

- Überblick 159
- Einrichtung - einfache Umsetzung 159
- zweifache Umsetzung 160

Varianten/Ausprägungen 162

- Ausprägungen/Varianten 162
  - Überblick 162
  - Workflow 162
  - Ausrägungsartikel mit eigenen Bilder 163
  - inaktive Ausrägungsartikel 164
  - Mapping (EANs Ausrägungsartikel) 164
  - WL Lagerausprägungen 165
- Ausprägungen durch Eigenschaften 165

Sonstiges

Jahreswechsel 168

- 1. Pausieren / beenden automatischer Datenabgleiche in der Windows-Aufgabenplanung 168
- 2. Sofern der WinLine-WebService nicht verwendet wird 168
- 3. Jahresabschluss in der WinLine 168
- 4. Nach dem Jahreswechsel 168
  - Im Fall von Import per Actionserver 168

- Im Fall von Import per WebService 169

Logging 171

- Überblick 171
- Manuelle Konfiguration 171
- E-Mail-Benachrichtigungen 172
- Plugin-Logger / Debug-Modus 172

FAQ

- FAQ 174
  - Problembehebung 174
    - Die automatische Synchronisierung funktioniert nicht mehr 174
      - Webservice bzw. WinLine Actionserver neustarten 174
      - Aufgabenplanung prüfen 174
      - Taskmanager prüfen 174
      - Logdateien prüfen 174
    - Bestellungen werden nicht importiert 174
    - Sync-Fehler nachverfolgen 175
    - Artikel werden nicht synchronisiert 175
    - Bild in Shopware ist defekt / fehlerhaft 175

Änderungshistorie

- Release Notes 177
  - v3.2.0 177
    - Neue Funktionen 177
    - Update-/Installationshinweise 178

- Fehlerbehebungen 178
- Sonstiges 179

- v3.1.0 179
  - Neue Funktionen 179
  - Fehlerbehebungen 180

- v3.0.3 180
  - Fehlerbehebungen 180

- v3.0.2 181
  - Sicherheits-Updates 181
  - Fehlerbehebungen 181
  - Sonstiges 181

- v3.0.1 181
  - Fehlerbehebungen 181
  - Sonstiges 181

- v3.0.0 182
  - Neue Funktionen 182
  - Update-/Installationshinweise 182
  - Fehlerbehebungen 182
  - Sonstiges 183
  - Major Release 183

- v2.12.4 183
  - Fehlerbehebungen 183

- v2.12.3 183
  - Fehlerbehebungen 184

- v2.12.2 184
  - Fehlerbehebungen 184
  - Sonstiges 184

- v2.12.1 184
  - Fehlerbehebungen 184
- v2.12.0 185
  - Neue Funktionen 185
  - Fehlerbehebungen 185
  - Sonstiges 185
- v2.11.6 185
  - Fehlerbehebungen 185
  - Sonstiges 185
- v2.11.5 185
  - Fehlerbehebungen 186
- v2.11.4 186
  - Fehlerbehebungen 186
- v2.11.3 186
  - Fehlerbehebungen 186
- v2.11.2 186
  - Fehlerbehebungen 186
- v2.11.1 186
  - Fehlerbehebungen 187
- v2.11.0 187
  - Neue Funktionen 187
  - Fehlerbehebungen 187
  - Sonstiges 187
- v2.10.1 188
  - Fehlerbehebungen 188
- v2.10.0 188
  - Neue Funktionen 188
  - Fehlerbehebungen 188

- Sonstiges 188
- v2.9.6 189
  - Prelude 189
  - Fehlerbehebungen 189
- v2.9.5 189
  - Fehlerbehebungen 189
- v2.9.4 189
  - Fehlerbehebungen 189
  - Sonstiges 189
- v2.9.3 189
  - Fehlerbehebungen 190
- v2.9.2 190
  - Fehlerbehebungen 190
- v2.9.1 190
  - Fehlerbehebungen 190
- v2.9.0 190
  - Neue Funktionen 190
- v2.8.1 191
  - Fehlerbehebungen 191
  - Sonstiges 191
- v2.8.0 191
  - Neue Funktionen 191
  - Fehlerbehebungen 191
  - Sonstiges 192
- v2.7.0 192
  - Neue Funktionen 192
  - Fehlerbehebungen 192

- v2.6.0 192
  - Prelude 192
  - Neue Funktionen 192
  - Nicht mehr unterstützte Funktionen 193
  - Fehlerbehebungen 193
  - Sonstiges 194
- v2.5.0 194
  - Neue Funktionen 194
  - Fehlerbehebungen 195
- v2.4.2 195
  - Fehlerbehebungen 195
  - Sonstiges 195
- v2.4.1 195
  - Fehlerbehebungen 195
- v2.4.0 195
  - Neue Funktionen 196
  - Fehlerbehebungen 196
  - Sonstiges 196
- v2.3.0 196
  - Neue Funktionen 196
  - Fehlerbehebungen 197
  - Sonstiges 197
- v2.2.1 197
  - Sicherheits-Updates 197
  - Fehlerbehebungen 197
- v2.2.0 197
  - Prelude 197
  - Neue Funktionen 197

- 429 - Generelle Grundfunktionen für Shopware 6 Funktionieren nun. 197
- 444 Daten für das Management-Interface werden Userverzeichnis 198
- 542 Mapping-Dateien können nun über die erweiterte Ansicht 198
- 546 - Ist die option 'skip_inaktiv_variants' in der Konfiguration 198
- 551 Über den Aufruf 198
- 575 Es kann in der Umsetzungstabelle nun ein Wildcard Symbol 198
- 606 XML-Refaktorierung nun an weiteren Stellen möglich 199
- 607 - Es kann nun über das Flag -on bzw. --onlyNew gesteuert 199
- 609 Es kann im Mapping nun auf jede Ausprägung des Hauptartikels 199
- 610 Wird eine Artikelnummer beim übertragen angegeben und es 199
- 611 - Es können nun Gutschriften importiert werden wenn ein Status 199
- 623 - ID-Lookup-Liste für Artikel und Varianten, damit diese nicht 199
- 625 Artikelnummern mit Sonderzeichen werden nun unterstützt 200
- 636 Neuer Sync: Nachträglich in der Winline geänderte Versandart 200
- PA-34 Die Generierung von Code für Plugins ist nun dynamisch und 200
- SH-17 - Das übertragen der Tracking-Nummern ist eine separate 200
  - Bekannte Probleme 200
  - Update-/Installationshinweise 201
- 444 Die Datei jobs.sqlite aus dem Installationsverzeichnis muss vor 201
- 613 Kopieren von Werten nach der gleichen Bedingung nun möglich 202
- 634 Bestell- und Zahlungsstatus werden aus der Winline heraus 202
  - Fehlerbehebungen 202
- 463 Leerzeichen aus default Installationspfad entfernt. Der AppData 202
- 549 Ausgabe des Traceback wenn im Artikel eine Exception aufkommt. 202
- 550 Im Mapping kann nun auf das ShopAPI Objekt zugegriffen werden 202
- 553 Optionen bei denen alle Varianten inaktiv sind werden nicht 203
- 556 Problem behoben wenn ein Hauptartikel mit Ausprägungen 203
- 567 - Update auf neue WLAPI version benötigte umstellungen im 203

- 570 Durch das Entfernen der Kategorie-Eigenschaft und der 203
- 571 Problem behoben wenn die Kategorieeigenschaft-Optiomn eine 203
- 606 Leerzeichen-Bug behoben 204
- 612 - Fehler behoben bei dem am ende der Preisübermittlung nur die 204
- 616 - Wenn in der Config keine Preisliste steht wird eine Leere 204
- 618 - Update auf Patchversion 11.9.1 von WLAPI um Bug mit der HSL 204
- 619 XML-Refactor: Leere Tags werden auf None gesetzt 204
- 620 - Update auf WLAPI v11.9.2 für Performanz-Optimierung 204
- 621 - Die Preise der Varianten werden nun nicht mehr übertragen 204
- 627 Fix für den Kundenexport falls ein None-Wert gelöscht werden 205
- 628 Timestamp für den syncFilteredCustomers wird nun von dem 205
- 629 - Fehlerfall in dem Beschreibung gar nicht gesetzt ist 205
- PA-35 - Fixes Issue with parameters 205
- PA-4 - syncCustomersFiltered erweitert damit die Attribute 205
- SH-10 Installer-Template update auf neuste version und alte config 205
- SH-11 Bestellungen werden erst ab konfiguriertem Datum 205
- SH-14 Wenn die Textfelder, über die die Bestellungen für den 206
- SH-15 Flask version muss erst mal fixiert sein. 206
- SH-25 - Behebt ein Problem Belege in der WinLine zu finden, wenn 206
- SH-28 - Der Parameter -a/--all wird nicht korrekt an die Aufgabe 206
- SH-29 - Filterparameter wird wieder korrekt mit StatusId gefüllt. 206
- SH-31 Durch die neue Einstellung "ignore_missing_wl_konto" kann 207
- SH-32 Nachdem auf der XML Datei gearbeitet wurde, werden alle None 207
- SH-36: 207
- SH-37 - Wenn die XML Datei zu einem Kunden nicht 207
- SH-38 Behebt Fehler bei dem die locale Variable die ober geordnete 207
- SH-46 - Behebt Duplikate im erlaubten Protokoll 208
- SH-54: Mappinghelper nutzt nun einen YAML loader dadurch kann 208

- SH-8 Behebt crash wenn firstname oder lastname None sind. 208
  - Sonstiges 208
- 441 - Aktualisierung der Templates 208
- 489 - Dokumentation Kundenimport 208
- 501 - Update von SQLAlchemy auf Version 1.4.18 Update von WLAPI auf 208
- 545 Die Änderungshistorie wird für die Dokumentation automatisch 208
- 555 Log-Level für das Fehlschlagen von AsyncIO requests war 209
- 563 - Dokumentation für Ausprägungen 209
- 565 - Dokumentation Sendungsverfolgung und Statusupdate 209
- 579 - Dokumentation für die Umsetzungstabellen des Mappings 209
- 580 - Dokumentation für den Kundenimport (Plugin, xml-refactoring, 209
- 583 - Generelle Aktualisierung der Dokumentation 209
- 592 - Dokumentation Aktionserver-Restart script 210
- 594 - Über die Config kann nun gesteuert werden ob 210
- 603 Sequenz-Diagramm für die Statusupdates 210
- 608 Update WLAPI zu Version 11.9.0. Es kann nun über die Property 210
- 635 - Gitlab-CI hat ein altes Image gezogen. Neues image sorgt 210
- PA-35 - Neue Funktion um RTF zu Plaintext zu Konvertieren. 211
- PA-40 - Extra Route für Kunden- und Bestellungsabruf in einer 211
- SH-18 Update auf WLAPI version 11.10 für dynamische 211
- SH-19 - Bestellungen können nun in SW5 in mehrere Stati nach 211
- SH-20 - Einfacher Verlauf der Statusupdates welche 1 Woche zurück 211
- SH-42 - Neue Route die erst die Kunden und anschließend die 211
- SH-43 - Der Text(Sprache) der aus den Eigenschaften übertragen 212
- SH-48 Plugin um Rabattleisten als Kundengruppen zu übertragen. 212
- SH-9 Anlage Dokumentation Leitfaden 212
  - v2.1.0 212
    - Neue Funktionen 212

- 321 - Für den Abruf der Artikel aus der WinLine wird nun 212
- 456 - Plugin-System implementiert. Im Installationsverzeichnis 212
- 457 - Über das Mapping kann nun ein Feld aus der WinLine als Liste 213
- 458 - Es können nun Ausprägungsgruppen explizit ausgeschlossen 213
- 475 - In der Config können nun mehrere Zusatzfelder hinterlegt 213
- 488 Neuer Config-Parameter xmlexport_customers. In das hier 213
- 491 - Trackingnummern werden nun zurückgeschrieben sobald diese in 213
- 497 - Es können nun Archivbilder explizit den Varianten in Shopware 214
- 498 - Übermittelt Trackingnummern aus Shopware in Lieferscheine in 214
  - Bekannte Probleme 214
- 494 - Images can not directly transmitted to WooCommerce. 214
- 508 - Nicht alle Bilder werden gesetzt. 214
- 542 - Durch den Umbau funktioniert die Bearbeitung des Mappings aus 214
  - Update-/Installationshinweise 215
  - Fehlerbehebungen 215
- 487 - Fixxes Progressbar bug welches den text anhand von 215
- 496 - Bugfix für den die URL parameter immer wieder an den Aufruf 215
- 499 - Keine doppelten Optionen an den Shop übermitteln, da dieser 215
- 503 - Kommas in Artikelnummern werden nun ersetzt 216
- 503 - Nummer wurde nach jedem ersetzen wieder zurückgesetzt. 216
- 504 - Beim dem Abrufen der Artikel aus dem Shop werden die 216
- 506 - Da die Option aus dem Mapping entfernt wurde, wurde bei dem 216
- 507 - Abhängigkeit der Webpreisliste in den Einstellung behoben. 216
- 508 - Durch das umbenennen einer Variable kam es zu Konflikten wenn 216
- 513 - Es werden nun nur noch Ausprägungen in Shopware angelegt die 217
- 514 - Anpassung an WLAPI release für Relation die umbenannt und 217
- 516 - Unnötige Logger ausgaben bei der Artikelnummer konvertierung 217
- 519 - Varianten-Liste konnte None enthalten, was beim übertragen in 217

- 520 - Die ID Liste enthält bereits die in die Ursprungs form 217
- 521 - Pfad muss erst gejoint werden. 217
- 522 - Artikelnummer wurde an einigen stellen von Sonderzeichen 217
- 526 - Wenn die Variant-Daten die an den Shop übermittelt werden 218
- 529 - Hier wurde die Methode raise_for_status nicht aufgerufen und 218
- 530 - Artikel_Data mapping nachdem die Eigenschaften abgefragt 218
- 543 - Die neue Mapping Struktur wird nun in den tests verwendet. 218
- 544 - Fehlender Config-Eintrag für Tests nachgetragen. 218
  - Sonstiges 218
- 473 - Logging für Pluginaufrufe ergänzt. 219
- 476 Update Template-Projekt auf Version 0.4.4. 219
- 527 - print für folder 219
  - v2.0.0 219
    - Neue Funktionen 219
- 368 Preisübertragung wird nun mit AsyncIO durchgeführt um den 220
- 368 LRU_Cache ohne Limit für die Funktion 220
- 368 Die Anzahl der maximalen parallelen TCP-Verbindungen kann über 220
- 392 Optionale Fehleranalyse mit Sentry. Wird ein Logging-Server in 220
- 452 Sentry-Config in App-Data und User-Data berücksichtigen. 221
  - Bekannte Probleme 221
- 368 Logging überarbeitet um Datenmüll zu veringern wenn Debug-Level 221
  - Update-/Installationshinweise 221
- 437 Die Beispiel-Konfigurationsdatei wurde überarbeitet und 222
  - Nicht mehr unterstützte Funktionen 222
- 368 woo_wl.update_article_prices() sollte nicht mehr verwendet 222
- 437 Einige Config-Felder sind aus der Konfigurationsdatei entfernt 222
  - Kritische Fehler 222

- 413 Wird der Produkt Sync ohne Parameter gestartet so wird kein 222
  - Fehlerbehebungen 223
- 191 Libreoffice beenden wenn die Konvertierung einen Fehler läuft. 223
- 438 Behebt einige Bugs mit Varianten und Hauptartikel mit Varianten 223
- 446 Problem mit der Default Config behoben im Bezug auf den 223
- 462 - Filtert nun die tatsächlichen Chargen aus der Liste der 223
  - Sonstiges 224
- 237 Fehlerbehandlung wenn die Requests fehlschlagen. Prinzipiell 224
- 415 In der CI keine neuen Environments pro MR erstellen, da die 224
- 435 Dokumentation für Actionserver, Webservicevorlagen und minimal 224
- 464 Integration von automatisiertem Release-Prozess inklusive 224

Überblick

`Aktuelle Version:` v3.2.0 (Änderungshistorie)

`Homepage:` www.newenergy-bremen.de *(http://www.newenergy-bremen.de)*

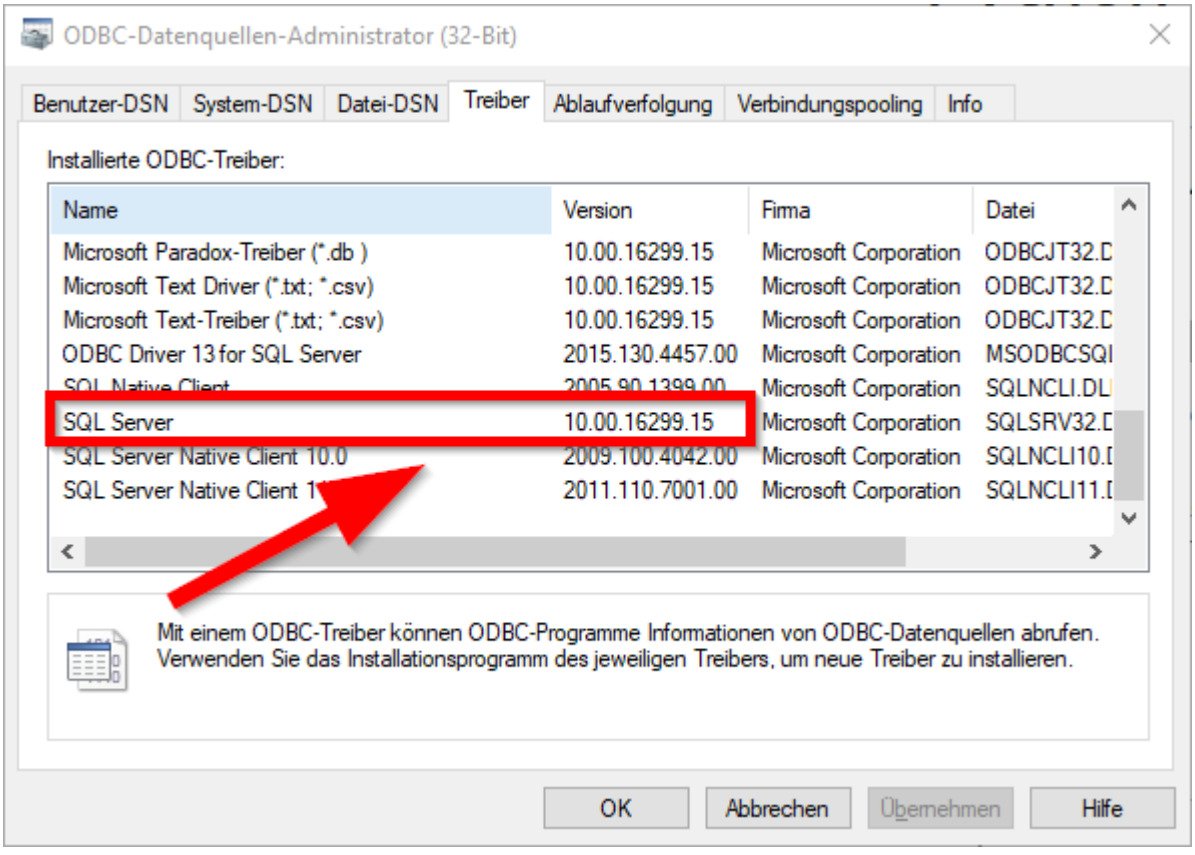
`Kontakt:` info@newenergy-bremen.de *(mailto:info@newenergy-bremen.de)*

`Entwickler:` new energy GmbH

`Kompatibilität:` Shopware 6

Systemanforderungen

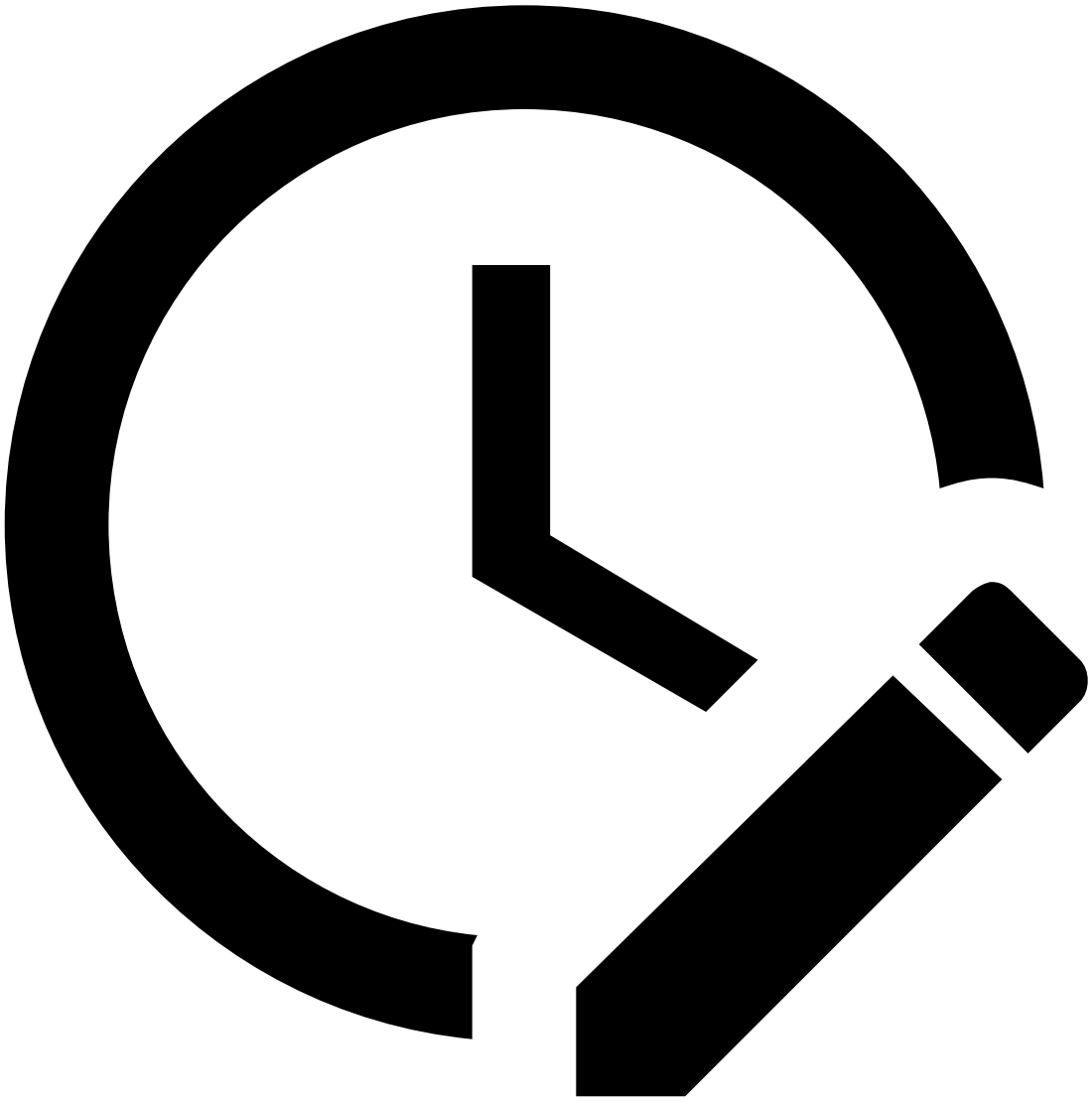
Anforderung	Details
Betriebssystem	Windows Server 2016 oder neuer / Windows 10/11
Shopware	Shopware 6 (aktuelle Version empfohlen)
ODBC-Treiber	ODBC Driver 18 for SQL Server oder neuer
WinLine Module	CWLFAKT, Batchbelege oder WebServices, Archiv2
Netzwerk	Zugriff auf Shopware-API (HTTPS) und WinLine SQL-Server



Die vollständige Funktionalität kann nur sichergestellt werden wenn die Systemanforderungen erfüllt werden. Für optionale Funktionen werden ggf. weitere WinLine-Module benötigt — siehe Feature-Matrix.

Funktionsweise

Die Shop-Anbindung für die WinLine wird als Service auf dem WinLine-Server eingerichtet. In einem festen Intervall prüft dieser, ob Stammdaten der Artikel geändert wurden und aktualisiert diese im Web-Shop. Danach werden alle Bestellungen und Kunden im Shop abgerufen und entweder über den WinLine Webservice oder per EXIM/XML-Import in die WinLine übertragen.



Erste Schritte

Diese Seite gibt einen Überblick über die empfohlene Reihenfolge der Einrichtung und hilft bei der Entscheidung, welche Funktionen für Ihre Installation relevant sind.

Empfohlene Einrichtungsreihenfolge

Die Einrichtung erfolgt in drei Phasen:

Phase 1: Grundlagen (Pflicht)

Diese Schritte sind für jede Installation erforderlich:

1. **Installation** — Software installieren, Konfigurationsdateien anlegen
2. **Shop-Verbindung** — API-Zugangsdaten und SQL-Verbindung konfigurieren
3. **Webartikel** — Artikelstammdaten synchronisieren
4. **Artikelgrafik** — Bilder und Downloads übertragen
5. **Preise** — Preise synchronisieren (Standardpreise, ggf. Kundengruppen)
6. **Kategorien** — Artikel in Shop-Kategorien einordnen
7. **Statusupdate & Tracking** — Bestellstatus und Sendungsverfolgung einrichten
8. **Aufgabenplanung** — Automatische Synchronisierung konfigurieren

Phase 2: Kundenverwaltung (empfohlen)

1. **Kundensynchronisation** — Kundendaten zwischen Shop und WinLine abgleichen
2. **Kundenimport** — Kunden aus dem Shop in die WinLine importieren

Phase 3: Optionale Erweiterungen

Diese Funktionen können je nach Bedarf aktiviert werden — in beliebiger Reihenfolge:

- **Varianten/Ausprägungen** — Artikel mit Größen, Farben etc.
- **Eigenschaften** — Filterbares Merkmal im Shop (benötigt WinLine-Modul)
- **Crossselling** — Produktempfehlungen (benötigt WinLine-Modul)
- **Mehrsprachigkeit** — Artikeldaten in mehreren Sprachen
- **Mindestbestellmenge** / **Losgröße** — Bestellmengen steuern
- **Disposition** — Lieferzeiten und Lagerbestand (benötigt WinLine-Modul)
- **Handelsstücklisten** — Produktbündel (benötigt WinLine-Modul)
- **Konfigurator** — Individuelle Produktkonfiguration (nativ in Shopware 6)
- **Gutscheine** — Gutscheinverwaltung im Shop

Feature-Matrix

Übersicht aller Funktionen mit ihren Abhängigkeiten:

Kernfunktionen (immer enthalten)

Funktion	Beschreibung	Richtung
Webartikel	Artikelstammdaten synchronisieren	WinLine → Shop
Artikelgrafik	Bilder und Downloads aus Archiv II	WinLine → Shop
Preise	Standard-, Gruppen-, Staffel- und Individualpreise	WinLine → Shop
Kategorien	Artikel-Kategorisierung (Variante Untergruppen)	WinLine → Shop
Varianten	Ausprägungen (bis zu 2 Gruppen)	WinLine → Shop
Statusupdate	Bestellstatus und Sendungsverfolgung	WinLine ↔ Shop
Kundensync	Kundendaten abgleichen	Shop → WinLine
Kundenimport	Kunden in WinLine importieren	Shop → WinLine
Mapping	Feldzuordnung zwischen Systemen	—
Umsetzungstabellen	Werte-Transformation	—
Mindestbestellmenge	Min-/Max-Bestellmengen	WinLine → Shop
Losgröße	Bestellschrittweite	WinLine → Shop
Mehrsprachigkeit	Artikeldaten in mehreren Sprachen	WinLine → Shop
Gutscheine	Gutscheinverwaltung	Shop (manuell)
Konfigurator	Individuelle Produktkonfiguration (nativ in SW6)	Shop (manuell)
Management-Interface	Web-Oberfläche für Sync und Planung	—

Funktionen mit zusätzlichen Modulen

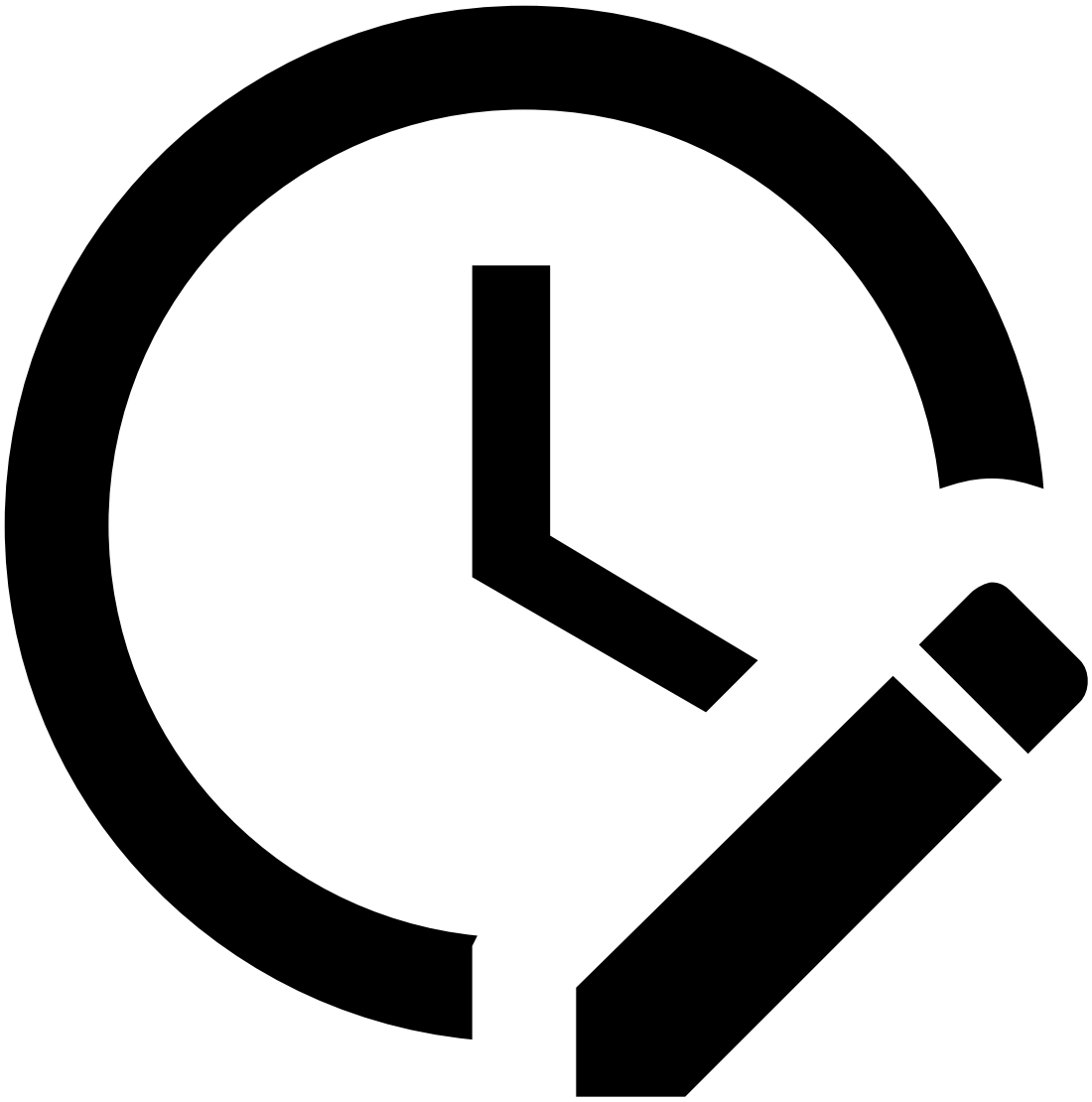
Funktion	Benötigtes Modul	Beschreibung
Kategorien (Multi-Variante)	Multi-Kategorien	Mehrere Kategorien pro Artikel, hierarchisch
Eigenschaften	Artikeleigenschaften	Filterbare Merkmale im Shop anzeigen
Crossselling	Crossselling	Produktempfehlungen (Zubehör, Alternativen)
Disposition	Lieferzeiten + Disposition	Verfügbarkeit und Lieferzeiten berechnen
Handelsstücklisten	Handelsstücklisten	Produktbündel synchronisieren

Hilfreiche Werkzeuge

Werkzeug	Beschreibung	Dokumentation
`ess.exe healthCheck`	Verbindungen und Konfiguration prüfen	Installation
`ess.exe syncProducts -a`	Alle Artikel synchronisieren (Initialsync)	Webartikel
Management-Interface	Web-Oberfläche für Sync-Steuerung	Management-Interface
Logdateien	Fehlersuche und Diagnose	Logging

Tipp

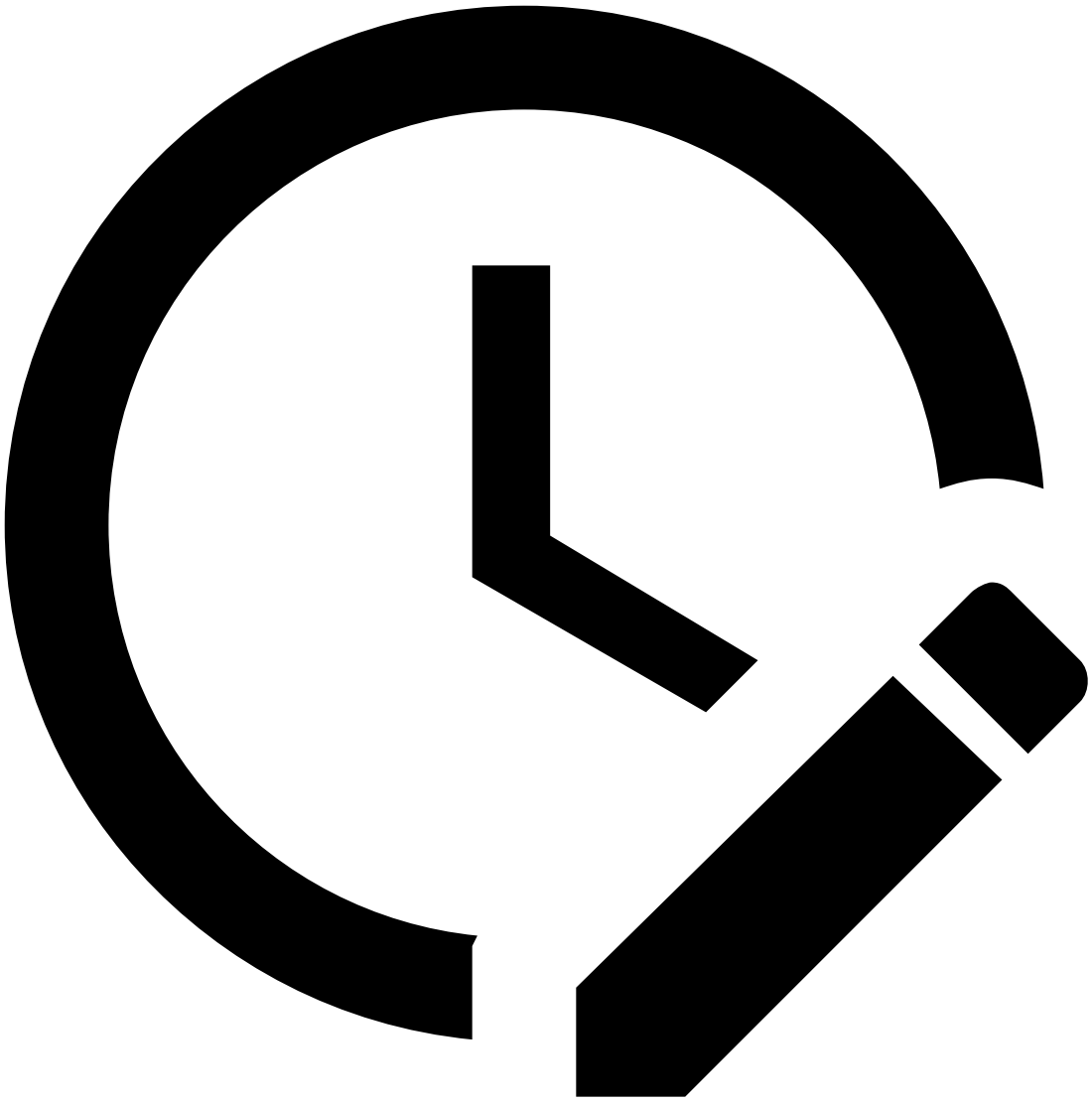
Verwenden Sie die Checkliste als Leitfaden während der Ersteinrichtung. Bei Problemen hilft die FAQ mit Lösungen für häufige Fragen.



2026-03-25

Checkliste für Installation und Einrichtung

- [ ] Daten für die Initialkonfiguration abfragen (Siehe Konfiguration und Erste Schritte)
- [ ] Installation
- [ ] Installationsdateien vom Bereitstellungsverzeichnis herunterladen
- [ ] ZIP Datei auf Zielsystem entpacken
- [ ] Konfigurationsdateien anlegen
    - Variante 1 (manuell)
    - [ ] `conf` Ordner anlegen
    - [ ] Inhalt aus `Beispiel-Konfiguration\SW6` Ordner in `conf` Ordner kopieren, ggf. `shopware6.ini` in `config.ini` umbenennen
- [ ] Shopware 6 Einrichtung
- [ ] Integration im Shopware-Admin anlegen (Einstellungen > System > Integrationen)
- [ ] Access Key ID und Secret Access Key notieren
- [ ] Steuersätze im Shop prüfen (müssen mit den WinLine-Steuersätzen übereinstimmen)
- [ ] Ggf. Kundennummern-Einstellung prüfen (Shopware generiert Kundennummern: Ja/Nein)
- [ ] Konfiguration
- [ ] `config.ini` aus dem `conf` Ordner anpassen
    - [ ] Werte aus Konfiguration übernehmen
    - [ ] Shop-URL und API-Zugangsdaten eintragen (`[SHOP]`)
    - [ ] WinLine Mandant konfigurieren (`[WL]`)
    - [ ] SQL-Verbindung konfigurieren (`[SQL]`)
    - [ ] WebService-Zugangsdaten eintragen (`[EWL]`)
- [ ] Mapping-Dateien anpassen (`conf/mapping/`)
- [ ] WinLine Einrichtung
- [ ] Beschlagwortung anlegen (Webshop Grafik, Webshop Download)
- [ ] WebService-Vorlage importieren und anpassen
- [ ] Webartikel-Eigenschaftsfeld anlegen/prüfen
- [ ] Initialisierung
- [ ] `init.bat` per Doppelklick ausführen (fordert Adminrechte an, ruft `ess.exe init` auf) oder alternativ `ess.exe init` in einer Admin-Shell starten – legt nötige Dateien und Beispielkonfigurationen ab
- [ ] Test
- [ ] `ess.exe healthCheck` ausführen um Verbindungen und Konfiguration zu prüfen
- [ ] Testbestellung im Shop anlegen
- [ ] `ess.exe syncProducts -a` ausführen und prüfen ob Artikel im Shop erscheinen
- [ ] `ess.exe syncOrders` ausführen und prüfen ob Bestellung in WinLine importiert wird
- [ ] Logdateien auf Fehler prüfen
- [ ] Aufgabenplanung einrichten
- [ ] Aufgabe für Artikelsync (1x täglich empfohlen)
- [ ] Aufgabe für Bestellsync (alle 15 min empfohlen)
- [ ] Ggf. weitere Aufgaben für Preis-/Bestands-/Kundensync



2026-04-16

Installation

Installation ausführen oder Programm in einen beliebigen Ordner kopieren. Das Programm ist nach der Einrichtung einsatzbereit.

WinLine

Einführung zur Bestellungsynchronisierung

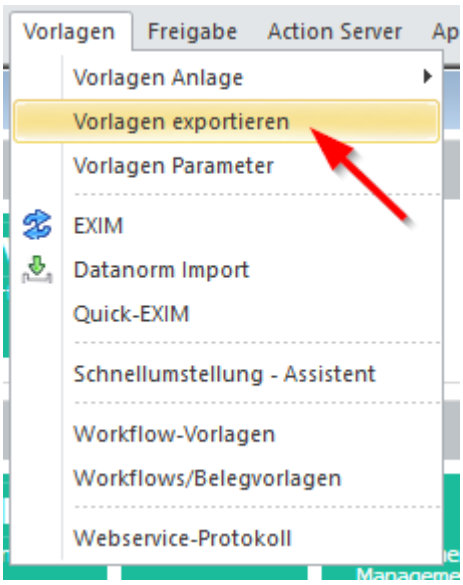
Für die Bestellungsynchronisierung wird in der WinLine entweder der Webservice oder der Action Server benötigt. Die nachfolgenden Anweisungen sind für die Einrichtung mit Webservice.

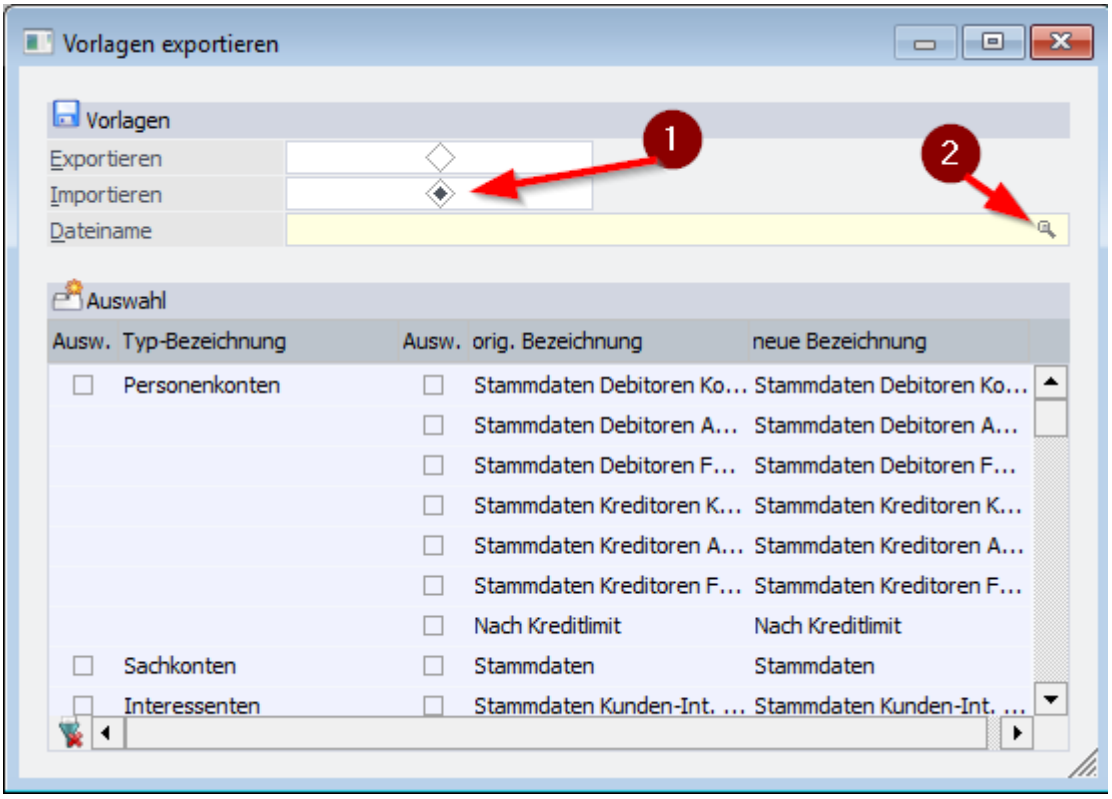
Abweichende Einstellungen mit dem Action Server finden sie hier: Einrichtung mit dem Action Server

Grundsätzliches

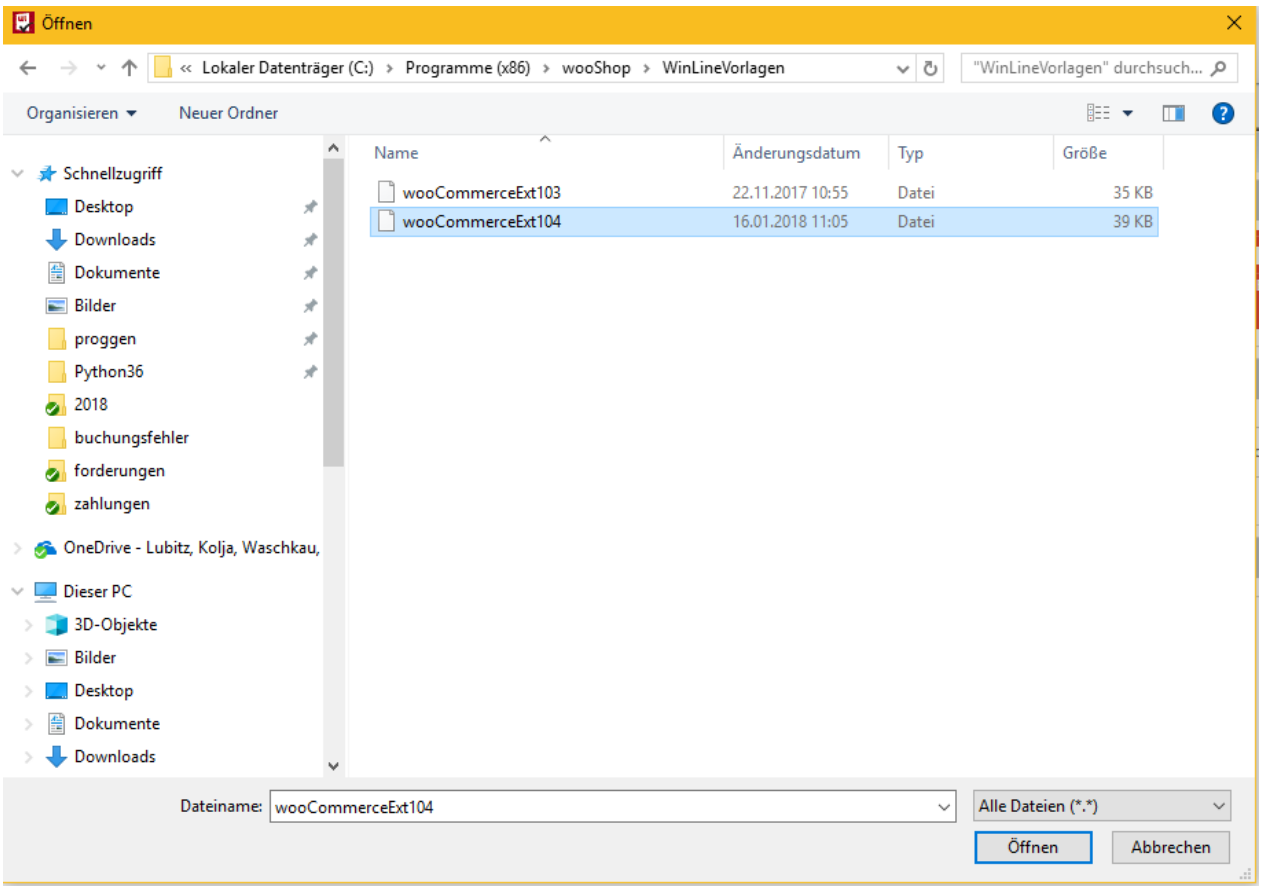
1. Klären sie welche Daten aus dem Web-Shop für den Belegimport verwendet werden sollen.
2. Passen sie anhand dieser Vorgaben die Konfigurationsdatei und die Mapping-Datei an. (Siehe Kapitel Config)
3. Importieren sie die WebService-Vorlage (Siehe Kapitel Web-Vorlage)
4. Passen sie die Webvorlage an die Konfigurationsdatei und das Mapping an. (Siehe Kapitel Anpassen der WebService-Vorlage)
5. Richten sie eine Aufgabe in der Windows-Aufgabenplanung oder dem Management Interface ein.
6. Überprüfen sie mit einer Testbestellung ob die Synchronisierung funktioniert
7. Prüfen sie den Beleg in der WinLine

Web-Vorlage



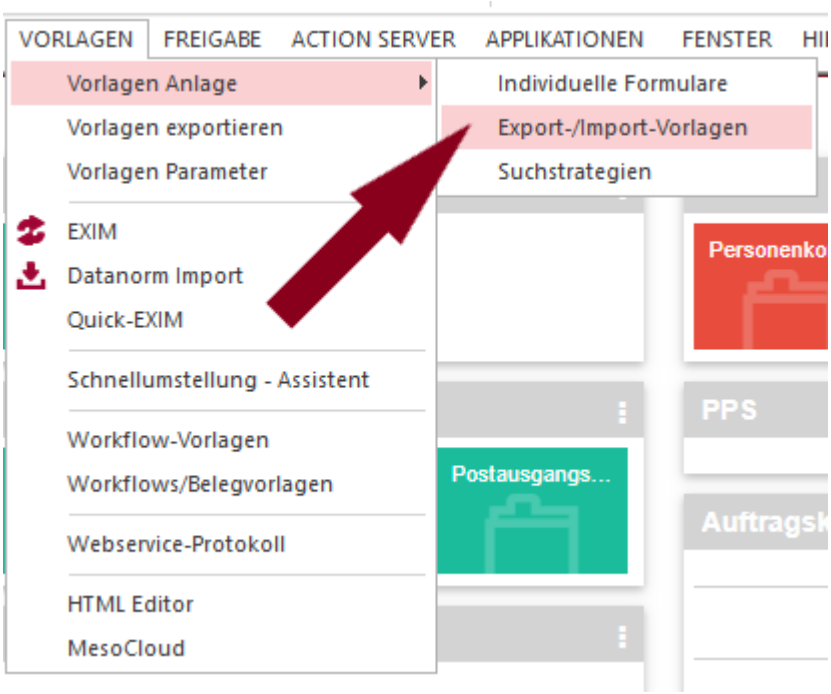


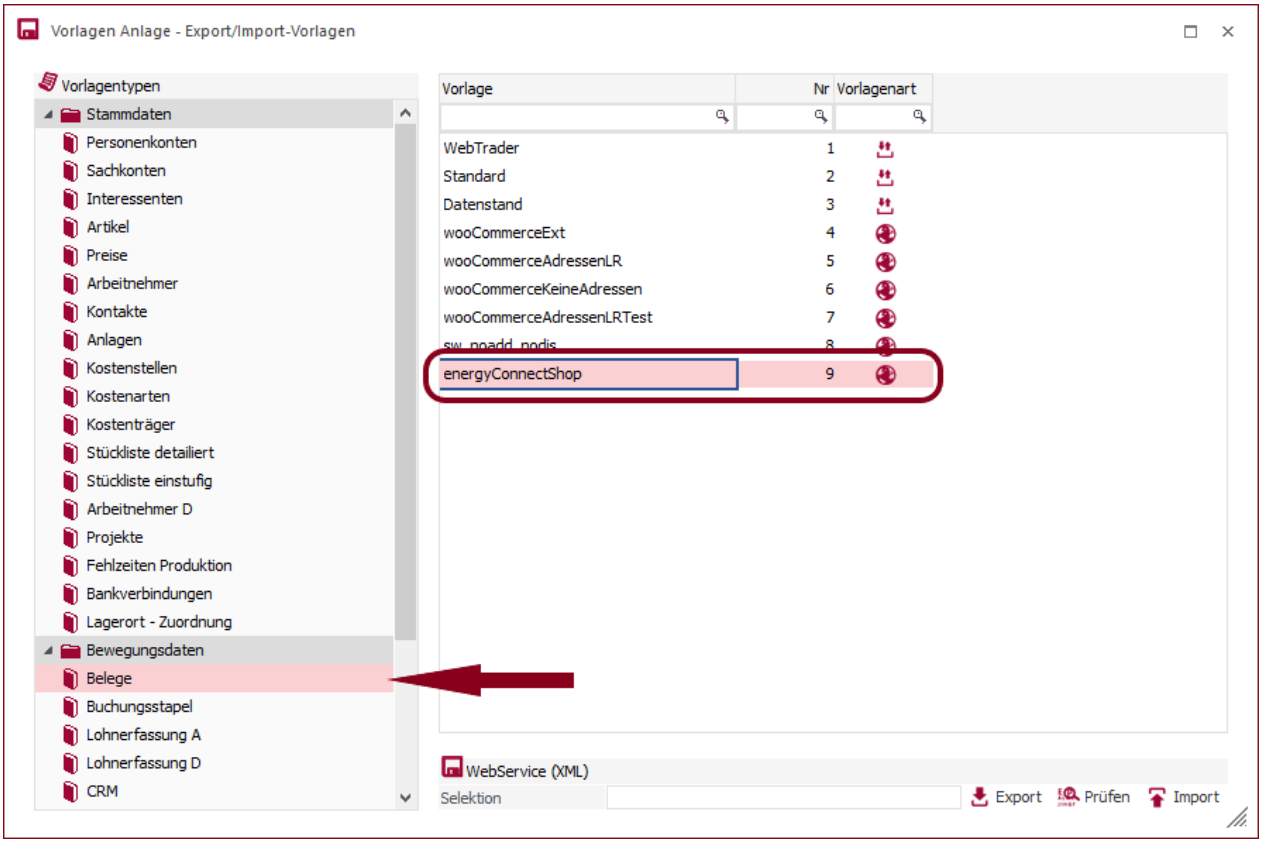
WinLine Vorlage für installierte WinLine-Version wählen. Die Vorlagen befinden sich im Programmpfad im Ordner `WinLineVorlagen`



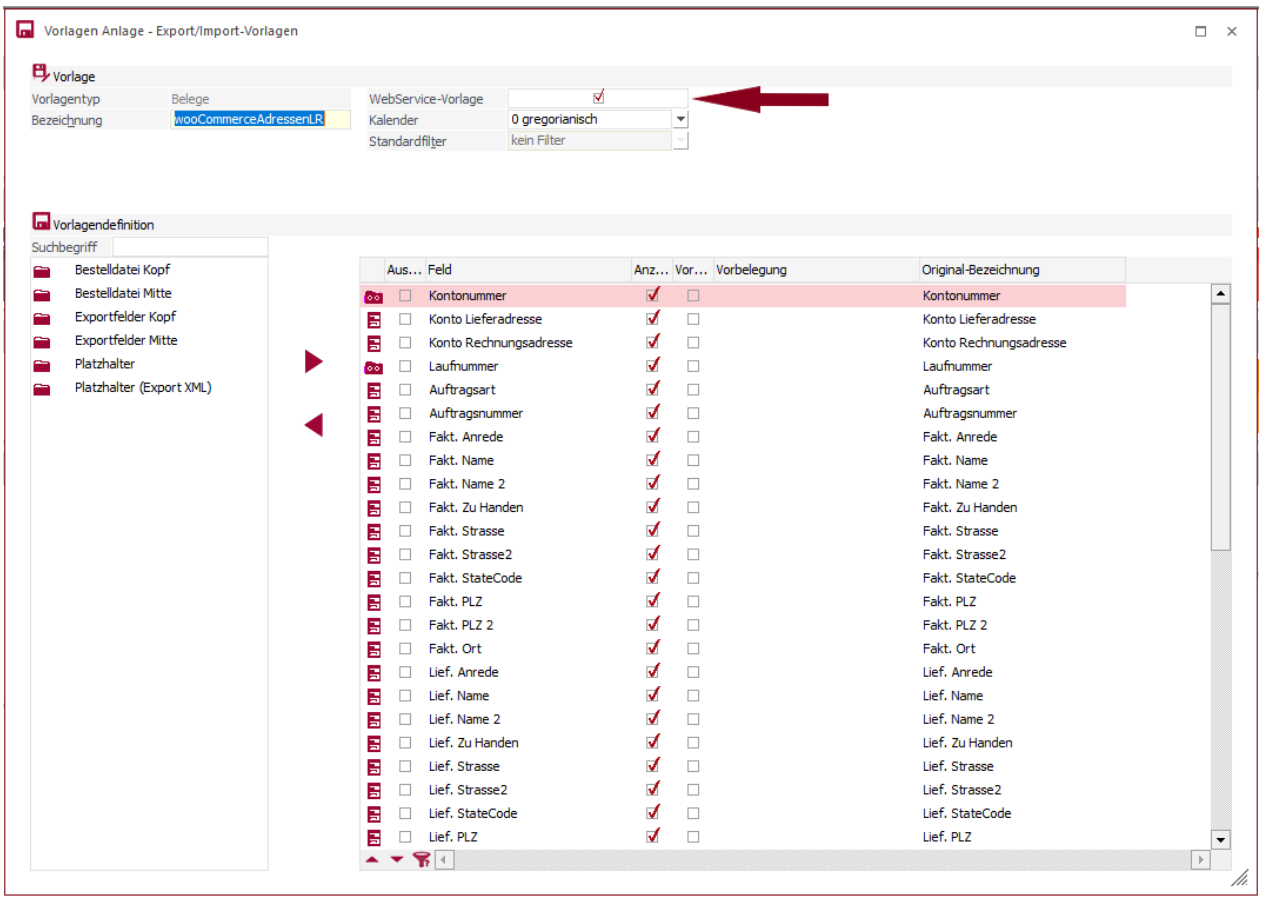
In der Vorlage `energyConnectShop_WL105` ist eine Minimalvorlage enthalten, die entsprechend der verwendeten Einstellungen und Mapping-Datei angepasst werden muss (Siehe Kapitel Config Welche Einstellungen welche Auswirkungen auf die Vorlage haben). Diese Vorlage muss als WebService-Vorlage markiert werden.

Setzen der Vorlage als WebService-Vorlage





Folgender Haken muss bei beiden Vorlagen manuell gesetzt werden.



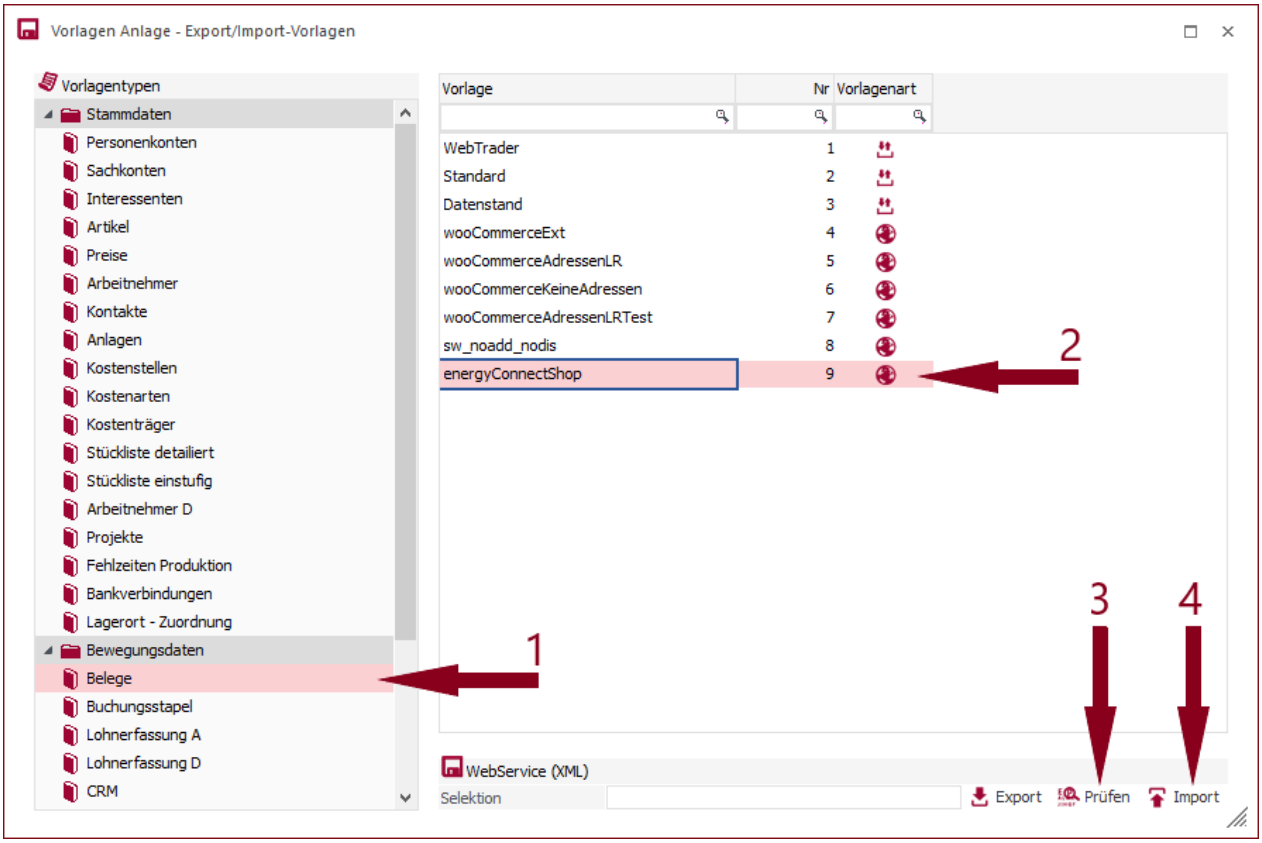
Hinweis: Wenn sie die Bezeichnungen der Vorlagen ändern, müssen sie diese neuen Bezeichnungen in der Config-Datei hinterlegen. (Siehe Config Anpassungen)

Anpassen der WebService-Vorlage

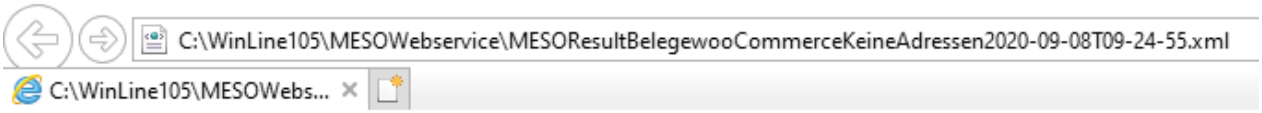
NOTE: Die Anpassung der WebService-Vorlage sollte erst gemacht werden nachdem die Konfigurations-Datei und das Mapping an die Anforderungen des Kunden angepasst wurden.

Der einfachste Weg die Vorlage an die Einstellungen und das Mapping anzupassen ist es, eine Testbestellung aus dem Shop abzurufen und als XML-Datei zu speichern. 1. Legen sie zunächst eine Testbestellung im Shop an. 1.1 Im Standard werden alle Bestellungen vom Shop abgerufen die im Status ‘offen’ stehen. Wurde ein anderer Status für das Abrufen in der Konfigurations-Datei hinterlegt muss der Status der Bestellung manuell geändert werden, damit die Bestellsynchronisierung die Testbestellung abruft. 2. Hinterlegen sie in der Config in dem Feld ‘xmlpath’ im Segment ‘WL’, einen Pfad zu einem Ordner in welchem die XML-Dateien gespeichert werden sollen. 3. Führen sie den Bestellsync einmal aus. 3.1 In dem Ordner liegt jetzt die Bestellung als XML-Datei. 3.2 Nutzen sie einen XML-Formatter um die Datei lesbarer zu machen. 4. Gehen sie alle Elemente, ausgenommen **BELEGKEY**, unter dem Element **`energyConnectShopT025`** durch und fügen sie die fehlenden Felder in der WebService-Vorlage hinzu. 4.1 Achten sie darauf die exakt gleiche Reihenfolge der XML-Datei beizubehalten. 4.2 Sie müssen keine Felder aus der Vorlage entfernen. Die Felder **`Textzeilen1-10`** müssen nicht entfernt werden auch wenn sie in der XML-Datei nur teilweise auftauchen. 5. Prüfen sie ob die Bestellung mit der angepassten Vorlage eingelesen werden kann. 5.1 Wählen sie die Vorlage

aus (siehe Pfeil 1. - 2. Abb. Unten) und klicken sie anschließend auf ‘Prüfen’ (siehe Pfeil 3. Abb. unten). Wählen sie im folgen File-Explorer die XML-Datei aus.

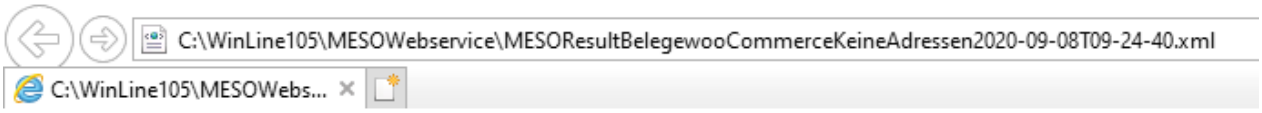


5.2 Die Winline führt dann eine Schema-Prüfung des XML durch und gibt das Ergebnis zurück. Dafür öffnet sich dann der Browser um das Ergebnis anzuzeigen. Wenn das XML nicht zu der Vorlage passt, wird in dem Ergebnis das erste abweichende Element angezeigt. Passen sie die Vorlage erneut manuell an, bis sie zu dem XML passt. Nehmen sie die nötigen Änderungen im Mapping vor bis das XML zur Vorlage passt. 5.3 Wenn das XML zur Vorlage passt, sieht die Ausgabe folgendermaßen aus:




```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <MESOWebServiceResult>
    <OverallSuccess>true</OverallSuccess>
</MESOWebServiceResult>
```

5.4 Sie können nun manuell die Bestellung importieren, indem sie im Dialog anstatt auf “Prüfen” auf “Import” klicken (Pfeil 4. in der Abbildung). Wenn der Beleg erfolgreich eingelesen werden konnte, erhalten ein folgendes Ergebnis:




```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <MESOWebServiceResult>
    <OverallSuccess>true</OverallSuccess>
  - <ResultDetails>
      <KeyValue>BELEGKEY: 1</KeyValue>
      <VoucherNumber>545</VoucherNumber>
      <Success>true</Success>
    </ResultDetails>
  - <ResultDetails>
      <KeyValue>BELEGKEY: 1 - Beleg wurde erfolgreich gedruckt</KeyValue>
      <VoucherNumber>LS-1133</VoucherNumber>
      <Success>true</Success>
    </ResultDetails>
</MESOWebServiceResult>
```

6. Entfernen sie den Ordner-Pfad aus der Konfigurations-Datei, damit die Bestellungen per WebService direkt eingelesen werden.

Config

Sammelkonto

In der Konfigurations-Datei kann in dem Feld `sammelkonto` eine Personenkontonummer hinterlegt werden, auf die alle Bestellungen/Belege gebucht werden, die keine Personenkontonummer in Shopware haben. Dies sollte die Personenkontonummer eines Sammelkontos in der WinLine sein und kann bspw. für B2C-Bestellungen verwendet werden, für die keine Personenkonten angelegt werden sollen. Darüber hinaus kann mit der Option ‘always_sammelkonto = True’ bestimmt werden, ob alle Bestellungen grundsätzlich auf das angegegebene Sammelkonto gebucht werden sollen. Damit würden die in SW vergebenen Kundennummern beim Bestellimport ignoriert werden.

```
; Kundennummer für Orders ohne oder mit in der WinLine nicht hinterlegten Kundenn
sammelkonto = 50000
always_sammelkonto = False
```

Vorlagen

Adressen

Es gibt 4 verschiedene Belegabrufarten bezüglich der Adressen.

	WinLine zieht Adressen	nur Lieferadresse aus dem Shop	nur Rechnungsadresse aus dem Shop	Beide Adressen aus dem Shop
addressFromShop	False	True	False	True
billFromShop	False	False	True	True

Belegstufe

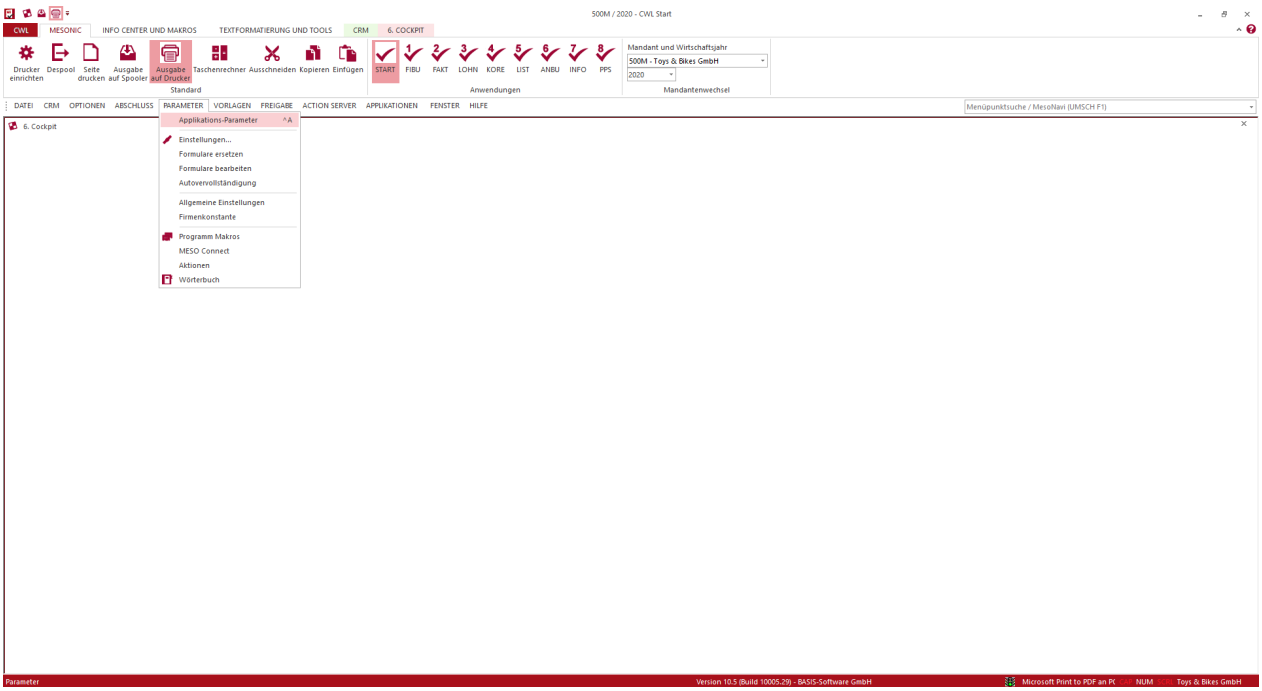
Im Segment `ORDERS` im Feld `import_sequences` wird die Reihenfolge der Belegabrufe aus dem Webshop definiert. Der Wert im Beispiel `Bestellung` ist dann der Segmentname in der dieser Abruf konfiguriert wird.

Im Feld `import_belegstufe` (Segment hängt von der `import_sequences` ab) kann definiert werden in welcher WinLine Belegstufe, die vom Webshop abgerufenen Bestellungen, erfasst werden. Mögliche Belegstufen: `nicht_drucken`, `Angebot`, `Auftrag`, `Lieferschein` und `Rechnung`.

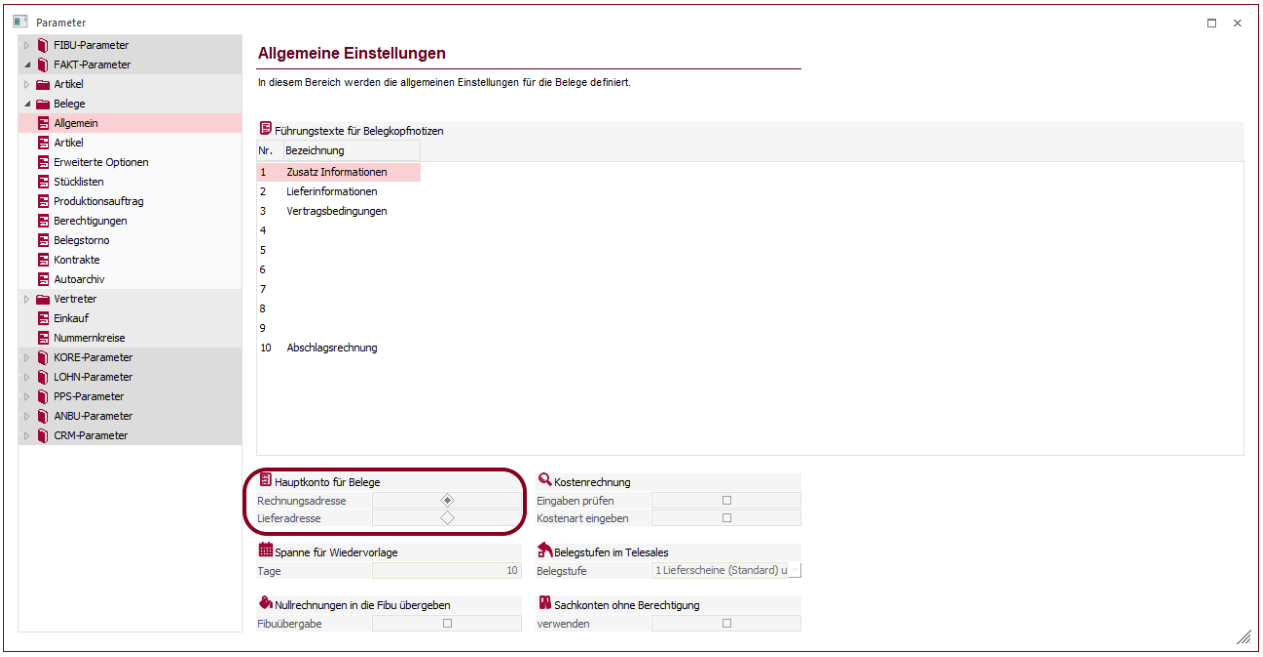
```
[ORDERS]
import_sequences = ["Bestellung"]

[Bestellung]
status_fetch = OrderState.open
status_success = OrderState.inProgress
status_fail = OrderState.open
; status_tracking = OrderDeliveryState.shipped
import_belegstufe = Lieferschein
check_existing = True
```

Hauptkonto für Belege

Die in der WinLine erfassten Belege können entweder anhand der Liefer- oder anhand der Rechnungsadressen geführt werden. Dies ist eine grundlegende Einstellung der WinLine. Prüfen sie welche Art das Hauptkonto für Belege hat. 1. Öffnen sie die Applikations-Parameter



2. Prüfen sie den Wert unter `FAKT-Parameter -> Belege -> Allgemein`



Steht dieser Parameter auf `Rechnungsadresse` muss das Konfigurations-Feld `rechnungsadresskonto` auf dem Wert `True` stehen und das Konfigurations-Feld `lieferadresskonto` auf `False`. Tauschen sie die Werte wenn der Parameter in der WinLine auf `Lieferadresse` steht.

```
[WL]
;Nutze Rechnungsadresskonto aus dem Personenkonto für die Belege
rechnungsadresskonto = True
;Nutze Lieferadresskonto aus dem Personenkonto für die Belege
lieferadresskonto = False
```

Mapping - order.yml

Welche Felder in den Beleg übermittelt werden, wird in der `order.yml` definiert. Mit Hilfe von Umsetzungstabellen sind hier auch Wertübersetzungen leicht realisierbar.

OrderSync Einrichtung mit Action Server

Für den Action Server wird, wie für den Webservice auch, eine gesonderte WinLine-Lizenz benötigt. Damit der Actionserver automatisiert Bestellungen aus dem in der Config hinterlegten Verzeichnis (bspw.: xmlpath = C:\temp\xml) abholt, muss eine XML-Vorlage und ein Makro eingespielt werden.

Makro für Orders einlesen

WEBSHOP_BESTELLUNGEN.MMR

Das XML-Verzeichnis, welches in der Config hinterlegt wurde muss mit dem im Makro übereinstimmen. Die Verzeichnisse (import, error, success) müssen vorher angelegt werden.

Bei der Config-Option ‘batch-orders = True’ muss das nachfolgende angepasste Belegmakro verwendet werden. Dieses wird nicht automatisch (Actionserver) ausgeführt, sondern per Button aus dem Cockpit.

WEBSHOP_BESTELLUNGEN.MMR

Die Stapelverarbeitung macht eine manuelle Sichtprügung im Zuge des Belegimports möglich.

XML-Vorlagen

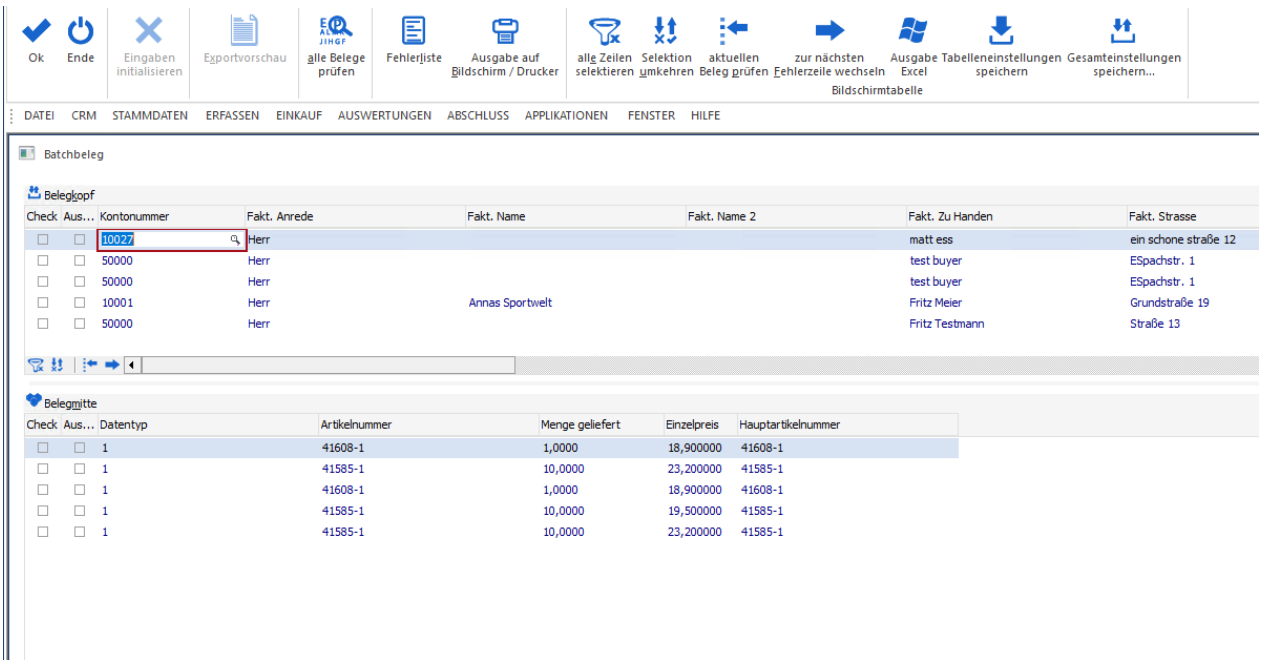
Da XML-Vorlagen nicht exportiert/importiert werden können, muss eine Vorlage anhand einer erstellten Bestellung erstellt werden. (Es kann auch dieses Muster verwendet werden Musterbestellung) Die Vorlage sollte gemäß Makro mit “webshop_bestellung” bezeichnet werden.

Bestellungen als Batch oder in Einzeldateien

Es ist möglich die Bestellungen als Batch zu generieren oder einzelne Bestellungen (Sammel-XML oder einzelne XML-Dateien).

```
; Batch Order: Bestellungen zu einer XML zusammenführen (True, False)
batch_orders = True
```

Mit einem Batch lassen sich Bestellabrufe gebündelt verarbeiten und leichter einer manuellen Sichtprüfung unterziehen. Gerade bei einer großen Anzahl von Shop-Bestellmengen ist ein solches Vorgehen interessant.



Action Server als Dienst starten/einrichten (siehe auch ggf. Whitepaper “Action Server als Dienst”)

Die erforderlichen Dateien finden sich bei einer “normalen” WinLine Installation im Unterverzeichnis EWL\BIN bzw. EWL\BIN64. Zusätzlich sollte am ausgeführten Computer das “C++ Runtime Package” installiert sein (vcredist_x86.exe). Dateien in Unterordner “ActionServer” im WinLine-Verbzeichnis kopieren.

mesoactionsvcmanager.config entsprechen der Installation anpassen:

```
servername=C:\Winline\CWLSTART.EXE
serverparams=/REPORTSERVER
```

Start über Powershell im entsprechenden Verzeichnis: `mesosvcmanager.exe -i2 --actionserver`

Ggf. prüfen, ob der Dienst ordnungsgemäß gestartet wurden und läuft.

Action Server Definition

Anschließend muss der Action Server eingerichtet werden. Unter “Action Server/Definition” das entsprechende Makro auswählen. Die Ausführungshäufigkeit und bestätigen.

NOTE: Hierbei ist zu beachten, dass für die Aktion ein Wirtschaftsjahr hinterlegt wird. Bei jedem Jahreswechsel muss dieses manuell auf das neue Wirtschaftsjahr gestellt werden.

Anschließend holt sich der Action Server die XML-Orders aus dem Verzeichnis und verschiebt diese in den Unterordner “webshop_bestellungen”. Die erfolgreich importierten Bestellungen landen im Belegmanagement unter “NICHT GERECHNET / GEDRUCKT”. Konnte eine Bestellung

nicht verarbeitet werden (bspws.: abweichende Artikelnummer) bleibt die Bestellung im Ursprungsverzeichnis. Dieses muss somit regelmäßig kontrolliert werden.

Action Server Dienst regelmäßig neustarten

Da es bei einigen Kunden regelmäßig vorkommt, dass der Actionserverdienst nicht mehr ordnungsgemäß ausgeführt wird, sollte das nachfolgenden Script `net stop <Dienst>' 'net start <Dienst>` in den Taskmanager gehängt werden, damit dieses einmal am Tag ausgeführt wird.

Vorlage für Aufgabenplanung: netstop_startMesonicDienst.xml
Vorlage Bat-Datei: netstop_start.bat

Fehlerüberwachung

Eine Echtzeit-Protokollierung von Programmfehlern und Überwachung der Performance ist über die Integration von Sentry *(https://sentry.io/)* möglich. Um diese zu aktivieren wird die Datei `sentry.ini` in einem der folgenden Ordnern abgelegt.

```
{AUSFÜHRUNGSVERZEICHNIS}\energy-connect-webshop\sentry.ini
C:\ProgramData\New Energy GmbH\Sync-Shop\sentry.ini
C:\Users\{AKTUELLER_USER}}\AppData\Local\New Energy GmbH\Sync-Shop\sentry.ini
```

sentry.ini

```
[Sentry]
;URL an die Fehler und Leistungsdaten übermittelt werden
dsn=https://66abe634323a4cbf827224c253b2bba2@sentry.baseline-bremen.local/3
;Gruppierung nach Server
server_name=meinservername
;Gruppierung unter dieser Umgebung
environment=test
;Anzahl der Anfragen für die Performancedaten erhoben werden 1.0 = 100%
traces_sample_rate=1.0
;Eigene SSL-Zertifikate
;ca_certs=pfad/zu/zertifikat.pem
```

OrderSync mit EXIM

Schnittstelle schreibt abgerufene Belege in Buchungstabelle.

Belege werden per WinLine Batchbelege importiert.

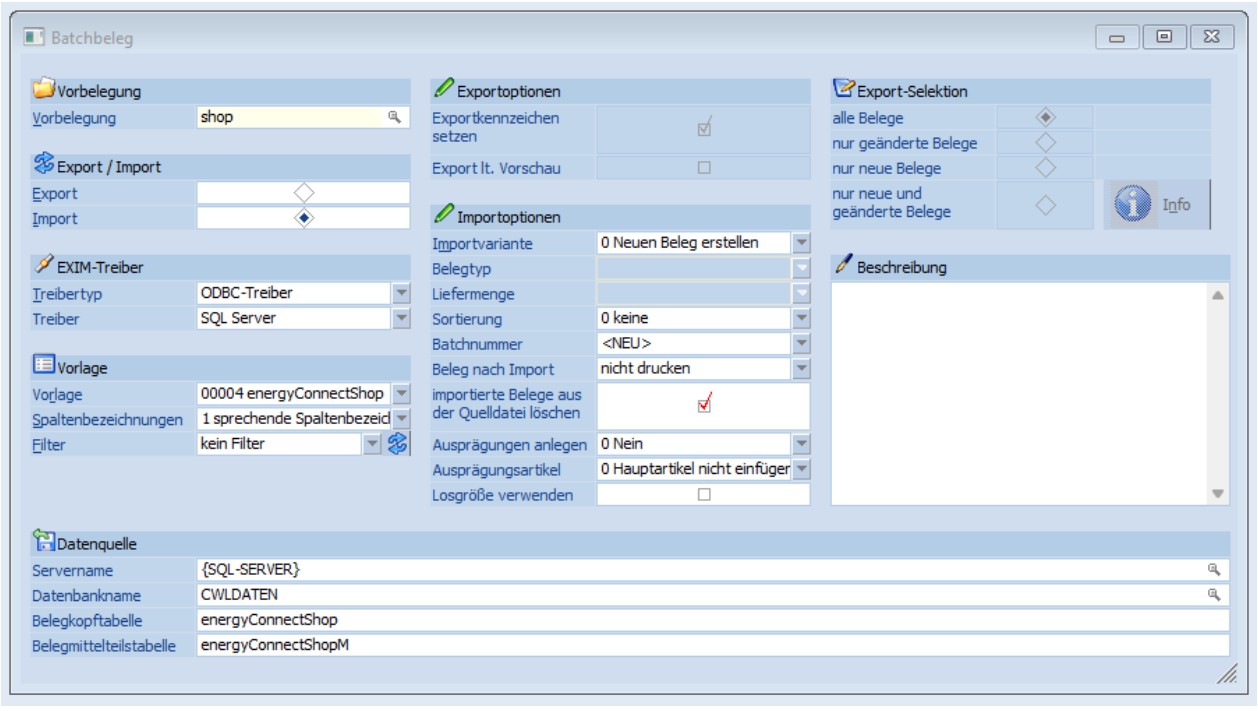
Konfiguration in der Schnittstelle

Config

Um die Belege per EXIM in die WinLine zu importieren, muss der Konfig-Wert `orders_via_exim` auf `True` gesetzt werden und die Importtabellen für Beleg-Kopf und Beleg-Mitte hinterlegt werden. Um die Tabellen anzulegen, siehe Kapitel `Tabellen durch WinLine anlegen lassen`

```
[WL]
...
orders_via_exim = True
exim_order_head = energyConnectShop
exim_order_body = energyConnectShopM
```

Tabellen durch WinLine anlegen lassen



siehe WinLine Doku: [WinLine FAKT > Bearbeiten > Batchbeleg]

BatchBeleg

siehe WinLine Doku: [WinLine FAKT > Bearbeiten > Batchbeleg]

Bestellimport Plugins

Rechnungskauf

Mit dem Plugin `rechnungskauf.py` können die von PayPal Rechnungskauf übergebenen Konto daten mit auf einen Beleg als Textposition importiert werden. Dafür muss in der `order.yml` für die Positionen ein extra Mapping angelegt werden:

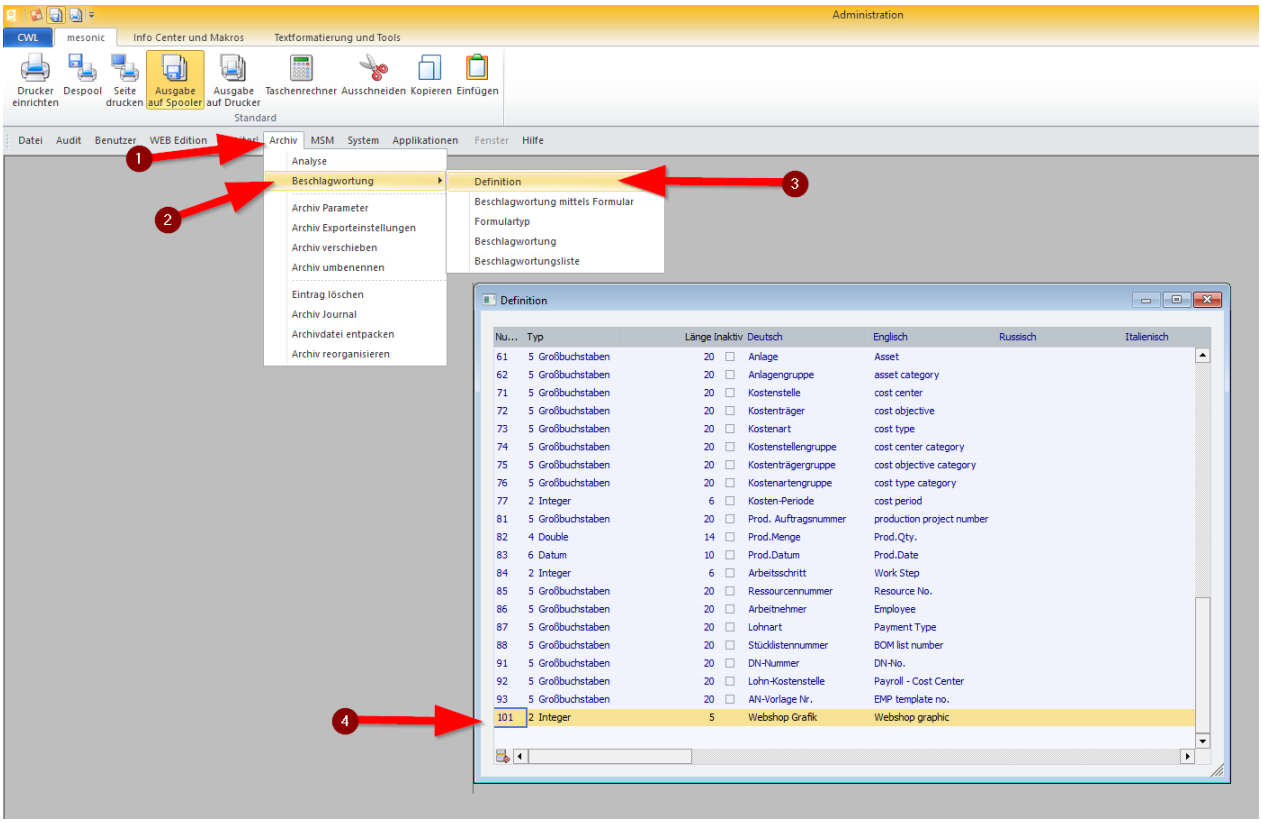
```
Order_Beleg_Positionen_paymentInfo:
  Datentyp:
    winline: Datentyp
    default: 3
  referenz:
    winline: <Feld in das der Text soll>
    shop: payment_reference
Order_Beleg_Positionen_paymentInfo_withText:
  Datentyp:
    winline: Datentyp
    default: 3
  Bankverbindung:
    winline: <Feld in das der Text soll>
    shop: deposit_bank_details
    env:
      - |
        bic = wl_data['bic']
        bic_str = f'BIC:'
        iban = wl_data['iban']
        iban_str = 'IBAN:'
        bank = wl_data['bank_name']
        bank_str = 'BANK:'
        account = wl_data['account_holder_name']
        account_str = 'EMPFAENGER:'
    eval: f'{bic_str} {bic}\n{iban_str} {iban}\n{bank_str} {bank}\n{account_str}
```

Die so angelegten Positions Mappings müssen dann noch um alle Felder ergänzt werden die im normal import vorhanden sind.

Weitere Einrichtungen innerhalb der WinLine

Beschlagwortung

Die Beschlagwortung muss um eine Definition für Webshop-Grafiken ergänzt werden. Diese kann im WinLine-Admin angelegt werden.

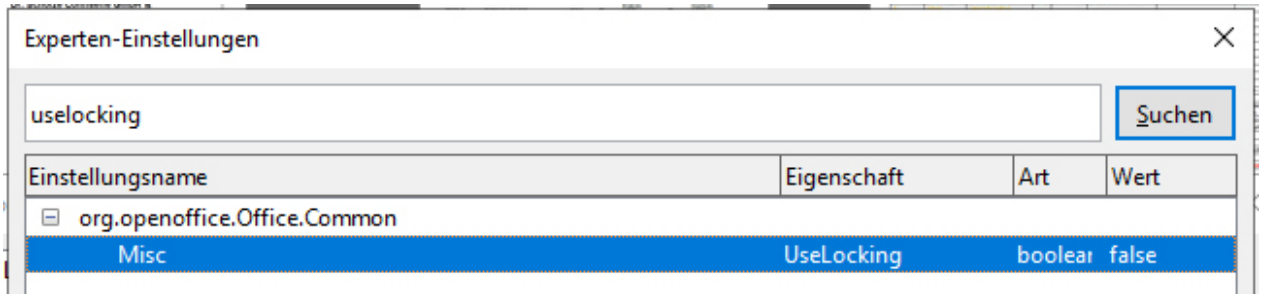


LibreOffice Installation - Beschreibungstexte konvertieren (optional)

LibreOffice muss nur installiert werden wenn eine Konvertierung von RTF-Beschreibungstexten in der WinLine in das HTML-Format des Web-Shops stattfinden soll.

LibreOffice installieren. https://de.libreoffice.org/download/download/ *(https://de.libreoffice.org/download/download/)*

Damit die Schnittstelle einwandfrei funktioniert, muss LibreOffice so konfiguriert sein, dass keine `lock`-Dateien angelegt werden. `Tools -> Options -> LibreOffice -> Advanced -> Button: Open Expert Configuration` Suche nach `UseLocking` und setzte diesen Wert auf `false` (doppel klick auf den Wert)



Pfad in Konfigurations-Datei eintragen:

```
[LIBREOFFICE]
sofficepath = C:\Program Files\LibreOffice\program\soffice.exe
```

Konfiguration und Mapping

Zur Konfiguration die Dateien `config.ini` aus dem Ordner Beispiel-Konfiguration in einen neuen Order "config" kopieren. Darüber hinaus die Mapping-Dateien `order.yml`, `article.yml` und `index.yml` einen einen Unterordner "config/mapping" kopieren. Pfadangabe in Config

```
; Mapping-Pfad innerhalb des Installationsverzeichnisses
MAPPING_PATH = C:\Program Files (x86)\energy-shop-sync-sentry\mapping\shopware
```

Für den Kundenimport und mögliche Umsetzungen, die Beispieldateien `add_tags.json` und `refactor_xml.json` in "config/xml_preferences" kopieren. Pfadangabe in Config

```
xml_preferences = C:\Program Files (x86)\energy-shop-sync-sentry\mapping\shopware
```

Management Interface als Windows-Service

Das Management Interface läuft als Windows-Service mit WinSW *(https://github.com/winsw/winsw)* als Wrapper und Waitress *(https://pypi.org/project/waitress/)* als Produktions-WSGI-Server.

Nach der Installation liegen im Programmverzeichnis (z.B. `C:\Program Files (x86)\energy shop-sync\`) neben `ess.exe` zusätzlich:

- `management_service.exe` – WinSW-Wrapper-Binary
- `management_service.xml` – Service-Definition
- `init.bat` – Erstkonfiguration (`ess.exe init`) mit automatischer UAC-Anforderung
- `install_service.bat` – Service registrieren und starten mit automatischer UAC-Anforderung

Empfohlen – Doppelklick auf `install_service.bat`. Das Skript fordert automatisch Adminrechte an und ruft anschließend `management_service.exe install` und `start` auf.

Alternativ – manuell in einer Administrator-Shell:

```
management_service.exe install
management_service.exe start
```

Anschließend ist das Management Interface unter `http://<host>:8001/` erreichbar. Der Dienst startet automatisch bei Systemstart (`delayedAutoStart`) und wird bei Fehlern nach 60 Sekunden neu gestartet.

Service deinstallieren:

```
management_service.exe stop
management_service.exe uninstall
```

Port oder Proxy-Optionen anpassen: Die Argumente in `management_service.xml` (`<arguments>manage serve</arguments>`) um Flags erweitern. Verfügbare Optionen:

- `-p <port>` – alternativer Port (Default: 8001)
- `-t <trusted-proxy-ip>` – Trusted-Proxy-IP für Reverse-Proxy-Setups
- `-a <auth-header>` – Header für den angemeldeten Benutzer (Default: `REMOTE_USER`)
- `-ph <proxy-header>` – Proxy-Indikator-Header (Default: `X_FORWARDED_FOR`)

Beispiel für IIS/nginx vor dem Interface mit übernahme des angemeldeten Windows-Benutzers:

```
<arguments>manage serve -t 10.0.0.5 -a REMOTE_USER</arguments>
```

Aufgabenplaner von Windows

Anstatt den Sync-Prozess als Service zu installieren, kann der Aufgabenplanner von Windows verwendet werden.

Hierbei sind einige Punkte zu beachten:

- Für jeden Synchronisierungsweg muss eine eigene Aufgabe erstellt werden (Bestell Sync, Artikel Sync, etc.)
- Das Ausführen der Aufgaben muss unabhängig der Benutzeranmeldung ausgeführt werden (es werden Adminrechte benötigt um diese Option zu aktivieren)
- Beim erstellen der Aktion muss das Ausführungsverzeichnis angegeben werden, da dort die Log-Datei erstellt wird.
- Beim Trigger wird empfohlen den allgemeinen Artikel-Sync 1 mal pro Tag auszuführen und den Bestellung-Sync ca. alle 15 min.
- Es wird empfohlen, solange eine Aufgabe noch läuft keine neue Instanz zuzulassen. Wird die gleiche Aufgabe mehr als einmal ausgeführt kann es bspw. zu Abweichungen im Bestand kommen.

Musteraufgaben: Bestands-Sync Order Sync Product Sync Product Sync all Management Interface

Sync Parameter

Den Synchronisationen können/müssen unterschiedliche Parameter mitgegeben werden, um entweder die gewünschten Daten zu übermitteln oder ggf. Daten auszulassen, die nicht übermittelt werden sollen.

Beispiel: `"C:\Program Files (x86)\energy shop-sync\ess.exe" syncProducts -a -wa -wc -cp "C:\Program Files (x86)\energy shop-sync\configs\demoshop.wantedweb.de.ini" -dddd`

-a = alle Artikel werden übermittelt, unabhängig ihres Änderungsdatums -wa = -wc = “without Categories” - ohne Übertragung der WinLine Artikeluntergruppen als Shop-Kategorien -n “Artikelnummer” = ein einzelner Artikel wird übermittelt -cp = -si = skip Images - keine Bildübertragung -sd = skip Download - keine Downloadübertragung -dddd = Debuglevel

Benutzung der Parameter

```
usage: ess.exe [-h] [-d] [-n ARTNR | -a] [-om] [-wc]
               [-wa] [-we] [-cp CONFPATH]
               [-cc CACHENAME] [-m] [-ai] [-sd] [-si]
               [-st] [-os] [-fv] [-vn] [-to DIRECTION]
               [-on] [-whc] [-np] [-odu]
               [-ep ENDPOINT] [-g] [-j JSON_DATA]
               [-jf JSON_FILE]
               [{syncProducts,syncOrders,syncCategories,syncAll,syncAttr,
                 syncPrices,syncCross,syncCustomers,syncBestand,
                 syncProperties,clearCache,test,version,syncImages,
                 syncDownloads,syncTranslations,syncFilteredCustomers,
                 syncCustomersWLNumber,syncCustomersWLtoSW,
                 validateArticleNumbers,preGenerateHTML,syncOrderStatus,
                 syncShipmenttype,syncCustomersAndOrders,healthCheck,
                 send,init,manage}]

Kommandlineinterface for Energy Connect Shop

positional arguments:
  init                  Legt nötige Dateien und Beispielkonfigurationen ab.
  syncProducts          Synchronisiert Artikel
  syncOrders            Synchronisiert Bestellungen
  syncCategories        Synchronisiert Kategorien
  syncAll               Synchronisiert Bestellungen, Kategorien und Produkte
  syncPrices            Synchronisiert die Preise der Artikel (ink. kundenspezifi
  syncCustomers         Synchronisiert die Kunden aus der Winline in den Shop
  syncBestand           Synchronisiert den Lagerstand aller Webartikel
```

```
  syncProperties        Synchronisiert die Artikeleigenschaften
  syncImages            Synchronisiert die Stammdaten/Archivbilder aller Webartik
  syncDownloads         Synchronisiert die Downloaddateien aller Webartikel
  syncTranslations      Synchronisiert die Artikelbezeichnungen und Artikelbeschr
  syncFilteredCustomers Synchronisiert Kundendaten die einen Tag zuvor geändert w
  syncCustomersWLNumber Synchronisiert die Winline-Kundennummer in die Shopware
  syncCustomersWLtoSW   Synchronisiert Kunden von der Winline in die Shopware
  syncOrderStatus       Synchronisiert Orderstatus von der Winline in die Shopwar
  syncShipmenttype      Synchronisiert Versandart von der Winline in die Shopware
  syncCustomersAndOrders Synchronisiert erst Kunden aus Shopware und dann Bestell
  syncCross             Synchronisiert die Cross-Selling-Artikel
  clearCache            Löscht den gesamten Shop Cache
  version               Zeigt Versionsinformationen an
  validateArticleNumbers Prüft ob Artikelnummern nach Konvertierung zurückgewande
  preGenerateHTML       Konvertiert RTF-Texte der Webartikel in einen Cache
  healthCheck           Testet die Verbindungen und Konfigurationen auf Probleme
  send                  Schickt einen Request an den konfigurierten Shop
  manage                Fuehrt Django-Management-Kommandos aus (z.B. manage migra

options:
  -h, --help            Hilfe anzeigen
  -d, --debug           Loglevel (-d: ERROR, -dd: WARNING, -ddd: INFO, -dddd: DEB
  -n ARTNR, --artNr ARTNR
                        Artikelnummer für Einzelsync
  -a, --all             Alle Artikel synchronisieren (ohne Änderungsdatum)
  -om, --onlyMain       Nur Hauptartikel übertragen (bei Einzelsync)
  -wc, --withoutCategories
                        Ohne Kategorien synchronisieren
  -wa, --withAttributes
                        SW5: Mit Attributen synchronisieren
  -we, --withEigenschaften
                        Ohne Eigenschaften synchronisieren
  -cp CONFPATH, --configPath CONFPATH
                        Pfad zu alternativer Config-Datei
  -cc CACHENAME, --clearCache CACHENAME
                        Name des zu löschenden Cache
  -m, --manage          Konsolenausgabe aktivieren
  -ai, --allImages      Alle Bilder erneut übermitteln
  -sd, --skipDownloads  Archiv2 Downloads nicht übermitteln
  -si, --skipImages     Archiv2 Bilder nicht übermitteln
  -st, --syncTranslations
                        Übersetzungen nicht übermitteln
  -os, --oldSync        Ohne AsyncIO synchronisieren
  -fv, --forcevariants  Hauptartikel als Varianten übertragen
```

```
-vn, --validateNumbers
                      Artikelnummern auf Konvertierbarkeit prüfen
-to DIRECTION, --to DIRECTION
                      Ziel für Kundensync ('winline' oder 'shop')
-on, --onlyNew        Nur neue Artikel übertragen
-whc, --withoutHealthCheck
                      Healthcheck überspringen
-np, --noProgress     Progress-Ausgabe unterdrücken
-odu, --onlyDeleteUnused
                      Nur ungültige Preise löschen (syncPrices)
-ep ENDPOINT, --endpoint ENDPOINT
                      Endpunkt für Request (Default: _action/sync)
-g, --get             GET statt POST ausführen
-j JSON_DATA, --json JSON_DATA
                      JSON-Daten für send-Befehl
-jf JSON_FILE, --json_file JSON_FILE
                      Pfad zu JSON-Datei für send-Befehl
```

Shopware

Grundeinstellungen

- Kundengruppe Shopkunden auf Brutto/Netto stellen. Die Kundengruppe wird für den Standardpreis verwendet, falls der Kunde als Sammelkunde bestellt oder keine kundenindividuellen Preise hinterlegt hat.
- Automatischer Vorschlag der Artikelnummer ausschalten.
- Abverkaufsartikel ohne Lagerbestand ausblenden: Nach Wahl, oder in Schnittstellen-Mapping setzen, falls nur nicht bestellbar sein soll.
- Lieferzeit im Warenkorb anzeigen: Nach Wahl.
- Shopware generiert Kundennummern auf “Nein” stellen, falls eine automatische KdNr-Vergabe nicht gewünscht ist (siehe Optionen für den Kundensync).

Shopware 6 API-Integration

Für die Anbindung an Shopware 6 muss eine Integration im Shopware-Admin angelegt werden:

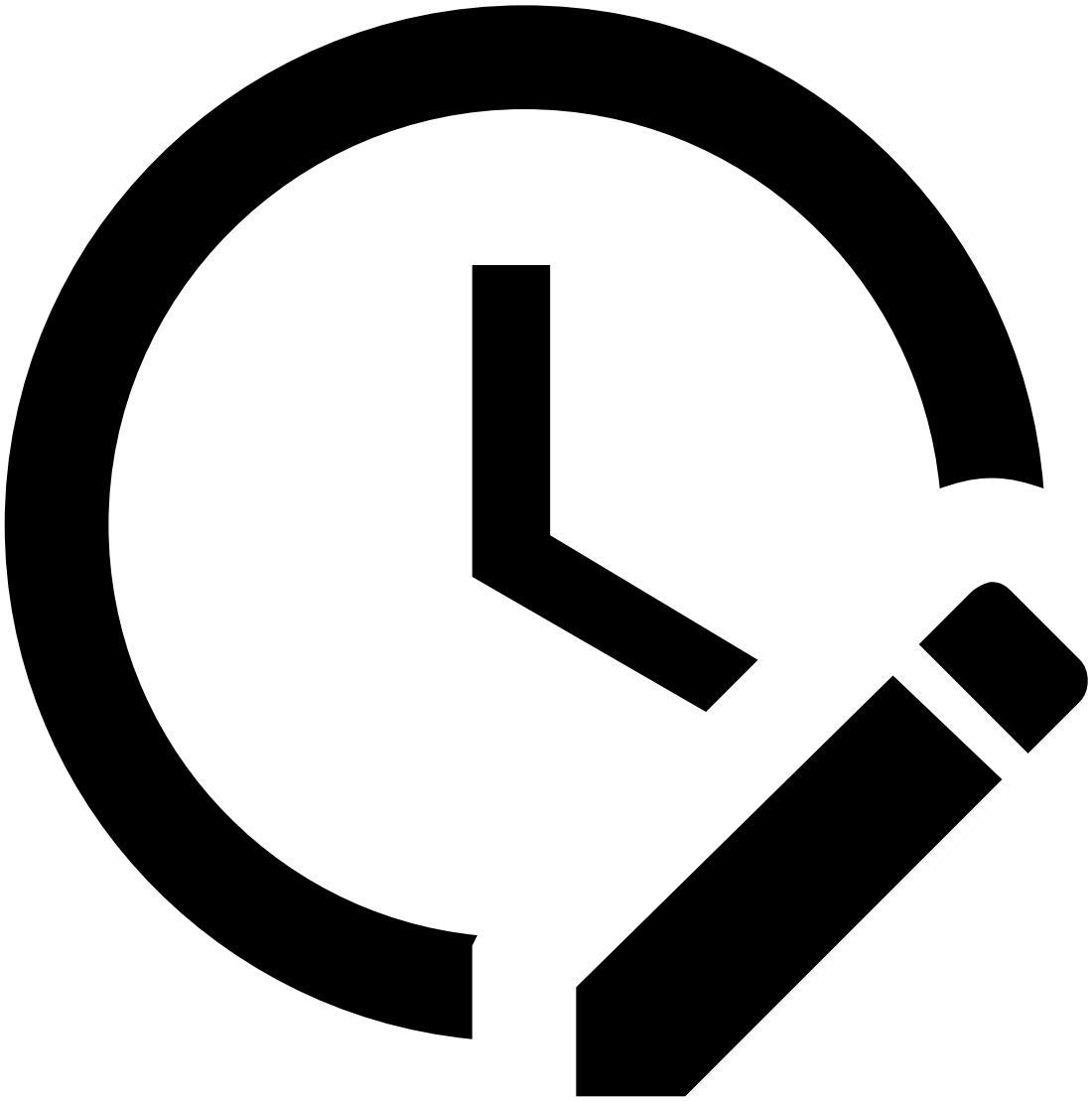
1. Im Shopware-Admin unter **Einstellungen > System > Integrationen** eine neue Integration anlegen
2. Zugangsdaten (Access Key ID und Secret Access Key) notieren
3. Die Zugangsdaten in der Konfigurationsdatei hinterlegen:

```
[SHOP]
url = https://mein-shop.de
consumer_key = SWIA...
consumer_secret = ...
api = shopware6
```

1. Verbindung testen mit `ess.exe healthCheck` — die Ausgabe sollte eine erfolgreiche Shop-Verbindung bestätigen

Hinweis

Eine vollständige Übersicht aller Shop-Verbindungsparameter finden Sie in der Konfigurationsreferenz.



Update

Migration Management Interface (Flask → Django 6)

> **Breaking Change ab v3.0.0**
>
> Ab Version 3.0.0 wird das Management Interface mit Django 6 betrieben. Die bisherige Flask-Version wird nicht mehr unterstützt.

management.ini anpassen

Folgende Sektionen sind obsolet und müssen **entfernt** werden:

Sektion	Grund
`[HOST]`	Server-Konfiguration wird nun von Django verwaltet
`[MANAGEMENT]`	Pfad zur management.ini wird automatisch bestimmt
`[SOCKET]`	WebSocket/SocketIO wurde durch AJAX-basiertes Polling ersetzt
`[PROFILER]`	Profiler-Funktion entfällt
`[SHOPURL]`	Shop-URLs werden automatisch aus der jeweiligen config.ini (`[SHOP]` → `url`) ausgelesen
`[SHOPPATH]`	Launcher-Pfad und config.ini-Pfad sind nun in `[SHOPS]` zusammengefasst

[SHOPS]-Sektion aktualisieren

Die `[SHOPS]`-Sektion enthält jetzt direkt den Pfad zur **config.ini** jedes Shops.

Vorher (Flask):

```
[SHOPS]
local = local

[SHOPPATH]
localpath = __python__
localconfig = conf\config.ini
```

Nachher (Django):

```
[SHOPS]
local = conf\config.ini
```

Startbefehl

	Flask (alt)	Django (neu)
Befehl	`minterface.exe`	`ess.exe manage serve`
Port	5000	8001

`ess.exe manage serve` startet Waitress als Produktions-WSGI-Server. Alternativ steht für Debug-Zwecke weiterhin `ess.exe manage runserver` zur Verfügung (Django-Entwicklungs-Server, nicht für Produktion).

Installation als Windows-Service

Im Installations-Verzeichnis liegen nach dem Setup zwei Dateien:

- `management_service.exe` — WinSW *(https://github.com/winsw/winsw)*-Wrapper
- `management_service.xml` — Service-Definition (Autostart, Restart bei Fehler)

Service registrieren/starten (Administrator-Shell):

```
management_service.exe install
management_service.exe start
```

Deregistrieren:

```
management_service.exe stop
management_service.exe uninstall
```

Port oder Proxy-Optionen lassen sich über das `<arguments>`-Feld in der XML ändern, z.B. `manage serve -p 8080 -t 10.0.0.5`.

Unveränderte Sektionen

Folgende Sektionen bleiben unverändert und müssen nicht angepasst werden:

- `[ACTIVE]` – Standard-Shop
- `[TIMER]` – Sync-Warteschlange und Timeout
- `[SYNCRENAME]` – Sync-Bezeichnungen
- Shop-spezifische Sektionen (z.B. `[LOCAL]`, `[MOSTWANTED]`)

Beispiel: management.ini nach Migration

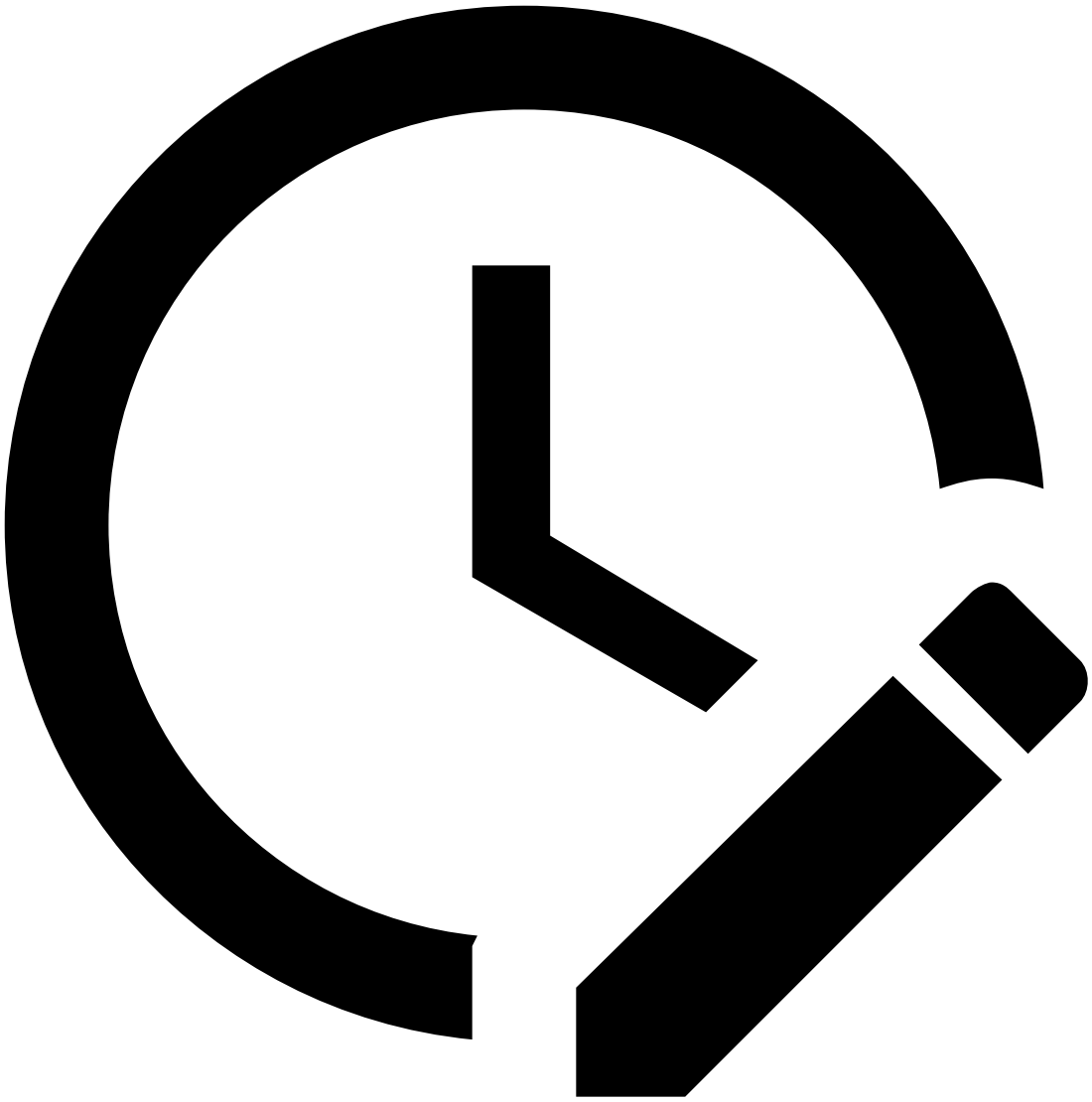
```
[ACTIVE]
default = local

[SHOPS]
local = conf\config.ini
```

```
mostwanted = C:\Program Files\energy shop-sync\config.ini

[TIMER]
wakeup = 1300
overtimekill = 85500

[LOCAL]
debuglvl = -d
-wc = -wc
-wa = -wa
-m =
-ai =
-to = -to wl

[SYNCRENAME]
syncProducts = Artikel
syncProductsAll = Artikel[--all]
syncOrders = Bestellungen
syncCategories = Kategorien
syncAttr = Eigenschaften
syncPrices = Preise
syncCross = Cross-Artikel
syncCustomers = Kunden
syncBestand = Bestand
syncAll = Alles
```

Cache update to version 0.1

Search and replace:

search	replace
`["`	`{"link": "`
`", "`	`", "position": "`
`]`	`}`



2026-04-16

Konfiguration

Die Konfiguration der Schnittstelle erfolgt über eine INI-Datei (z.B. `config.ini`). Die Standardwerte werden aus der `defaultconfig.toml` geladen und können in der INI-Datei überschrieben werden.

> **Optionale Parameter**
>
> Parameter die mit `; Optional` gekennzeichnet sind, haben keinen Standardwert in der `defaultconfig.toml` und müssen nur bei Bedarf in der `config.ini` gesetzt werden. Der angegebene Wert entspricht dem Code-Fallback.

Shop-Verbindung

```
[SHOP]
; URL des Shopware 6 Shops
url = https://mein-shop.de
; API-Zugangsdaten der Shopware 6 Integration
consumer_key = SWIA...
consumer_secret = ...
; Shopware 6 API
api = shopware6
; Anzahl der Artikel pro API-Request
chunk_size = 10
; Anzahl der Preise pro API-Request
chunk_size_prices = 100
; Maximale Anzahl der Ergebnisse pro Suchanfrage
query_limit = 500
; Wartezeit in Sekunden zwischen Requests bei Fehler
retry_wait_time = 5
; Maximale Anzahl der Wiederholungen mit Wartezeit
max_retry_with_wait = 3
; Optional: Root-Kategorie-ID für Kategorie-Sync
root_category_id =
; Optional: Max. gleichzeitige TCP-Verbindungen (Standard: 10)
tcp_limit = 10
```

Shopware 6

```
[SW6]
; Name des Sales Channels
salesChannelName = Storefront
```

WinLine WebService

```
[EWL]
ewluser = a
ewlpw = b
wlewlserviceurl = http://127.0.0.1:80/ewlservice/
```

WinLine Mandant & Datenbank

```
[WL]
; WinLine Mandant
mcomp = 500M
; Kundennummer für Bestellungen ohne oder mit in der WinLine nicht gefundener Kur
sammelkonto = 10000
; Alle Bestellungen auf Sammelkonto buchen
always_sammelkonto = False
; Verhalten bei fehlenden WL-Konten: ERROR, SKIP oder SAMMELKONTO
missing_wl_konto = ERROR
; Eigenschaftsfeldnummer für Webartikel-Markierung
webartikel = 5
; Eigenschaftsfeldnummer für Webkonto-Markierung (Personenkonto)
webkonto = 1104
; Eigenschaftsfeldnummer für Webkonto-Markierung (Kontakt)
webkonto_kontakt = 1
; Eigenschaftsfeldnummer für den Webkatalog
webkatalog = 1009
; Eigenschaftsfeld-Spalte für Shopware-Konfiguration (Standard: 14)
webeigenschaft = 14
; Preisfaktor aus dem Colli verwenden
preisfaktorauscolli = True
; Preislisten-Modul verwenden
syncprices_module = True
; interne WinLine Nummer(n) für die Webshop Preisliste(n) (kommagetrennt)
preislistennr = 1
; Default-Währung ISO-Code (leer = Mandant-Währung)
defaultwaerungiso =
```

```
; Shop-Variante aktivieren
shopvariante = False
; Inaktive Ausprägungen nicht übermitteln
skip_inaktiv_variants = True
; Ausprägungsgruppen die nicht als Varianten angelegt werden (kommagetrennt)
ignoreauspraegungsgruppen = 0
; Kategorien aus Eigenschaft (Nummer der Eigenschaft)
categories =
; Kategorien aus Artikeluntergruppen generieren
Kategorien_aus_Artikeluntergruppen = False
; Richtung der TrackingNr Übermittlung (True = Shop führend, False = WinLine führ
trackingnrtowinline = False
; WinLine Feld für Sendungsverfolgungsnummer
trackingnrfield =
; Feld für die Shop-Bestellnummer im WinLine-Beleg
shoporderid = C063_Textzeile1
; Feld für den Bestellstatus im WinLine-Beleg
orderstatus = C071_Textzeile9
; Feld für den Bezahlstatus im WinLine-Beleg
paymentstatus = C072_Textzeile10
; Feld für den Lieferstatus im WinLine-Beleg
deliverystatus = C071_Textzeile9
; Nutze Rechnungsadresskonto aus dem Personenkonto für die Belege
rechnungsadresskonto = True
; Optional: Nutze Lieferadresskonto aus dem Personenkonto für die Belege
lieferadresskonto = False
; Encoding der XML-Dateien
encoding = utf-8
; Bestellungen per EXIM importieren
orders_via_exim = True
; EXIM Tabelle für Beleg-Kopf
exim_order_head = energyConnectShop
; EXIM Tabelle für Beleg-Mitte
exim_order_body = energyConnectShopM
; Testmodus (keine Daten an Shop übermitteln)
test_mode = False
; Verzeichnis für XML-Export (leer = WebService verwenden)
xmlpath =
; Sammelkonto für Gastbestellungen
guest_sammelkonto = 10000
; WinLine-Feld für Shop-Belegnummer
shopbelegnummernfeld = C064_Textzeile2
; Artikelnummer statt ID verwenden
usenumberinsteadofid = False
```

```
; Shopware-Feld für Tracking-Nummer
shoptrackingnrfield = trackingCodes
; Kundenimport per EXIM
customers_via_exim = True
; EXIM-Tabelle für Kunden
exim_customers = EXIM_KONTAKT
; Optional: Rechnungen nach Trackingnr. prüfen (Standard: False)
check_invoices_for_tracking = False
; Optional: Feld für Shopware-Kunden-ID (Standard: C002_Kontonummer)
customer_shopware_id_field = C002_Kontonummer
; Optional: Verzeichnis für Kunden-XML-Export (Standard: ./xml/customers)
xmlpath_customers = ./xml/customers
; Optional: Zusatzfeld für Shopware-ID Kontakte
swid_kontakt =
; Optional: Zusatzfeld für Shopware-ID Personenkonten
swid_pkonto =
; Optional: Filter-ID-Länge (Standard: 0)
filter_id_length = 0
```

SQL-Verbindung

```
[SQL]
driver = ODBC Driver 18 for SQL Server
server = localhost
database = CWLDATEN
uid = meso
pwd = meso
charset = utf8
; Maximale Anzahl SQL-Parameter pro Query
max_sql_parameter = 2000
; SQL-Debug-Modus
debug = False
; Temp-Tabellen mit doppeltem Hash
temptable_two_hash = False
; Verschlüsselte Verbindung
encrypt = True
; SSL-Zertifikat immer vertrauen (nur für Entwicklung)
trustservercertificate = True

[SQLSYS]
database = CWLSYSTEM
; uid/pwd werden per Referenz aus [SQL] geerbt (${SQL:uid}, ${SQL:pwd})
```

```
[SQLARCHIV]
; database muss auf die Archiv-Datenbank gesetzt werden (kein Standardwert in def
; uid/pwd werden per Referenz aus [SQL] geerbt (${SQL:uid}, ${SQL:pwd})
```

Shop-Felder

```
[SHOPFIELDS]
; Lieferadresse aus dem Shop übernehmen
addressfromshop = True
; Rechnungsadresse aus dem Shop übernehmen
billfromshop = True
; WinLine Vorlage mit Liefer- und Rechnungsadresse
wlvorlageLR = wooCommerceAdressenLR
; WinLine Vorlage ohne Adressen
wlvorlageKA = sw_noadd_nodis
; Mitarbeiterkonten verwenden
mitarbeiterkonten = False
; WebService-Vorlage Kunden
wlvorlagekunden = Shopware6Kunden
; WebService-Vorlage Kontakte
wlvorlagekontakt = Shopware6Kontakt
```

Bestellungsimport

```
[ORDERS]
; Reihenfolge der Belegabrufe (JSON-Array mit Segmentnamen)
import_sequences = ["Bestellung"]

[Bestellung]
; Optional: Statustyp für die Suche (OrderState, order_delivery, order_transactio
status_type = OrderState
; Status zum Abrufen der Bestellungen
status_fetch = OrderState.open
; Status nach erfolgreichem Import
status_success = OrderState.inProgress
; Status bei Importfehler
status_fail = OrderState.open
; WinLine Belegstufe (nicht_drucken, Angebot, Auftrag, Lieferschein, Rechnung)
import_belegstufe = Lieferschein
; Prüfen ob Bestellung bereits importiert wurde
check_existing = True
```

Lieferung

```
[Lieferung]
; Status zum Abrufen
status_fetch = OrderDeliveryState.open
; Status nach erfolgreichem Import
status_success = OrderDeliveryState.shipped
; Status bei Importfehler
status_fail = OrderDeliveryState.open
; Status nach Tracking-Übermittlung
status_tracking = OrderDeliveryState.shipped
; WinLine Belegstufe
import_belegstufe = Lieferschein
```

Bezahlung

```
[Bezahlung]
; Status zum Abrufen (kommagetrennt für mehrere)
status_fetch = OrderTransactionState.refunded,OrderTransactionState.refunded_part
; Status nach erfolgreichem Import
status_success = OrderTransactionState.inProgress
; Status bei Importfehler
status_fail = OrderTransactionState.inProgress
; WinLine Belegstufe
import_belegstufe = Rechnung
; Gutschrift erstellen
gutschrift = False
```

Shop-Eigenschaften

```
[SHOPPROPERTY]
; Lieferstatus nach Tracking-Übermittlung
orderstatus_tracking = OrderDeliveryState.shipped
```

E-Mail-Benachrichtigungen

```
[MAIL]
; SMTP-Server
server =
; SMTP-Port
port = 0
; STARTTLS verwenden
use_tls = False
```

```
; SMTP-Benutzername
username =
; SMTP-Passwort
password =
; Betreff-Prefix für Fehlerbenachrichtigungen
subject_prefix = [SHOPWARE FEHLERLOG]
; Absender-Adresse
sender =
; Empfänger-Adresse
receiver =
; Maximale Anzahl gepufferter Nachrichten
capacity = 1000
; Log-Level für E-Mail-Versand
level = DEBUG
; E-Mail-Versand aktivieren
tosend = False
```

Archiv

```
[ARCHIV]
; interne WinLine ID (Archiv) für das Schlagwort: Webshop Grafik
archiv_webshop_grafik = 102
; interne WinLine ID (Archiv) für das Schlagwort: Webdokument
archiv_webdokument = 103
; interne WinLine ID (Archiv) für das Schlagwort: Artikelnummer
archiv_artikelnummer = 41
; interne WinLine ID (Archiv) für das Schlagwort: Artikelbezeichnung
archiv_beschreibung = 42
; Optional: Liste der Zusatzfelder für zusätzliche Grafiken (kommagetrennt)
zusatzgrafiken =
```

Bildkonvertierung

```
[CONVERT]
; Zielformat für LibreOffice Konvertierung
targetformat = html:HTML:EmbedImages
; Maximale Bildgröße (Breite,Höhe)
scale_images = 1280,720
; Hintergrundfarbe für Bilder (R,G,B)
background_color = 255,255,255
```

Artikeleigenschaften

```
[PROPERTY]
; Bedarfslistenbestand verwenden
bedarfslistenbestand = False
; Artikelmedien aus der WinLine verwenden
useartiklemedia = False
; Archiv-Bilder verwenden
usearchivimages = True
```

Cross-Selling

```
[CROSS]
; Cross-Selling aktivieren
crossselling_aktiv = True
; Keine Synchronisation bei Wert
notsync = 0
; Prioritätsbereich für Ersatzteile
ersatzteilevon = 1
ersatzteilebis = 99
; Prioritätsbereich für Zubehör
zbhrvon = 100
zbhrbis = 199
; Prioritätsbereich für Ersatz & Zubehör
ezvon = 200
ezbis = 299
```

Dynamisches Cross-Selling

```
[DYNACROSS]
; Name der Cross-Selling-Gruppe im Shop
cross_name = Unsere Vorschläge
; WinLine Zusatzfeld für die Gruppen-Zuordnung
cross_gruppe_feld = C229_Zusatzfeld29
; WinLine Zusatzfeld für die Anzeige-Zuordnung
cross_anzeige_feld = C230_Zusatzfeld30
; Sortierung (cheapestPrice / name / releaseDate)
sort_by = cheapestPrice
; Sortierrichtung (ASC / DESC)
sort_direction = ASC
```

Mehrsprachigkeit

```
[LANGUAGES]
; Liste der Sprachen (JSON-Array)
languages = ["Deutsch"]

[Deutsch]
; Name des Sales Channels für diese Sprache
salesChannelName =
; Beschreibungsfeld-Nummer
descriptionfield = 0
; Position der Sprache
position = 1
; Separate Kategorien für diese Sprache
separate_categorie = False
; Hauptsprache
primary_shop = True
; Locale-Code
local_code = de-DE
; Encoding
encoding = utf-8
```

Mapping

```
[MAPPING]
; Pfad zum Mapping-Ordner
mapping_path = conf\mapping\
```

Sonstige Einstellungen

```
[OTHER]
; Temporärer Ordner
tmppath =
; Pfad zu XML-Preferences für den Kundenimport
xml_preferences = conf\mapping\xml_preferences
; Bestellungen als Batch-XML
batch_orders = False
; Optional: Systemzertifikate für SSL verwenden (Standard: False)
truststore = False
; Optional: Kunden als Batch verarbeiten (Standard: False)
batch_customers = False
; Optional: Statische Dateinamen verwenden (Standard: False)
```

```
static_file_name = False

[LIBREOFFICE]
; Pfad zu LibreOffice
sofficepath = C:\Program Files\LibreOffice\program\soffice.exe
```

Feature-Flags

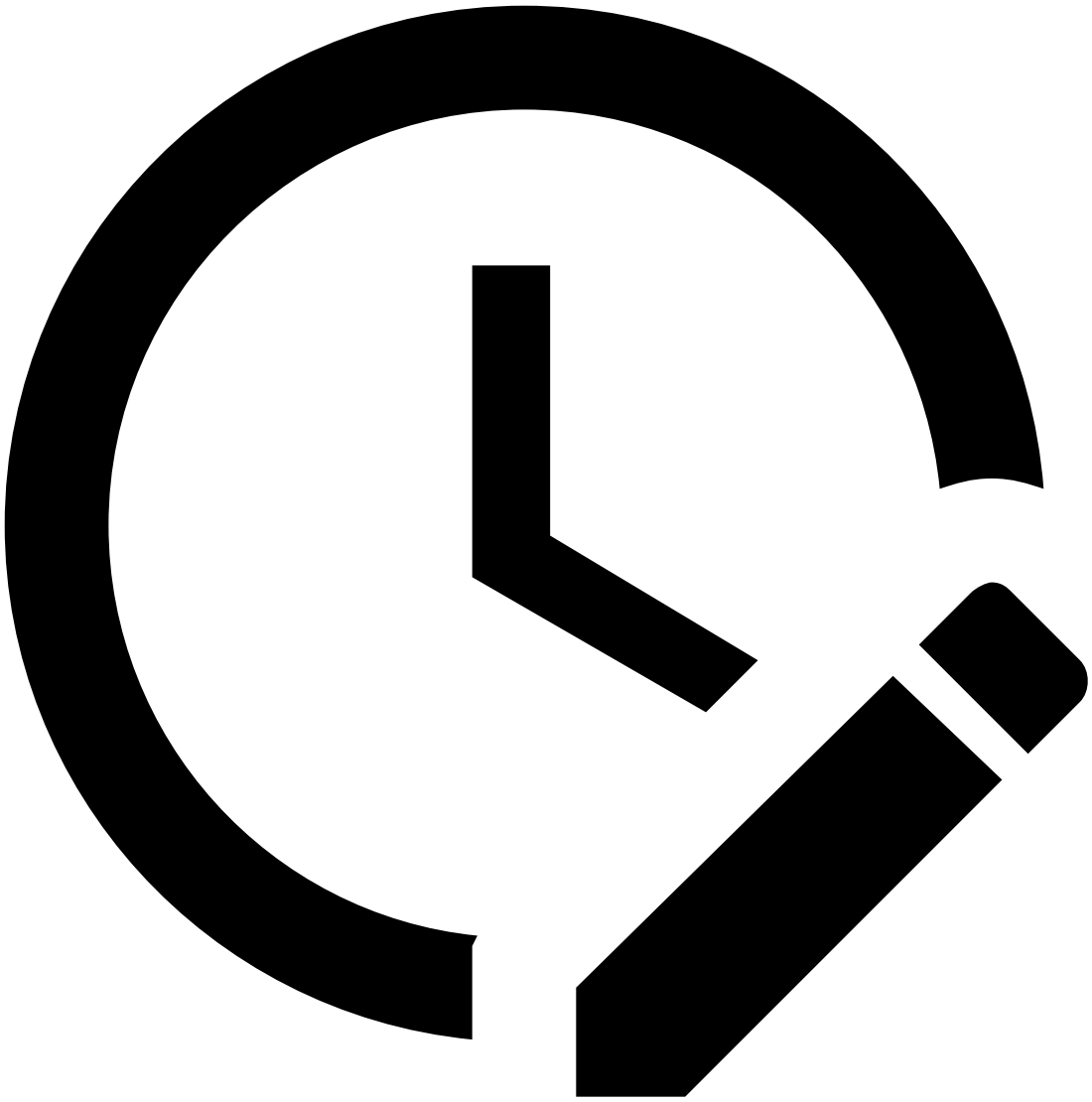
Über Feature-Flags können einzelne Funktionen der Schnittstelle aktiviert oder deaktiviert werden.

```
[FEATURES]
; Menge2 aus der WinLine als Preis verwenden
MENGE2 = False
; UVP aus dem allgemeinen Brutto-Preis ermitteln
UVP_FROM_ALLGEMEIN_BRUTTO = False
; UVP aus dem allgemeinen Netto-Preis ermitteln
UVP_FROM_ALLGEMEIN_NETTO = False
; Höchster Preis hat Priorität
PRICE_PRIORITY_HIGHEST = False
; Dynamisches Cross-Selling aktivieren
DYNAMIC_CROSS_SELLING = False
; Mapping für den Kundenimport verwenden
USE_MAPPING_FOR_CUSTOMERS = True
; Preisduplikate entfernen (z.B. bei überschneidender Gültigkeit)
REMOVE_PRICE_DUPLICATES = True
; Rabattmatrix-Preise zusammenfassen statt kundenindividuell zu übertragen
SUMMARIZE_ACCOUNT_MATRIX = True
; Kunden als Kontakte/Ansprechpartner in der WinLine anlegen
CUSTOMERS_AS_CONTACT = True
; Bestellungen per WebService in das aktuelle Wirtschaftsjahr importieren
WEBSERVICE_COMPANYYEAR = True
```

Exklusive Features

Einige Features schließen sich gegenseitig aus. Folgende Kombinationen sind **nicht** gleichzeitig verwendbar:

- `MENGE2` und `UVP_FROM_ALLGEMEIN_NETTO`
- `MENGE2` und `UVP_FROM_ALLGEMEIN_BRUTTO`
- `UVP_FROM_ALLGEMEIN_BRUTTO` und `UVP_FROM_ALLGEMEIN_NETTO`
- `MENGE2` und `PRICE_PRIORITY_HIGHEST`
- `UVP_FROM_ALLGEMEIN_NETTO` und `PRICE_PRIORITY_HIGHEST`
- `UVP_FROM_ALLGEMEIN_BRUTTO` und `PRICE_PRIORITY_HIGHEST`



2026-03-25

Funktionen

Webartikel

Überblick

Der Artikelsync überträgt Produktdaten aus der WinLine in den Shopware-Shop. Die Synchronisation erfolgt **in eine Richtung** (WinLine → Shopware) – Änderungen im Shop werden nicht zurück in die WinLine geschrieben.

Es stehen zwei Sync-Modi zur Verfügung:

- **Differenzieller Sync** – Überträgt nur Artikel, die seit dem letzten Sync geändert wurden. Dies ist der Standardmodus.
- **Vollsync** – Überträgt alle als Webartikel markierten Artikel, unabhängig vom Änderungsdatum.

Zusätzlich kann ein **einzelner Artikel** gezielt über das Management Interface synchronisiert werden.

Workflow

Der typische Ablauf für die Artikelsynchronisation:

1. **Artikel in WinLine als Webartikel markieren** – Die Webartikel-Eigenschaft auf `1` setzen. Artikel mit `0` oder leerem Wert werden ignoriert, Artikel mit `-1` werden im Shop deaktiviert.
2. **Sync starten** – Über das Management Interface, einen geplanten Job oder die Kommandozeile (CLI).
3. **Schnittstelle liest alle markierten Artikel** aus der WinLine aus.
4. **Nur geänderte Artikel werden übertragen** (bei differenziellem Sync). Beim Vollsync werden alle markierten Artikel übertragen.
5. **Neue Artikel werden im Shop angelegt**, bestehende aktualisiert und mit `-1` markierte Artikel deaktiviert.
6. **Ergebnis im Shop prüfen.**

> **Ersteinrichtung**
>
> Nach der Ersteinrichtung oder nach Änderungen an Mappings sollte immer ein **Vollsync** durchgeführt werden, um sicherzustellen, dass alle Artikel korrekt übertragen werden.

Verwendung

Differenzieller Sync

Überträgt nur Änderungen seit dem letzten Sync. Empfohlen für regelmäßige bzw. geplante Syncs.

Vollsync

Überträgt alle Webartikel unabhängig vom Änderungsdatum. Wird über `syncProductsAll` bzw. den CLI-Befehl `Artikel --all` ausgelöst. Empfohlen nach Ersteinrichtung oder Mapping-Änderungen.

Einzelartikel-Sync

Ein einzelner Artikel kann über das Management Interface per Artikelnummer synchronisiert werden. Dies ist hilfreich, um einzelne Artikel gezielt zu testen oder zu aktualisieren.

WinLine

Einrichtung

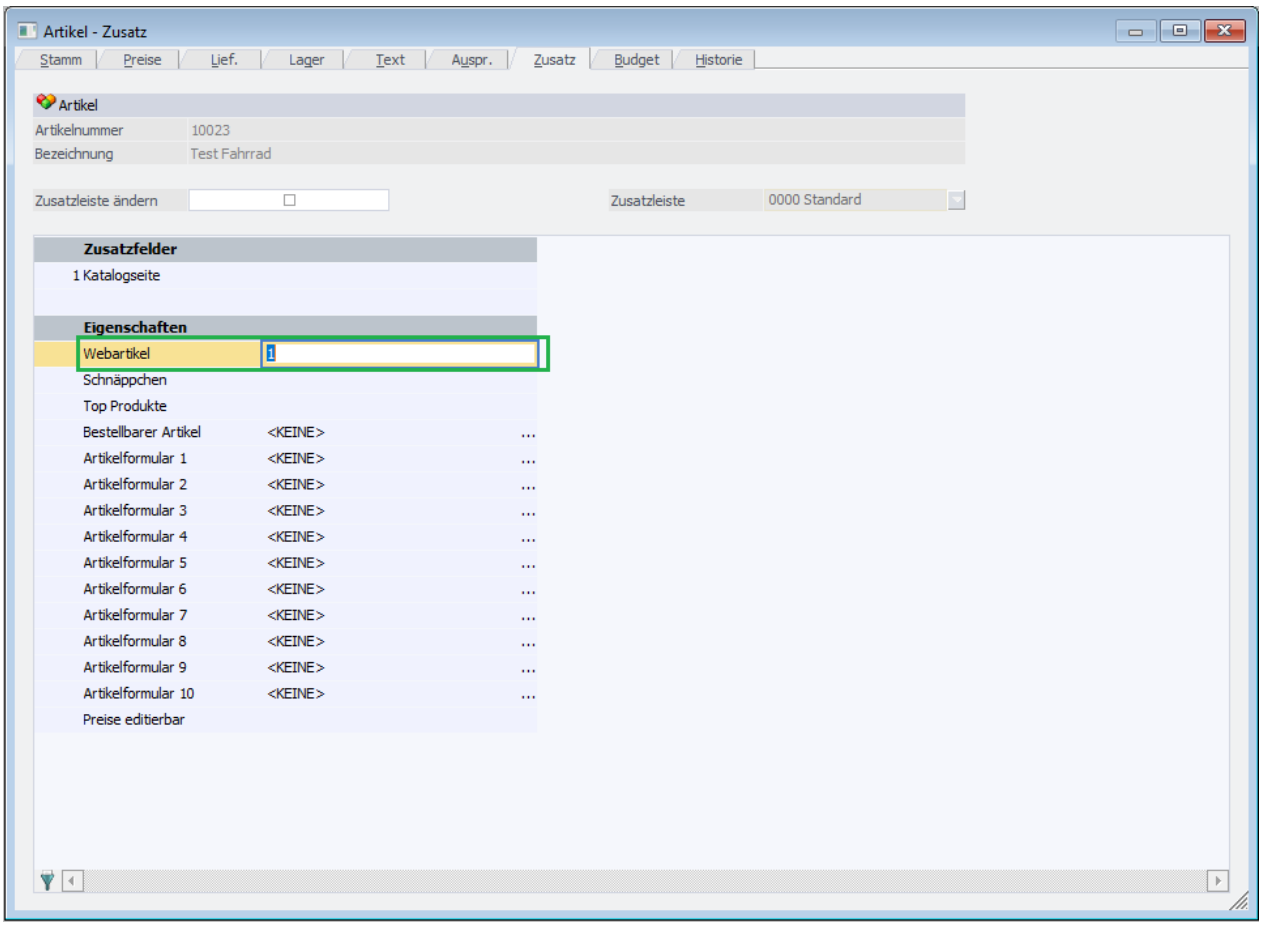
Eine Einrichtung ist nicht notwendig, sofern das schon vorhandene Eigenschaftsfeld “Webartikel” verwendet wird. Es ist zu empfehlen bei der Einrichtung mit einer Testeigenschaft zu arbeiten, um den “syncAll” anhand weniger Artikel zu testen.

Die Webeigenschaft sollte immer als Typ 2 (Integer) angelegt sein. Mögliche Optionen:

Wert	Beschreibung
1	Artikel synchronisieren
0/leer	Artikel wird nicht berücksichtigt
-1	Artikel wird im Shop “inaktiv” gesetzt

Verwendung

Um Artikel an den Shop zu übermitteln, müssen diese in der WinLine als Webartikel markiert werden



Schnittstelle

config.ini

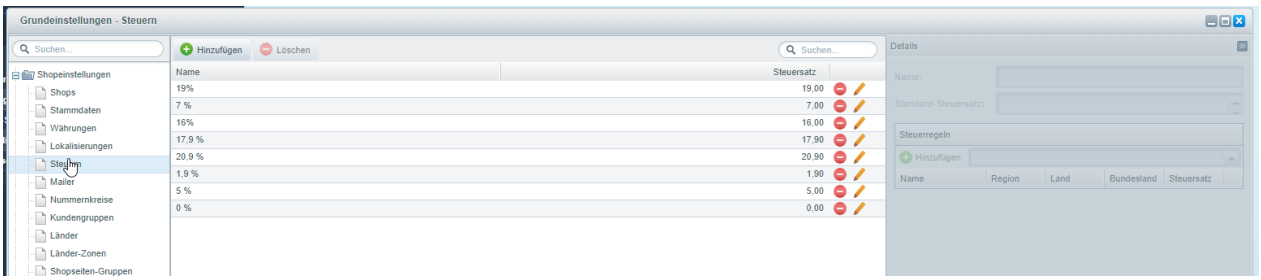
```
[WL]
webartikel = 5
```

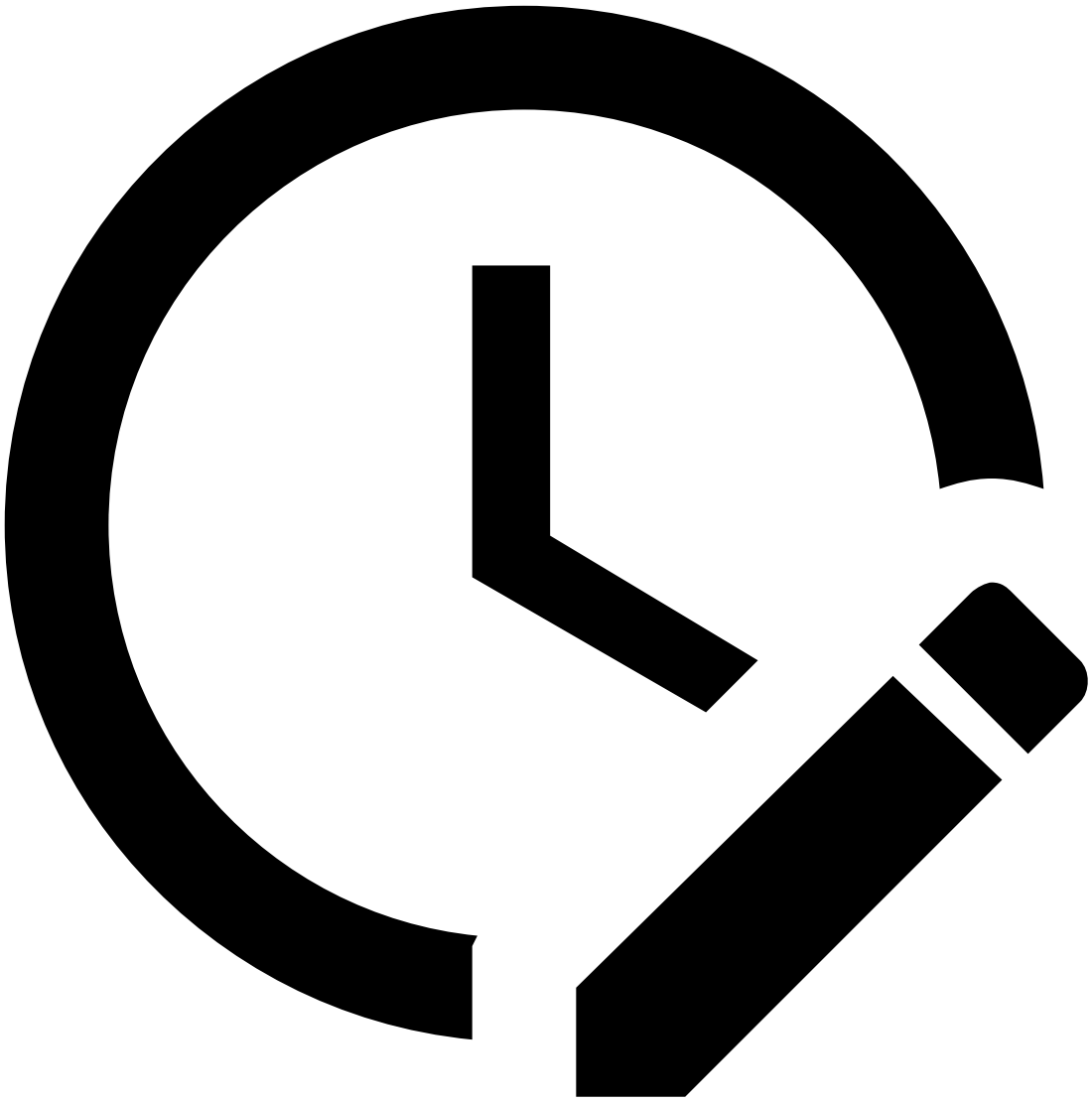
Steuersätze

Die Schnittstelle legt keine Steuersätze an. Die in der WinLine verwendeten Steuersätze müssen vorab in Shopware angelegt werden. Artikel mit nicht vorhandenen Steuersätzen werden nicht synchronisiert. Ist kein passender Steuersatz vorhanden, wird 19% als Fallback verwendet.

In Shopware 6 werden Steuersätze unter **Einstellungen > Shop > Steuern** verwaltet.

> **Screenshot aus einer älteren Shopware-Version**


Grundeinstellungen - Steuern
Suchen...
Hinzufügen
Löschen
Suchen...
Shopeinstellungen
Shops
Stammdaten
Währungen
Lokalisierungen
Steuern
Mailer
Nummernkreise
Kundengruppen
Länder
Länder-Zonen
Shopseiten-Gruppen
Name
Steuersatz
19%
19,00
7 %
7,00
16%
16,00
17,9 %
17,90
20,9 %
20,90
1,9 %
1,90
5 %
5,00
0 %
0,00
Details
Name:
Standard-Steuersatz:
Steuerregeln
Hinzufügen
Name
Region
Land
Bundesland
Steuersatz



Artikelgrafik & Downloads

Überblick

Die Schnittstelle synchronisiert Artikelbilder und Downloads aus der WinLine an Shopware. Es gibt zwei getrennte Sync-Befehle: **syncImages** für Bilder und **syncDownloads** für Downloaddateien.

Bilder werden über das WinLine Archiv II mit dem Schlagwort “Webshop Grafik” zugeordnet, Downloads über das Schlagwort “Webshop Download”. Die Bildreihenfolge im Shop wird über den Schlagwortwert gesteuert (1 = Vorschaubild, 2, 3, …).

Bilder werden vor der Übertragung automatisch auf eine konfigurierbare Maximalgröße skaliert. Nur geänderte Bilder werden bei einem erneuten Sync übertragen.

Workflow

Bilder synchronisieren

1. Schlagwörter “Webshop Grafik” und “Webshop Download” im WinLine Admin anlegen (einmalig, siehe Schlagwort anlegen)
2. Schlagwortnummer in der Config hinterlegen, wenn diese vom Standard abweicht.
3. Grafikdatei auf das Artikelfenster in der WinLine ziehen (Archiv II)
4. Schlagwort “Webshop Grafik” zuweisen und Reihenfolge setzen (1 = Vorschaubild)
5. Sync **syncImages** starten
6. Bilder im Shop prüfen — sie erscheinen in der konfigurierten Reihenfolge

Downloads synchronisieren

1. Datei über Archiv II dem Artikel zuordnen
2. Schlagwort “Webshop Download” zuweisen
3. Sync **syncDownloads** starten
4. Downloads im Shop prüfen

> **Alternativ: Artikel-Bild**
>
> Über das Feld “Artikel-Bild” in der WinLine kann auch ohne Archiv II ein einzelnes Bild pro Artikel synchronisiert werden (siehe Artikel-Bild).

Shopware

Die Grafikdateien werden die Medienverwaltung von Shopware synchronisiert. Über das Mapping kann gesteuert werden in welchen Unterordner Synchronisiert werden soll. Wird dieser Weggelassen, werden die Bilder ohne Unterordner in die Medienverwaltung geladen. Wird nachträglich ein Ordner angegeben, werden die Bilder in den entsprechenden Unterordner verschoben.

In der article6_overwrite.yml kann der Ordner folgendermaßen angegeben werden:

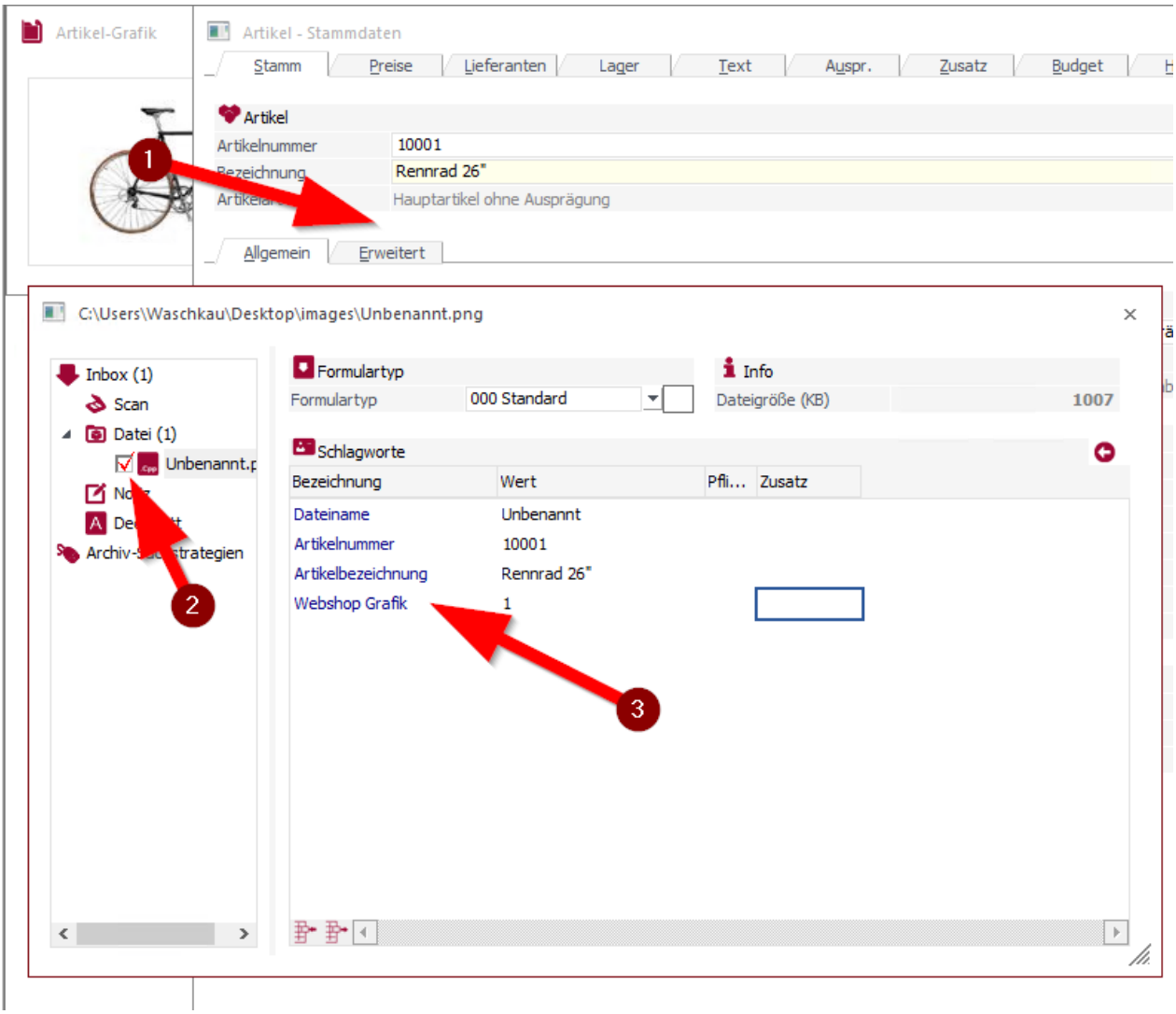
```
Artikel_Images:
  media_folder:
    shop: media.mediaFolderId
    default: 8c10d894dfd54473abb2e04bbe59fa91
```

Dies muss auf der Obersten Ebene der Yaml-Datei angegeben werden.

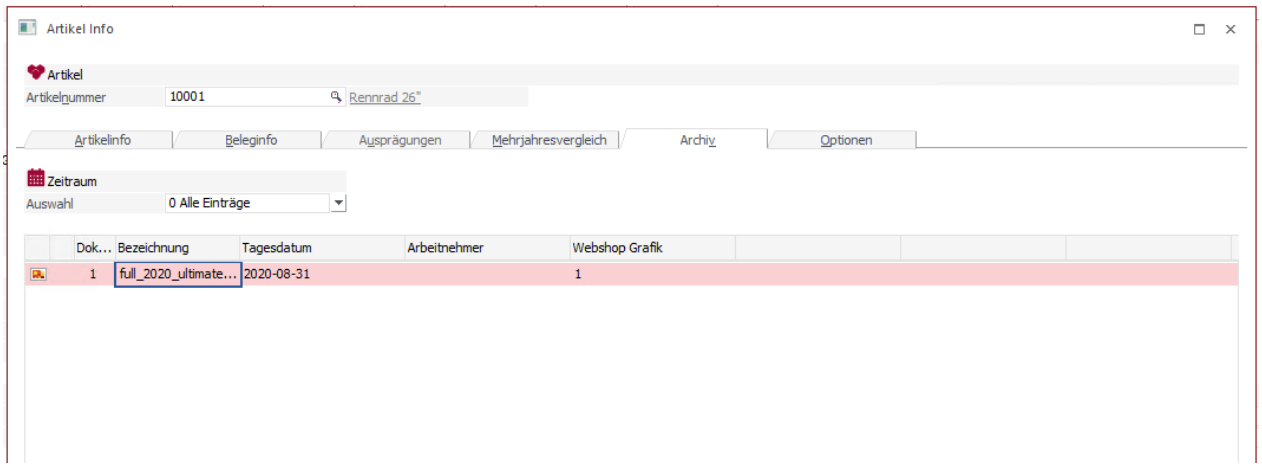
WinLine - Archiv II

Die Grafikdateien auf das Artikelfenster ziehen. Anschließend das Schlagwort “Webshop Grafik” hinzufügen. Über den Wert 1 - x wird die angezeigte Bildreihenfolge im Shop gesteuert. (1 =

Vorschaubild) Bei der nächsten Synchronisation werden die Artikelgrafiken übertragen.



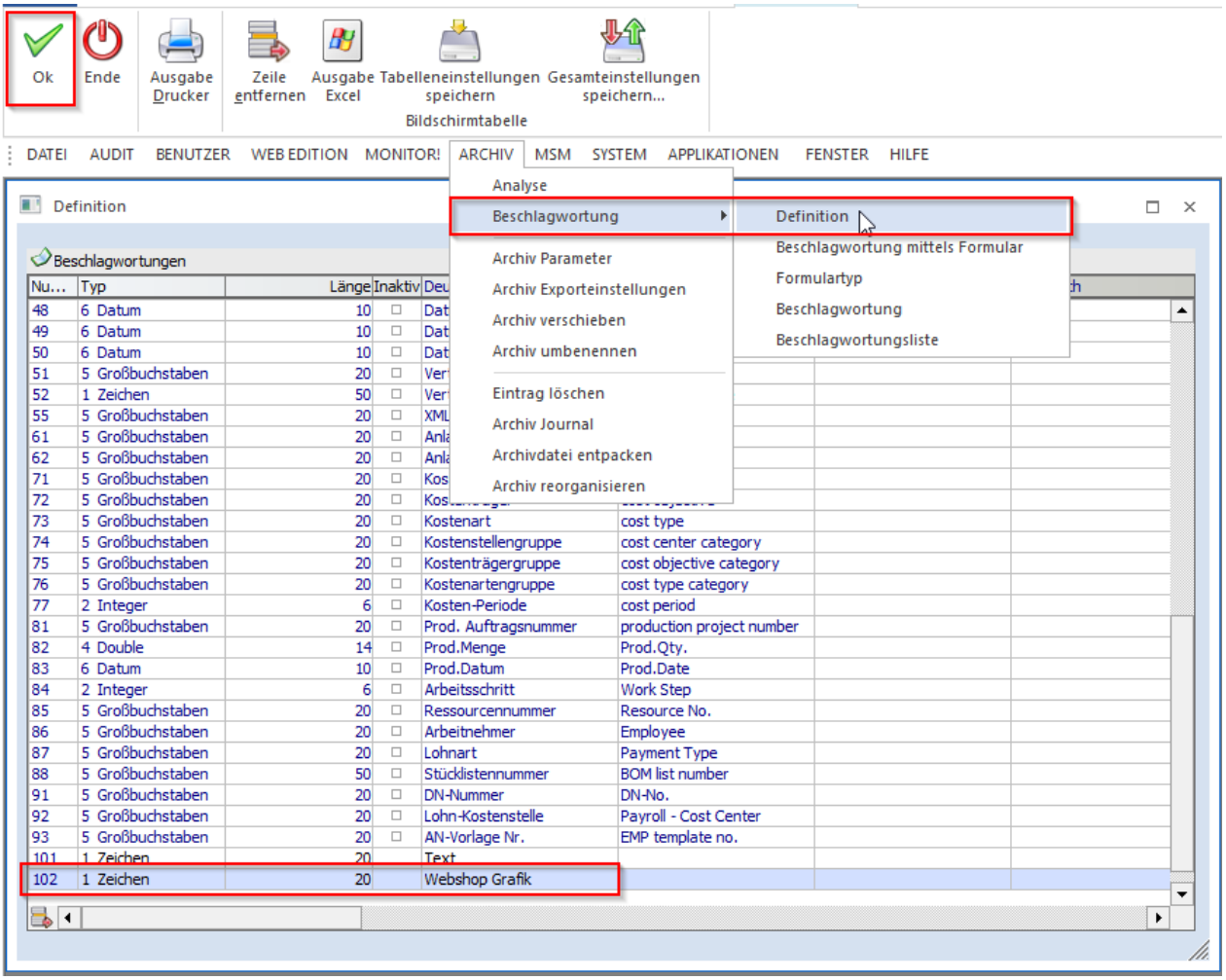
Über den Menüpunkt “Artikelinfo” “Archiv” können alle dem Artikel zugeordnete Grafiken aufgerufen, hinzugefügt und gelöscht werden.



WL Schlagwort anlegen

Bevor das Schlagwort “Webshop Grafik” hinzugefügt werden kann, muss dieses im WinLine Admin Bereich angelegt werden.

WinLine Admin aufrufen



Die Schlagworte werden über den WL-Adminbereich >Archiv >Beschlagwortung >Definition angelegt 1. Schlagwort: Webshop Grafik 2. Schlagwort: Webshop Download

Entsprechend der jeweiligen Bezifferung werden diese im Shop angezeigt:

Schnittstelle

WinLine Schlagwort IDs:

```
[ARCHIV]
; interne WinLine ID (Archiv) für das Schlagwort: Webshop Grafik
archiv_webshop_grafik = 102
; interne WinLine ID (Archiv) für das Schlagwort: Webdokument
archiv_webdokument = 103
; interne WinLine ID (Archiv) für das Schlagwort: Artikelnummer
archiv_artikelnummer = 41
; interne WinLine ID (Archiv) für das Schlagwort: Artikelbezeichnung/Beschreibung
archiv_beschreibung = 42
;Liste der Zusatzfelder für Archiv 1 Grafiken (kommagetrennt)
zusatzgrafiken =
```

Maximale Größe setzen

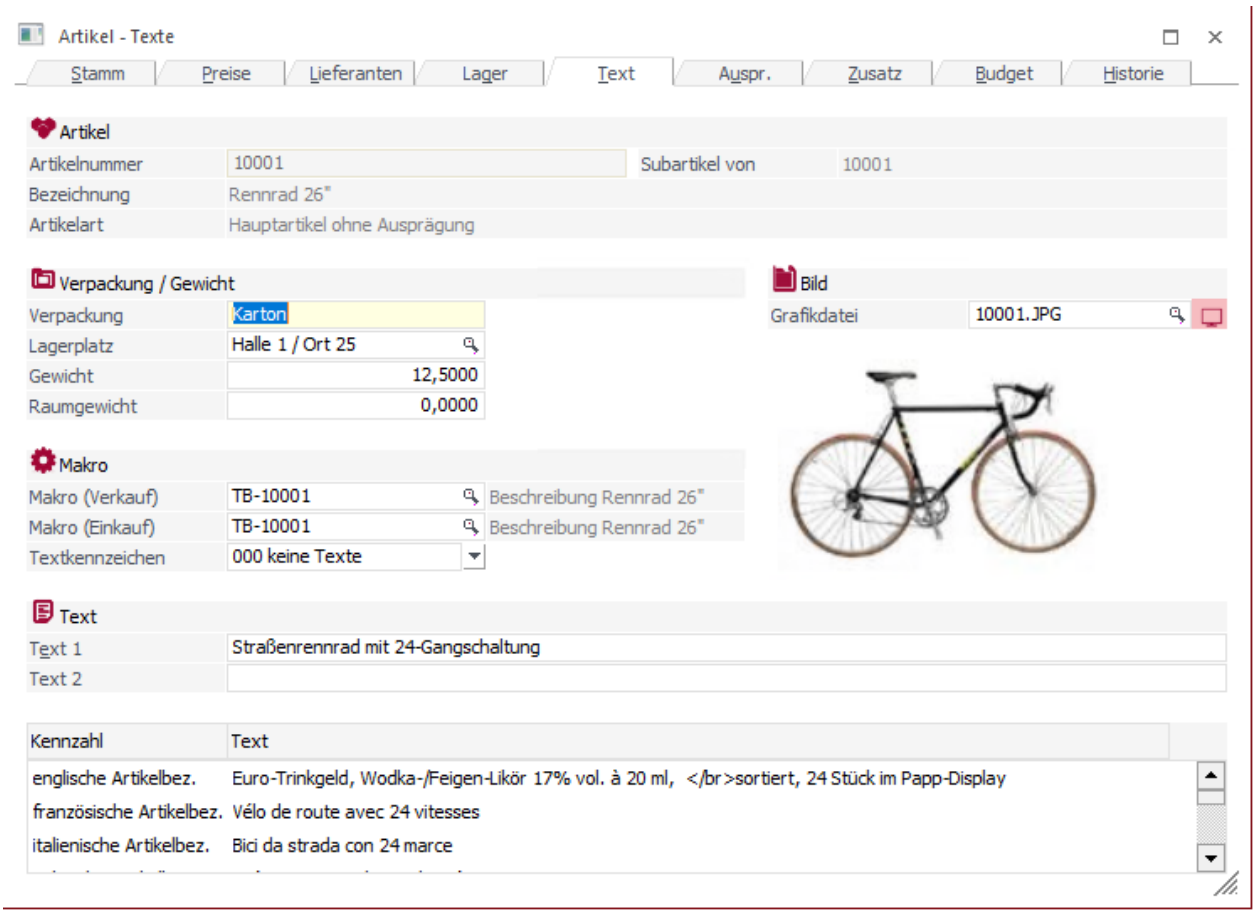
Werden in der WinLine hochauflösende Bildformate verwendet, dann ist es sinvoll die Bilder vor der Übertragung durch die Schnittstelle zu optimieren. In der Standardeinstellung skaliert die Schnittstelle automatisch auf eine sinnvolle Größe. Die maximale Größe für Bilder kann aber auch manuell in der Konfiguration gesetzt werden.

config.ini

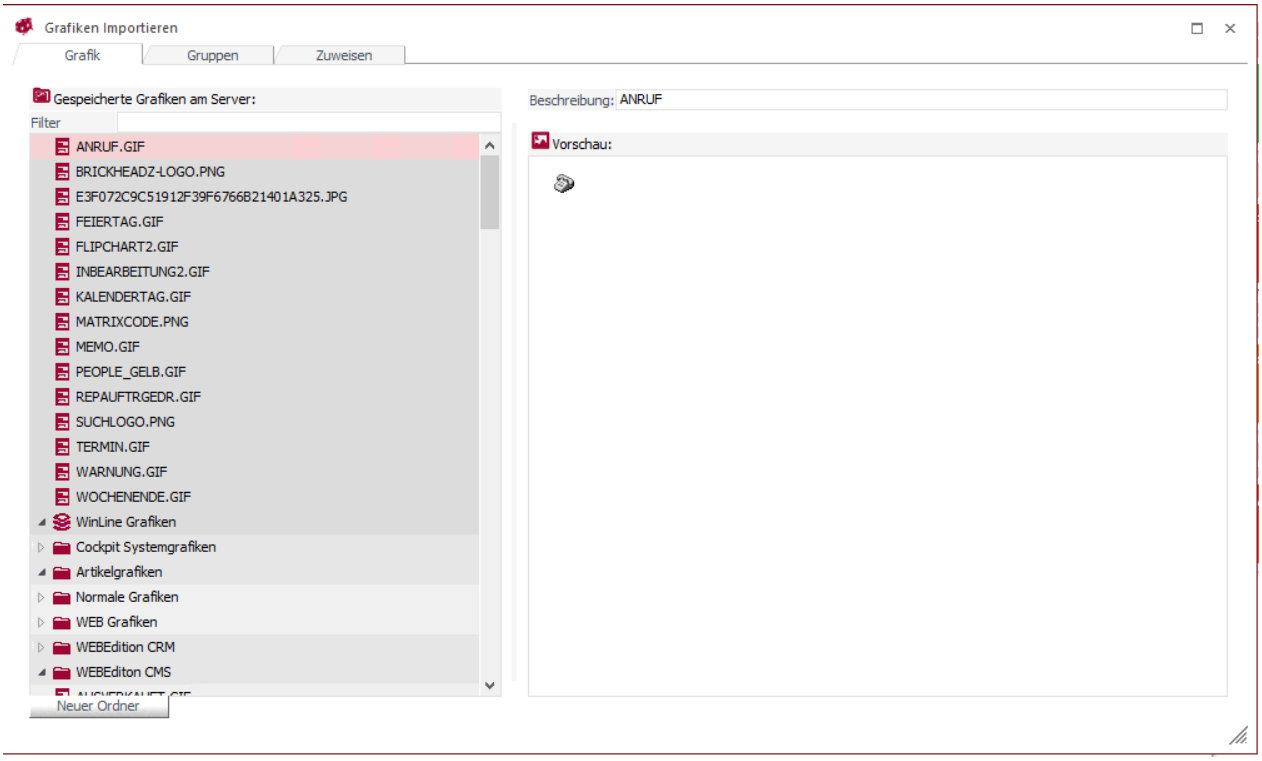
```
[CONVERT]
scale_images = 1280,720
```

WinLine - Artikel Bild

Alternativ ist es auch möglich das Feld Artikel-Bild der WinLine zu verwenden. Auf diese Weise kann nur eine Bilddatei pro Artikel synchronisiert werden. Grafikdateien über “Datei” “Grafiken importieren” Dateinamen im Artikel eintragen. Anschließend werden die Artikelbilder synchronisiert.



Über den Menüpunkt “Grafiken importieren” die Artikelbilder einspielen. Ein Massenimport ist über Drag and Drop möglich.



Bilder löschen

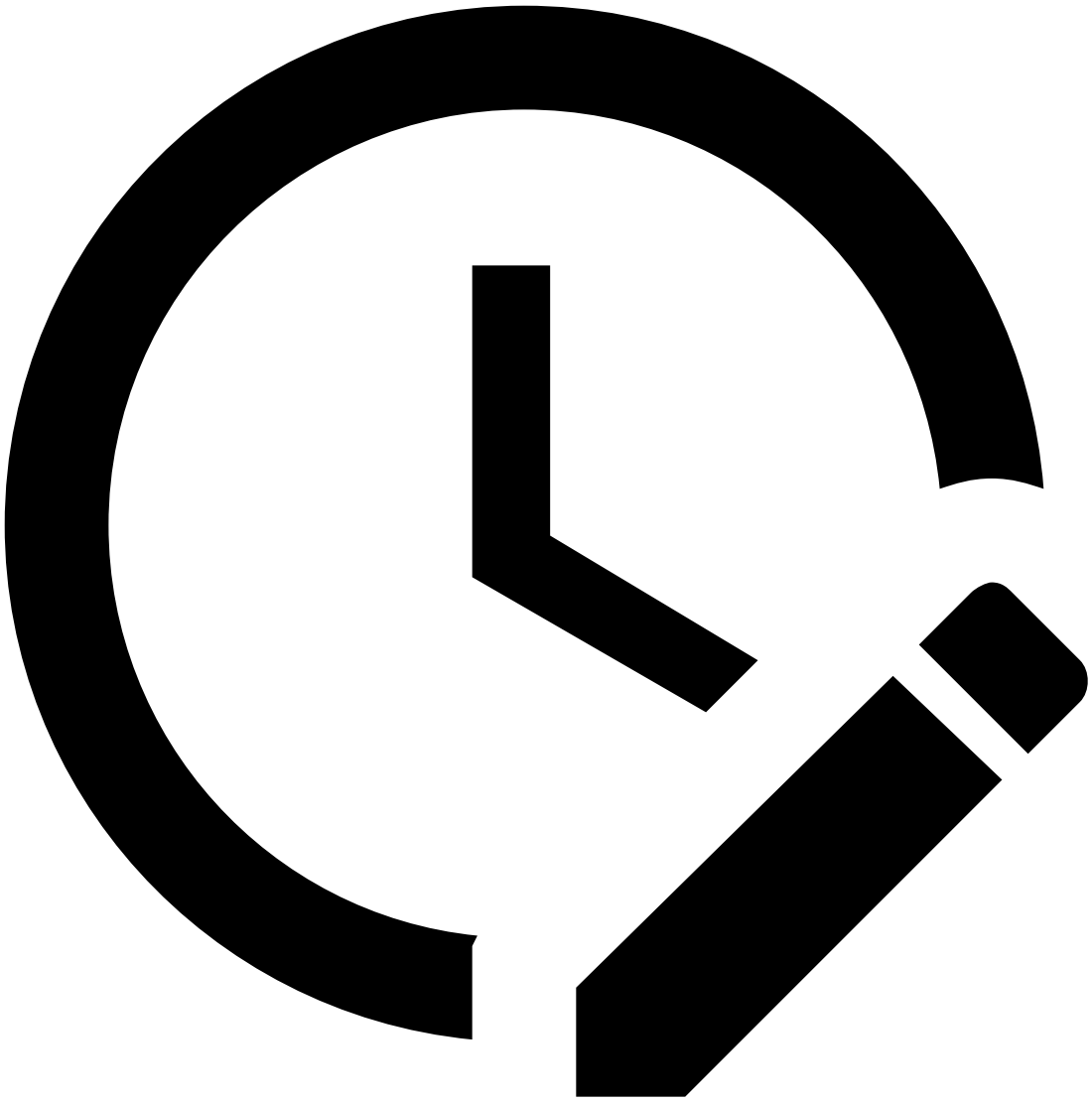
Standardmäßig bleiben einmal übertragene Bilder und Downloads in Shopware bestehen, auch wenn das Schlagwort “Webshop Grafik” bzw. “Webshop Download” in der WinLine später entfernt wird.

Mit dem Feature-Flag `REMOVE_ORPHAN_MEDIA` werden beim Sync alle `product_media`-Einträge entfernt, die für die synchronisierten Artikel (inkl. Varianten) gelten, aber in der aktuellen WinLine-Ergebnismenge nicht mehr enthalten sind. Damit führt das Entfernen des Schlagworts in der WinLine direkt zum Entfernen des Bildes/Downloads in Shopware 6.

```
[FEATURES]
REMOVE_ORPHAN_MEDIA = true
```

Sicherheitsnetz

Liefert die WinLine für **alle** synchronisierten Artikel 0 Medien (z.B. wegen DB-Fehler), wird der Cleanup übersprungen und eine Warnung ins Log geschrieben, damit nicht versehentlich alle Bilder im Shop entfernt werden.



2026-04-20

Cross-Selling

Modul: Crossselling

Um Crossselling an Shopware zu übermitteln wird das kostenpflichtige Modul Crossselling benötigt.

Überblick

Cross-Selling ermöglicht die Anzeige von Zubehör- und Ähnlichen Artikeln auf der Produktdetailseite in Shopware. Die Synchronisation erfolgt in Richtung **WinLine → Shopware**.

Es stehen zwei Varianten zur Verfügung:

1. **Manuelles Crossselling** über Ersatzartikel: Jeder Artikel erhält einen Crossselling-Container-Artikel mit zugewiesenen Ersatzartikeln. Die Priorität bestimmt den Typ (1–99: Ähnliche Artikel, 100–199: Zubehör, 200–299: Beide).
2. **Dynamisches Crossselling** über Zusatzfelder: Artikel werden automatisch anhand von Produktgruppen zugeordnet. Vereinfachte Pflege ohne manuelle Zuweisung pro Artikel.

Workflow

Manuelles Crossselling

1. Einen Artikel „Crossselling" in der WinLine anlegen (einmalig, dient als Container)
2. Den Crossselling-Artikel als Ersatzartikel dem gewünschten Produkt zuweisen
3. Im Crossselling-Artikel die gewünschten Crossselling-Produkte als Ersatzartikel hinzufügen
4. Priorität setzen (bestimmt den Crossselling-Typ)
5. Sync **syncCross** (Cross-Artikel) starten
6. Ergebnis auf der Produktdetailseite im Shop prüfen

Dynamisches Crossselling

1. Feature-Flag `DYNAMIC_CROSS_SELLING = True` in der `config.ini` setzen
2. Zusatzfelder für Gruppenzugehörigkeit und Anzeige konfigurieren
3. Artikel in der WinLine den gewünschten Gruppen zuweisen
4. Sync starten
5. Ergebnis im Shop prüfen

Cross-Selling Anzeige in Shopware

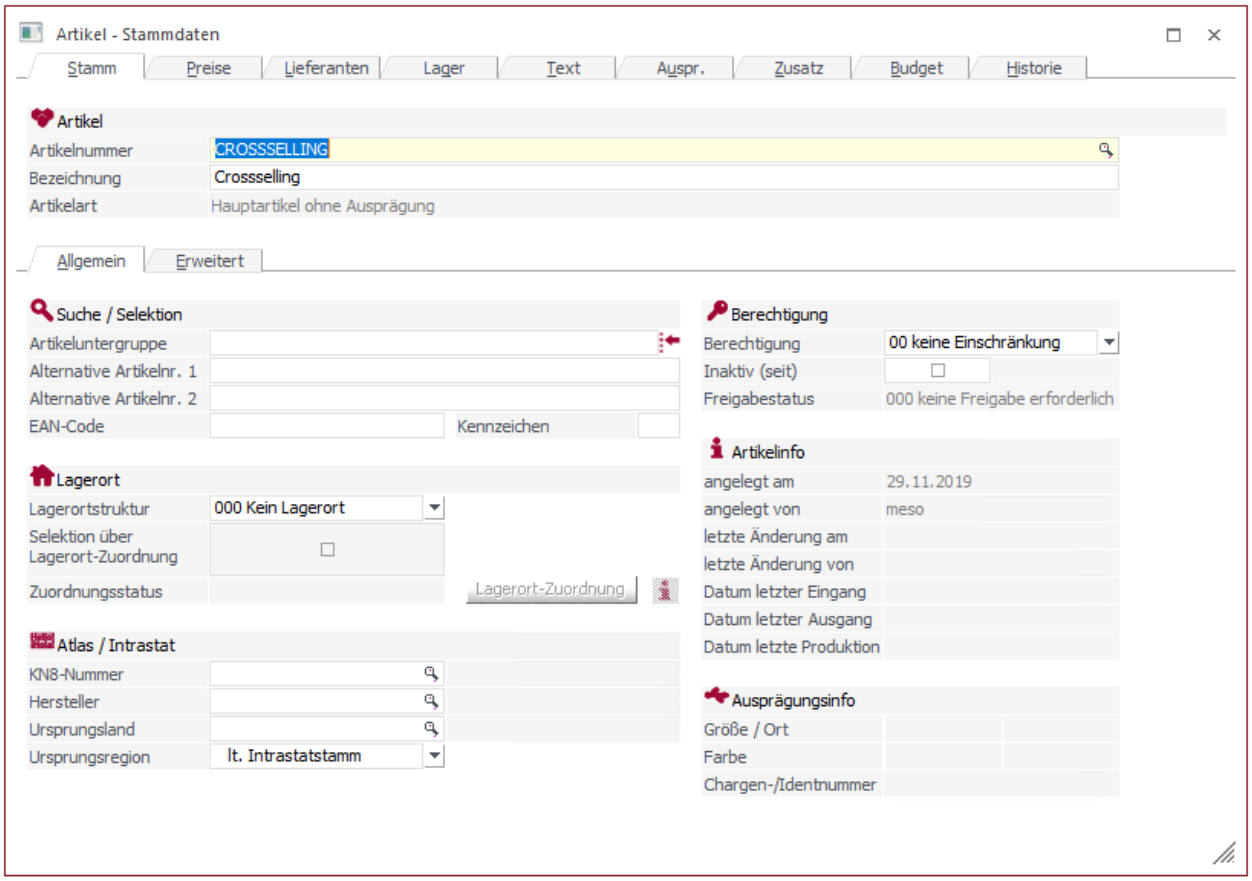
Für den Bereich Cross-Selling können über die Schnittstelle Zubehör-und Ähnliche-Artikel zugewiesen werden. Diese tauchen jeweils direkt unter dem Artikel auf.

Cross-Selling per Ähnliche Artikel

Im Block [CROSS] kann mit dem Wert “crossselling_aktiv=True/False” diese Funktion ein- und ausgeschaltet werden.

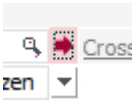
Zur Einrichtung in der WinLine

Einmalig muss ein Artikel „Crossselling“ angelegt werden.



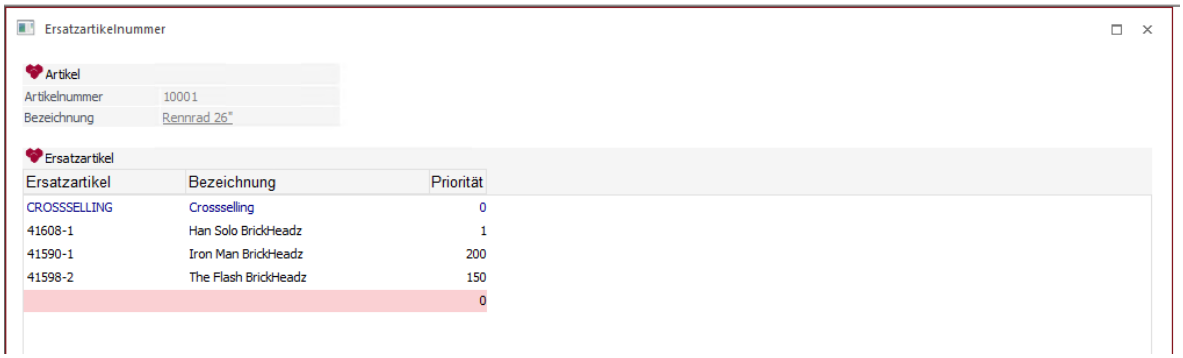
Den Crossselling-Artikel als Ersatzartikel zuweisen

Erklärung: Der Crossselling-Artikel dient lediglich als Container für die jeweils gewünschten Crossselling-Artikel



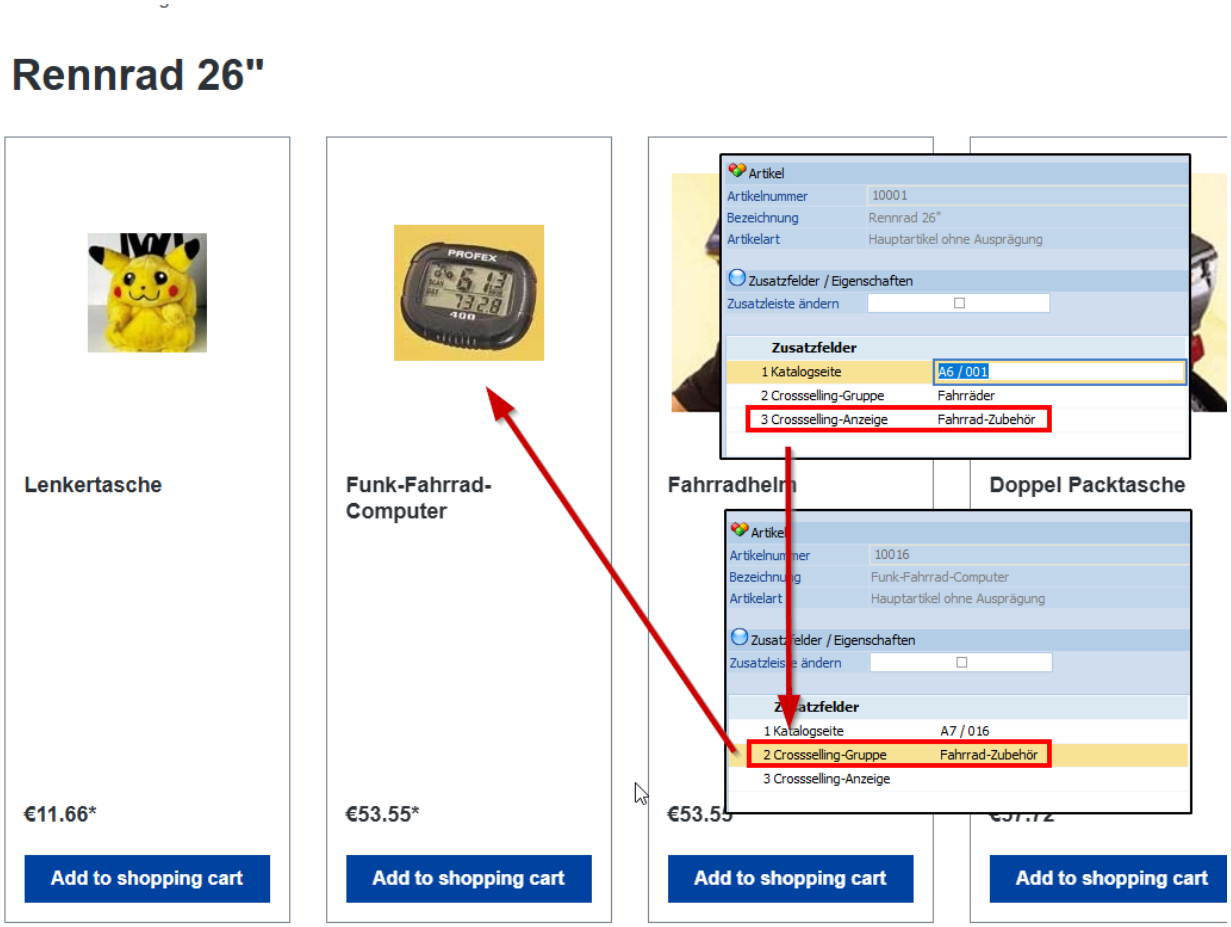
Prioritäten–Steuerung „Ähnliche Artikel“ und Zubehör-Artikel“

- Priorität 1 - 99 = Ähnliche Artikel
- Priorität 100 - 199 = Zubehör-Artikel
- Priorität 200 - 299 = Ähnliche & Zubehör-Artikel



Dynamisches Crossselling / Zusatzfelder

Es ist auch möglich, Zusatzfelder in der Winline zu definieren, die Produkten die Möglichkeit gibt, Teil einer dynamischen Produktgruppe zu sein und welche Produktgruppe bei dem jeweiligen Artikel angezeigt wird. Der Vorteil ist eine vereinfachte Pflege der Stammdaten, da nicht jeder Artikel eine manuelle Zuweisung der Cross-Sellingartikel benötigt.

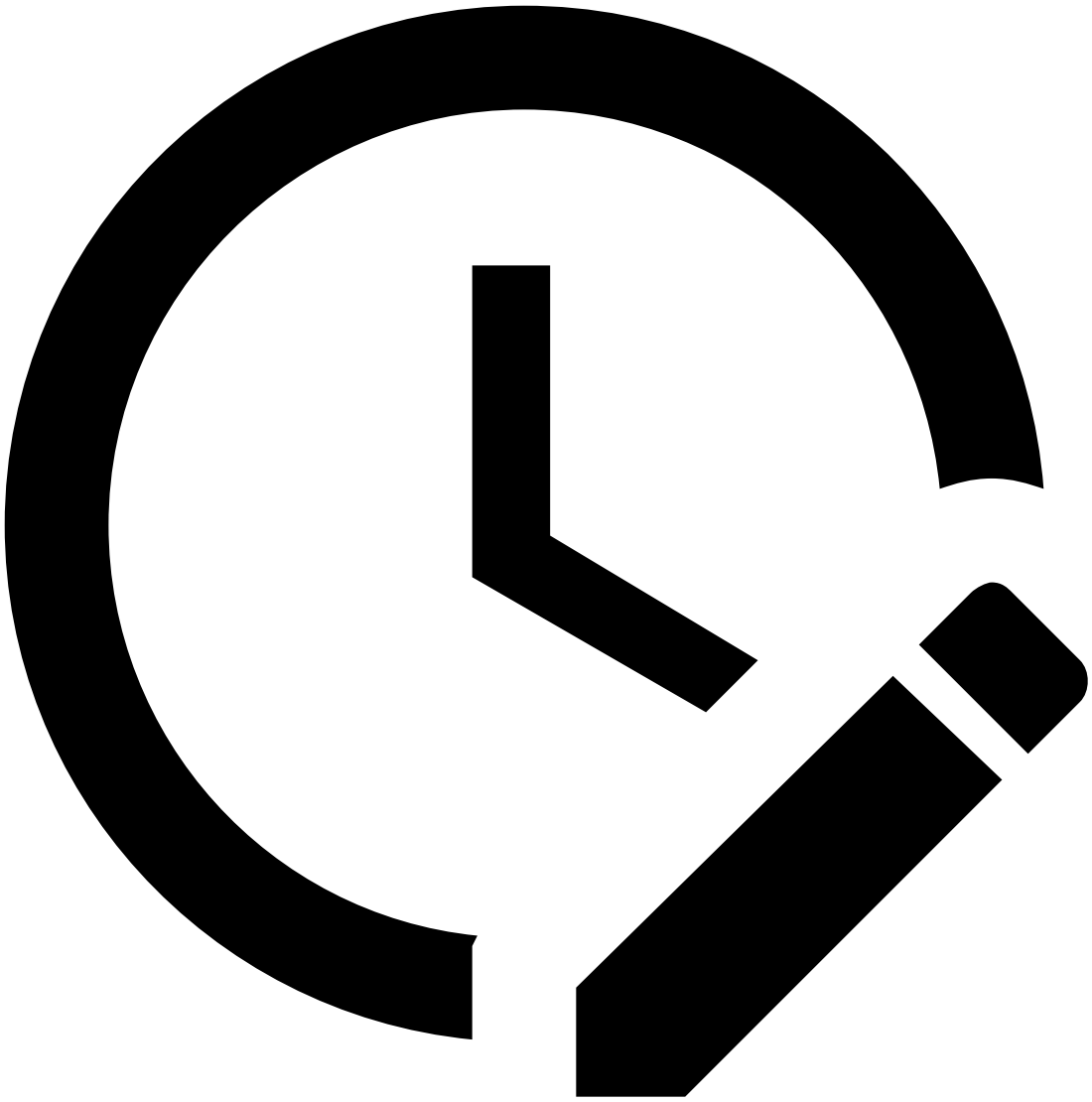


In der config.ini kann das Feature DYNAMIC_CROSS_SELLING aktiviert und mit den Werten in dem Block [DYNACROSS] konfiguriert werden.

> **Feature-Flag**
>
> Das Feature ist standardmäßig **deaktiviert** (`DYNAMIC_CROSS_SELLING = False` in `[FEATURES]`) und muss explizit aktiviert werden.

Die Standardwerte sind hier:

```
cross_name = Unsere Vorschläge
cross_gruppe_feld = C229_Zusatzfeld29
cross_anzeige_feld = C230_Zusatzfeld30
sort_by = cheapestPrice (mögliche Werte: cheapestPrice / name / releaseDate)
sort_direction = ASC (mögliche Werte: ASC / DESC)
```



2026-03-25

Lieferzeiten + Disposition

Modul: Lieferzeiten + Disposition

Für die Verwendung der hier beschriebenen Funktionen wird das Modul Lieferzeiten + Disposition benötigt.

Überblick

- Das Modul synchronisiert verfügbare Lagerbestände und Lieferzeiten von der WinLine an Shopware.
- Der Bestand wird unter Berücksichtigung der Artikelbedarfsvorschau berechnet.
- Bei einer Verfügbarkeit von 0 wird die Lieferzeit automatisch aus den Wiederbeschaffungstagen (WBT) des A-Lieferanten und den Puffertagen des Artikels errechnet.
- Bestellungen nicht verfügbarer Ware werden automatisch für den nächsten Lieferzeitpunkt eingeplant.

Workflow

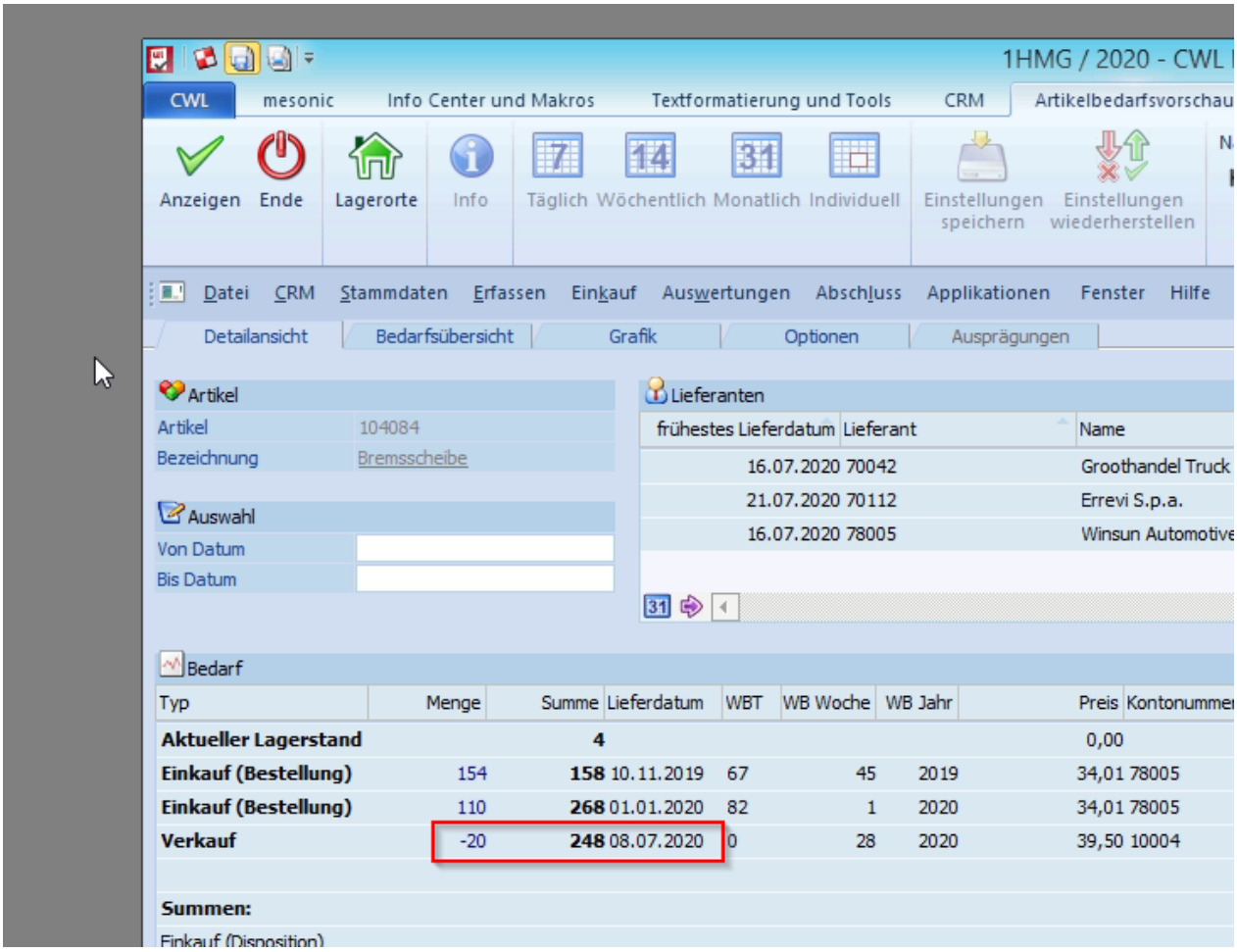
1. Wiederbeschaffungstage des A-Lieferanten in der WinLine hinterlegen.
2. Puffertage im Artikelstamm pflegen.
3. Sync **syncBestand** (Bestand) starten.
4. Die Schnittstelle berechnet den verfügbaren Bestand (Lagerbestand minus offene Aufträge).
5. Ist der Bestand > 0: Artikel wird als verfügbar angezeigt.
6. Ist der Bestand = 0: Lieferzeit wird aus WBT + Puffertage berechnet und im Shop angezeigt.
7. Ergebnis im Shop prüfen (verfügbare Menge und Lieferzeit).

Sync-Befehl

Der Bestandssync wird über den Befehl **syncBestand** im Management Interface ausgeführt.

WinLine verfügbarer Lagerbestand

Artikelbedarfsvorschau



WinLine

In der WinLine sind die Lieferzeiten des A-Lieferanten und die Puffertage im Artikel zu hinterlegen.

Schnittstelle

In der Schnittstelle ist keine weitere Konfiguration erforderlich.

Mapping-Konfiguration

Über das Mapping kann gesteuert werden, ob Bestände mit oder ohne Berücksichtigung der Handelsstücklisten übermittelt werden. Die Lieferzeit kann über `eval`-Ausdrücke im Mapping individuell angepasst werden (siehe Beispiele unten).

Bestände mit und ohne Handelsstücklisten

Schnittstelle

Mapping

Wenn die Bestände ohne Berücksichtigung der Bestände in der Handeslstückliste übermittelt werden sollen, muss 1 Feld im Mapping angepasst werden. Je nachdem ob das Dispositions-Modul verwendet wird handelt es sich um ein anderes Feld.

1. Ohne Dispositions Modul: * Mit Handeslstückliste: `lagerstand_gesamt` * Ohne Handelsstückliste: `lagerstand`

Beispiel 1: Wiederbeschaffungstage werden direkt als Lieferzeit übergeben

```
"Bestand":{
    "stock": {
        "winline": "lagerstand_gesamt",
        "shop": "inStock"
    },
    "time": {
        "winline": "wbt",
        "shop": "shippingTime",
        "env": ["from decimal import Decimal"],
        "eval": "Decimal(wl_data)"
    }
},
```

Beispiel 2: Rechne extra 2 Tage Lieferzeit für Shopbestellungen auf die WBT

```
"Bestand":{
    "stock": {
        "winline": "lagerstand_gesamt",
        "shop": "inStock"
    },
    "time": {
        "winline": "wbt",
        "shop": "shippingTime",
        "env": ["from decimal import Decimal"],
        "eval": "Decimal(wl_data) + Decimal('2')"
    }
},
```

Beispiel 3: Fixe Lieferzeit von 5 Tagen wenn die WBT 3 oder niedriger ist, ansonsten ist der Artikel nicht lieferbar (shippingTime = 0)

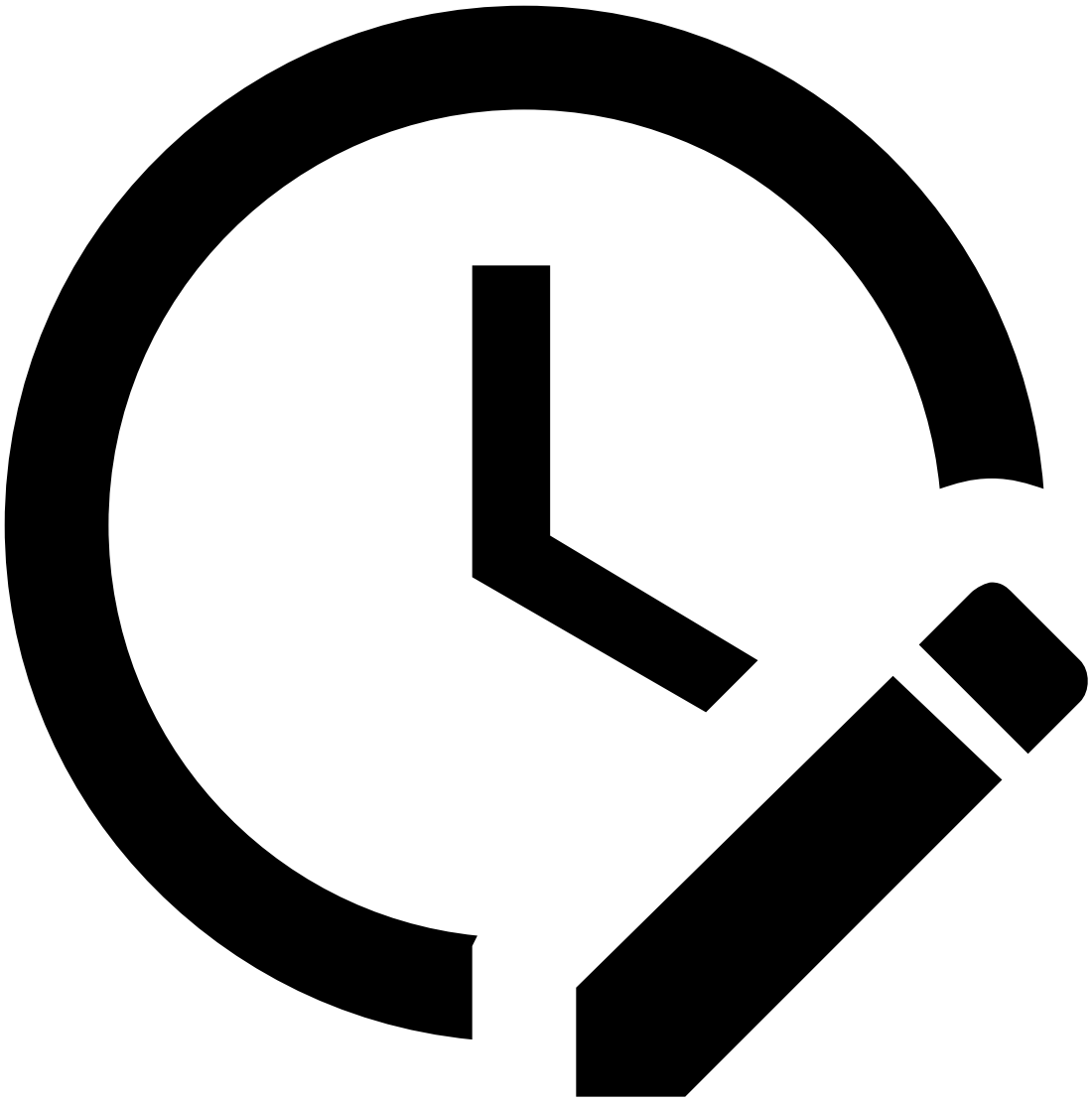
```
"Bestand":{
    "stock": {
        "winline": "lagerstand_gesamt",
        "shop": "inStock"
    },
    "time": {
        "winline": "wbt",
        "shop": "shippingTime",
        "env": ["from decimal import Decimal"],
        "eval": "5 if Decimal(wl_data) <= Decimal('3') else 0"
    }
},
```

1. Mit Dispositions Modul: * Mit Handeslstückliste: `verfuegbarer_lagerstand_gesamt`
   * Ohne Handelsstückliste: `verfuegbarer_lagerstand`

Beispiel:

```
"verfuegbarer_Bestand":{
    "stock": {
        "winline": "verfuegbarer_lagerstand_gesamt",
        "shop": "inStock"
    },
    "time": {
        "winline": "wbt",
        "shop": "shippingTime",
        "env": ["from decimal import Decimal"],
        "eval": "5 if Decimal(wl_data) <= Decimal('3') else 0"
    }
}
```

NOTE: Für mehr Beispiele bezüglich der Funktionalitäten mit dem WBT siehe 1. Ohne Dispositions Modul



2026-03-25

Eigenschaften

Modul: Artikeleigenschaften

Um Eigenschaften in Shopware filterbar oder auf Produktseite anzeigbar zu machen, wird das kostenpflichtige Modul Artikeleigenschaften benötigt.

Überblick

Die Schnittstelle überträgt Artikeleigenschaften aus der WinLine an Shopware. Eigenschaften können im Shop als Filter in Produktlisten sowie auf der Produktdetailseite angezeigt werden.

Folgende Darstellungsarten werden unterstützt:

- **Text** – Darstellung der Ausprägung als Text
- **Bild** – Darstellung der Ausprägung anhand von Bildern
- **Dropdown** – Darstellung der Ausprägung als Dropdown
- **Farbe** – Darstellung der Ausprägung anhand von Farbkacheln

Im Standard werden alle Eigenschaften eines Webartikels an den Shop übertragen, mit der Einstellung **nicht auf Produktseite anzeigen** und **nicht filterbar**. Über ein Konfigurationsfeld im Eigenschaftenstamm der WinLine kann die Darstellung und Sichtbarkeit gezielt gesteuert werden.

Workflow

Der typische Ablauf zur Einrichtung und Übertragung von Eigenschaften ist wie folgt:

1. **Eigenschaft anlegen** – Im Eigenschaftenstamm der WinLine eine neue Eigenschaft anlegen (unterstützte Typen: 2, 4, 7, 8 oder M).
2. **Darstellung konfigurieren** – Im Übersetzungsfeld (Standard: 14. Feld) die gewünschte Darstellung eintragen, z.B. `Text -nF` für eine Texteigenschaft die nicht filterbar ist.
3. **Eigenschaft zuweisen** – Die Eigenschaft dem gewünschten Artikel zuweisen.
4. **Sync starten** – Den Artikelsync oder Eigenschaftensync ausführen.
5. **Ergebnis prüfen** – Im Shop kontrollieren, ob die Eigenschaft auf der Produktseite und ggf. im Filter korrekt angezeigt wird.

Sync-Befehl

Der Eigenschaftensync wird beim regulären Artikelsync mit übertragen, sofern dies nicht per `-we` deaktiviert wurde.

WinLine

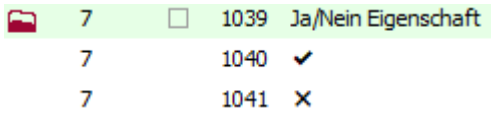
Einrichtung

Eigenschaftentypen

Von der Schnittstelle werden die Eigenschaftentypen 2, 4, 7, 8 und M unterstützt.

Checkbox

Checkbox-Eigenschaften werden nicht unterstützt. Stattdessen kann aber eine Eigenschaft des Typ 7 mit den Unicode-Zeichen ✔ (U+2714) und ✘ (U+2716) angelegt werden.



Um Eigenschaften verwenden zu können, müssen diese als erstes über den Eingeschaftenstamm der WinLine angelegt werden. Dier Eigenschaftenstamm findet sich in der Start-Ansicht unter dem Menüpunkt `Optionen -> Eigenschaften`. Im Standard wird jede Eigenschaft die einem WebArtikel zugeordnet ist an den Shop übertragen, mit den Einstellungen nicht auf Produktseite anzeigen und nicht Filterbar.

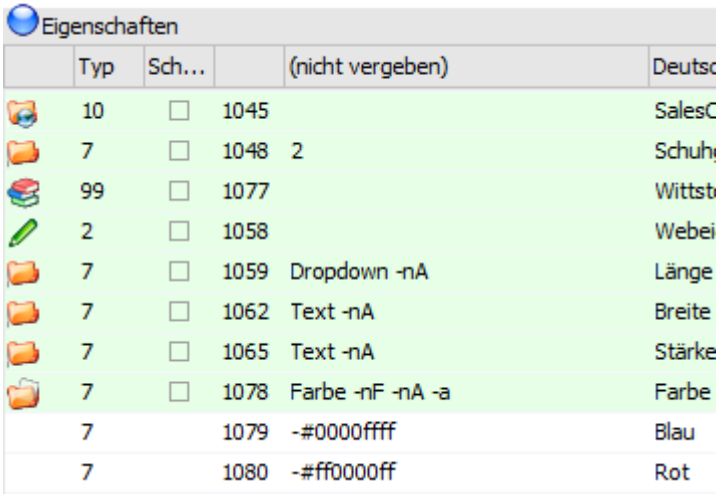
Die an Shopware zu übertragenen Eigenschaften der WinLine können direkt in der WinLine Konfiguriert werden. Shopware 6 kann diese für die Varianten verwenden. Hierzu wird ein Modul/Plugin benötigt. Die Konfiguration wird über eines der Übersetzungsfelder der Eigenschaften getätigt. Im Standard wird das Feld “(nicht vergeben)” bzw. das 14te Feld verwendet.

Mögliche Parameter sind wie folgt:

```
Eigenschaften Text parser
positional arguments:
  {Text,Bild,Dropdown,Farbe,web,1}
                        Text - Darstellung der Auspraegung als Text
                        Bild - Darstellung der Auspraegung anhand von Bildern
                        Dropdown - Darstellung der Auspraegung als Dropdown
                        Farbe - Darstellung der Auspraegung anhand von Farbkachel
                        web/1 - Eigenschaft soll an den Shop übermittelt werden
optional arguments:
  -h, --help            show this help message and exit
  -nF, --nichtFilterbar
                        Im Produktfilter von Produktlisten NICHT anzeigen
                        (Default: True)
  -nA, --nichtAnzeigen  Auf der Produktdetailseit NICHT anzeigen (Default:
```

```
                        True)
  -a, --alphanumerisch  Sortierung (Default: position)
  -# HEX, --Hex HEX     Farbe in Hex notation für die Auswahl nach Farbe.
                        (Default: None)
```

Beispiel:



Verwendung

Wird eine Shopeigenschaft für einen Artikel gesetzt, dann wird diese automatisch beim nächsten Übertragungslauf der Schnittstelle an den Shop übermittelt und ggf. neu angelegt.

Shop

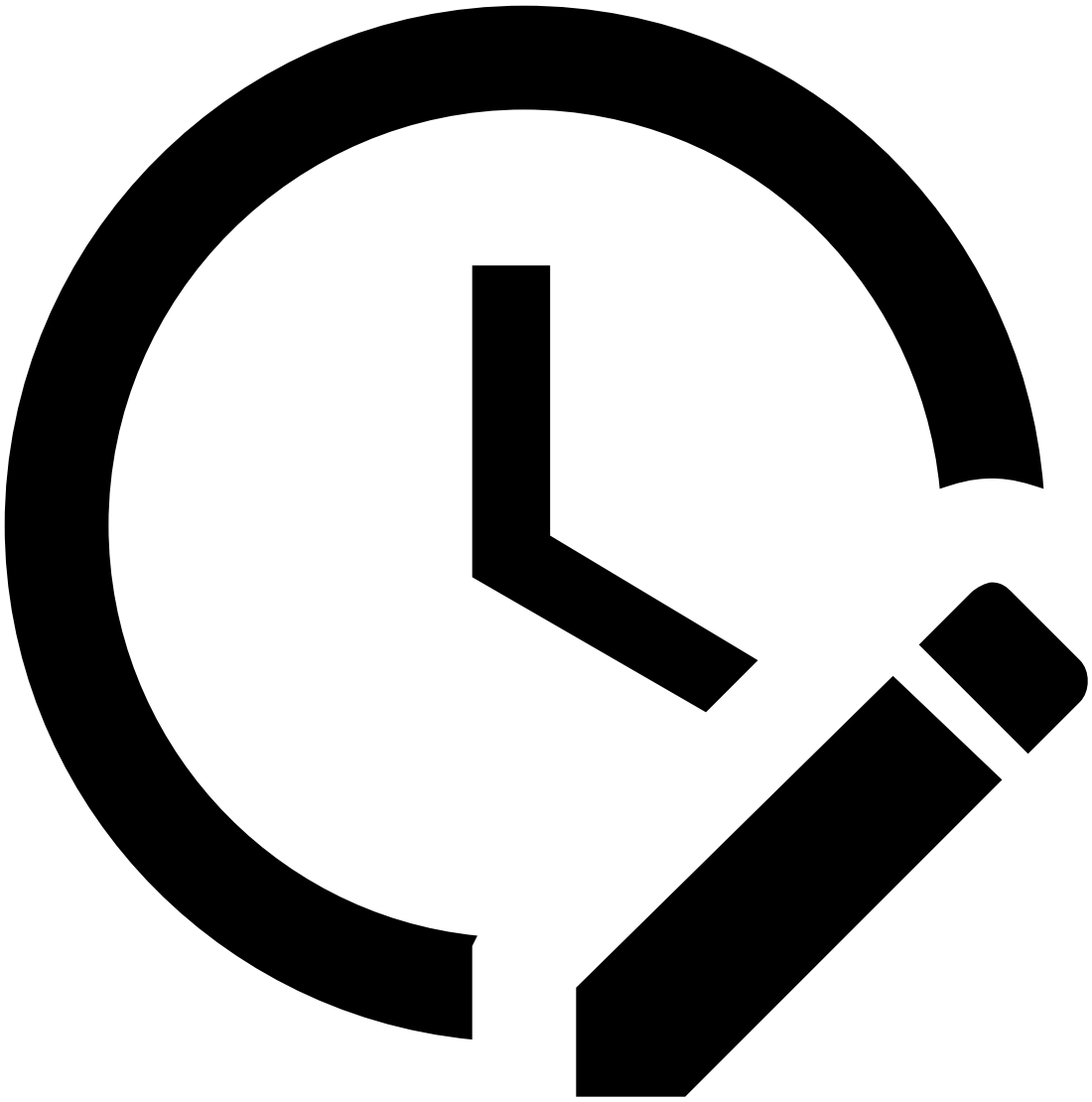
Einrichtung

Im Shop ist keine gesonderte Einrichtung notwendig. Die Reihenfolge der Eigenschaften wird von der Reihenfolge in den WinLine-Eigenschaften bestimmt.

Schnittstelle

Einrichtung

Wenn eine abweichende Spalte im Eigenschaftenstamm zur Festlegung der Übertragung an Shopware verwendet werden soll, kann diese in den Einstellungen der Shopschnittstelle hinterlegt werden.



2026-03-26

Feldzuordnung

Änderungen im Mapping

Werden Zuordnungen im Mapping angepasst ist in jedem Fall eine komplette Synchronisation des geänderten Objekttyps notwendig. Wird z.B. das Mapping von Artikeln angepasst, so ist eine Komplettübertragung aller Artikel vorzunehmen.

Überblick

- Mapping-Dateien definieren die Zuordnung zwischen WinLine-Feldern und Shopware-Feldern
- Für jeden Datentyp (Artikel, Bestellungen, Kunden) existieren eigene Mapping-Dateien
- Die Dateien unterstützen sowohl JSON- als auch YAML-Format
- Über die `_overwrite`-Dateien (z.B. `article6_overwrite.yml`) können kundenspezifische Anpassungen vorgenommen werden, ohne die Standard-Mapping-Dateien zu verändern
- Im Mapping können Werte direkt übernommen, umgerechnet oder über Umsetzungstabellen transformiert werden

Workflow

Mapping anpassen:

1. Die passende Mapping-Datei identifizieren (z.B. `article6_overwrite.yml` für Artikel)
2. Gewünschte Feldzuordnung ergänzen oder anpassen
3. Bei Bedarf eine Umsetzungstabelle im Unterordner `Umsetzungstabellen/` anlegen
4. Komplettsync für den geänderten Datentyp durchführen (z.B. alle Artikel neu synchronisieren)
5. Ergebnis im Shop oder in der WinLine prüfen

Komplettsync erforderlich

Nach jeder Änderung am Mapping ist ein Komplettsync des betroffenen Objekttyps notwendig.

Mit Hilfe der Mapping-Dateien kann die Zuordnung zwischen WinLine- und Shopware-Feldern konfiguriert werden. Für Shopware 6 werden folgende Mapping-Dateien verwendet:

- `article6.yml` / `article6_overwrite.yml` - Artikel-Mapping
- `order6.yml` / `order6.json` - Bestell-Mapping
- `customer6.yml` - Kunden-Mapping
- `customer6_kontakt.yml` - Kontakte-Mapping

- `customer6_kontakt_exisiting.yml` - Bestehende Kontakte-Mapping
- `index.json` - Index-Datei für Mapping-Zuordnung

Die Mapping-Dateien unterstützen sowohl das JSON- als auch das YAML-Format.

Einfachste Beispiele wäre eine Direktübertragung eines Feldes der WinLine in eines im Shop (siehe EAN).
Des Weiteren ist naütlrich auch eine 1/0 Übertragung möglich, um bspw. einen aktiv/inaktiv im Shop zu generieren.

```
"Variant":{
    "EAN":{
        "shop": "ean",
        "winline": "C075_EANCode"
    },
    "ACTIVE":{
        "shop": "active",
        "winline": "C038_Inaktiv",
        "eval": "False if wl_data else True"
    }
},
```

Umsetzungstabellen

Soll ein Wert im Mapping zunächst umgesetzt werden, wird eine Umsetzungstabelle im JSON format benötigt, diese muss in dem Unterordner `/Umsetzungstabellen/<umsetzungsname>.json` angelegt werden. Dieser Unterordner muss sich in dem Ordner des Mappings befinden. Im Mapping kann über den Aufruf `self.umsetzungstabelle('<umsetzungsname>', wl_data)` ein Wert transformiert werden. Der `<umsetzungsname>` ist der name der Umsetzungstabelle.

In der Umsetzungstabelle wird ein Wildcard Zeichen (Default:*) unterstützt und wenn kein Match gefunden wird, wird der original Wert zurückgegeben.

NOTE: Wird ein anderes Wildcard-Zeichen benötigt, weil die Daten selber den Stern enthalten, kann der Aufruf im Mapping um den Parameter `wildcard='<Zeichen>'` ergänzt werden.

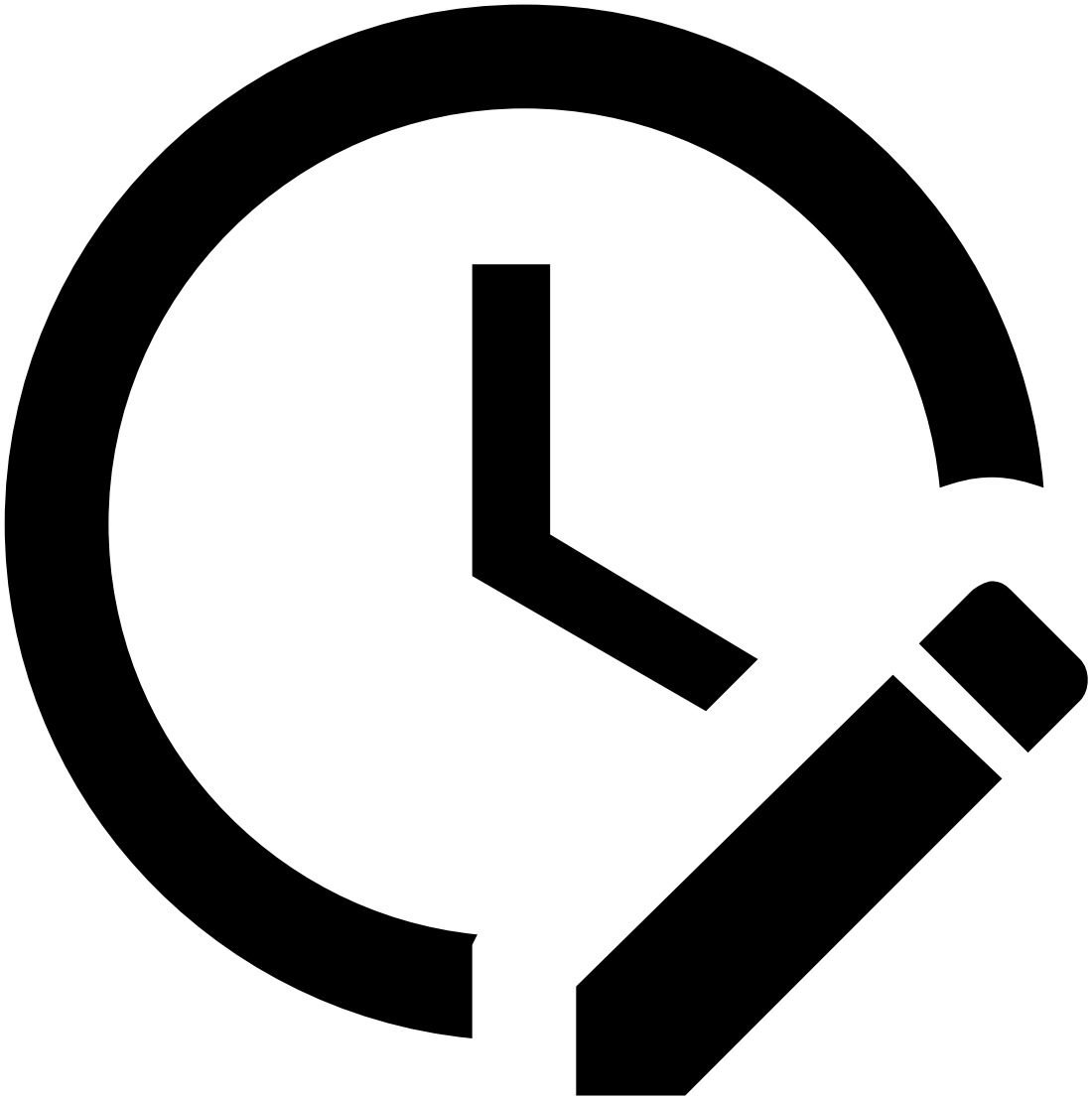
Die Felder `__comment__<NR>` werden ignoriert und dienen nur dazu Informationen über die Umsetzungen anzuhängen.

Beispiel Umsetzungstabelle:

```
{
    "__comment__1": "3: UPS",
    "Standard Versand - Zone 3": "3",
```

```
    "__comment__2": "Bahnexpress",
    "Express": "4",
    "Standard *": "5",
    "*": "6"
}
```

In dieser Beispiel Umsetzungstabelle werden Lieferarten umgesetzt. So wird der Wert `Standard Versand - Zone 3` zu `3`, alle anderen Werte die mit `Standard` beginnen zu `5` und wird kein Match gefunden wird der Wert `6` zurück gegeben. Ist kein Default Wert gewünscht (`"*": "6"`) kann dieser einfach weggelassen werden und es wird der Ursprüngliche Wert zurückgegeben.



Gutscheine

Überblick

- Gutscheine (Aktionscodes) werden ausschließlich in Shopware eingerichtet und verwaltet
- Es ist keine Synchronisation durch die Schnittstelle erforderlich
- Beim Bestellimport wird der eingelöste Gutschein als eigene Position auf dem WinLine-Beleg übernommen
- Voraussetzung: Der Gutschein-Artikel muss in der WinLine mit der entsprechenden Artikelnummer angelegt sein

Workflow

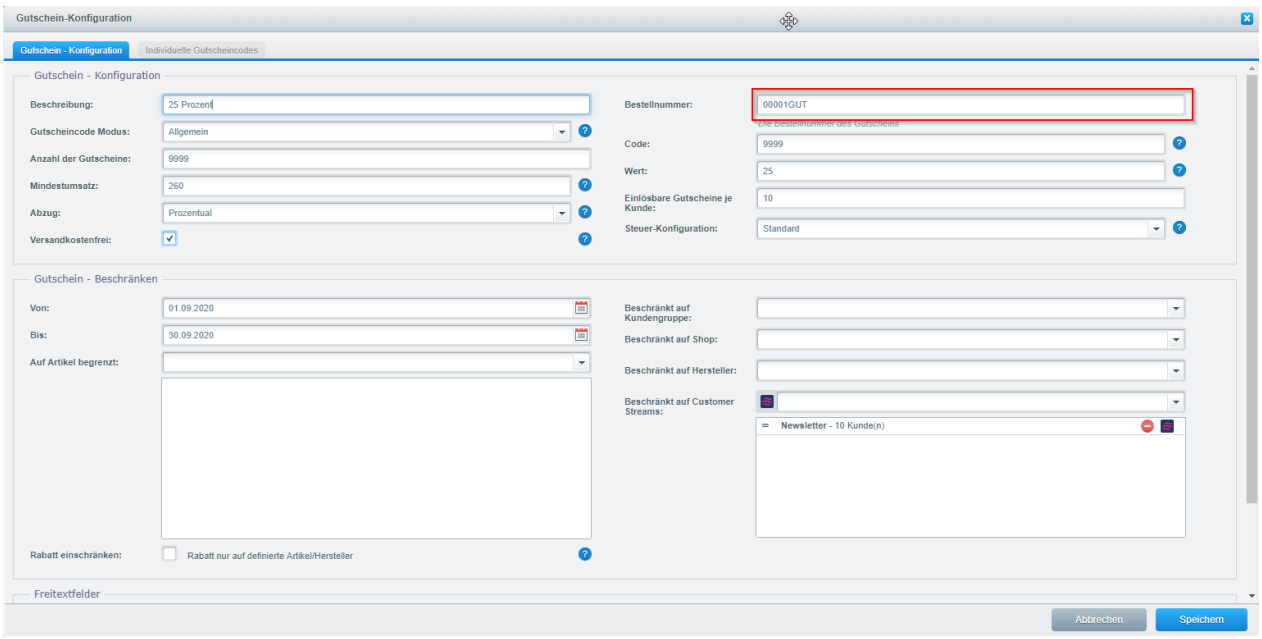
1. Gutschein/Aktionscode in Shopware anlegen (unter **Marketing > Aktionen**)
2. Einen Gutschein-Artikel in der WinLine anlegen (Artikelnummer muss mit dem Shopware-Aktionscode übereinstimmen)
3. Kunde löst den Gutschein bei einer Bestellung ein
4. Beim Bestellimport (syncOrders) erscheint der Gutschein als Position mit dem abgezogenen Wert auf dem WinLine-Beleg

Anlage in Shopware

Grundsätzlich können Gutscheine (Aktionscodes) in Shopware eingerichtet und verwendet werden.

Shopware 6 bietet unter **Marketing > Aktionen** eine Reihe von Optionen bei der Einrichtung von Aktionscodes. Es können allgemeine und individuelle Aktionscodes angelegt werden, die Anzahl und der Mindestumsatz angegeben werden. Darüber hinaus können diese zeitlich oder auf bestimmte Kundengruppen beschränkt werden. Der Aktionscode kann mit einem festen Wert oder einem prozentualen Anteil angelegt werden.

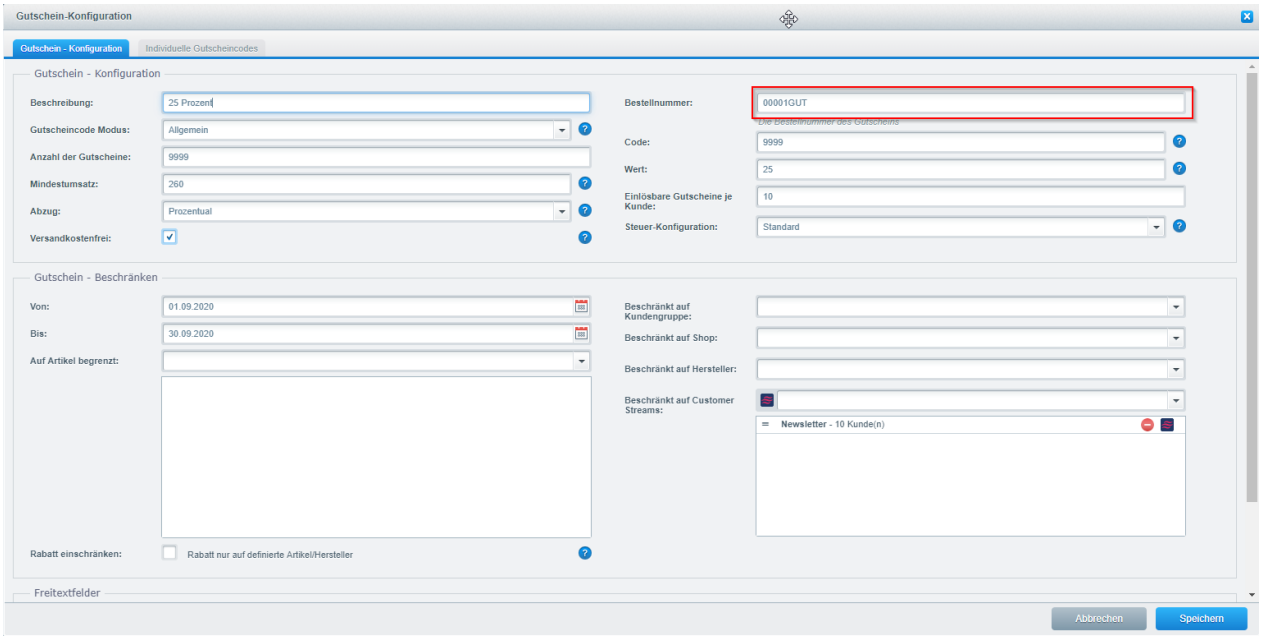
> **Screenshot aus einer älteren Shopware-Version**



Bestellung in Shopware

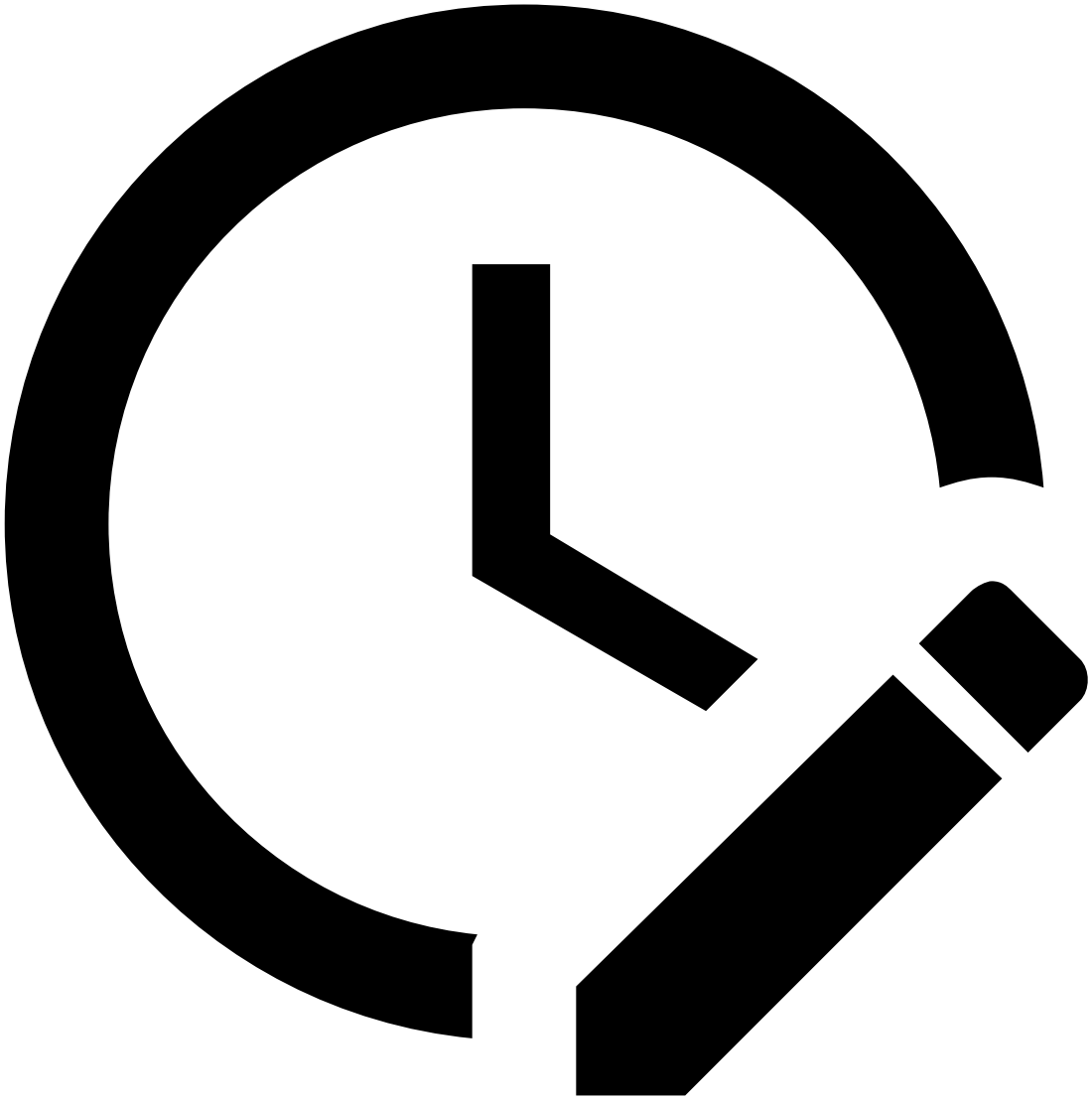
Nach Eingabe des Aktionscodes ergibt sich eine neue Position mit dem abgezogenen Wert. Diese Position taucht anschließend auf dem WinLine-Beleg auf.

Screenshot aus einer älteren Shopware-Version



Gutschein-Artikel in der WinLine

Damit diese Position auf dem Beleg ausgegeben werden kann, muss der Artikel in der WinLine mit der entsprechenden Artikel-Nr. angelegt worden sein. Dieser Gutschein-Artikel muss keinen eigenen Wert besitzen, da der Wert durch Shopware übergeben wird.



Handelsstücklisten

Modul: Handelsstücklisten

Um Handelsstücklisten an Shopware zu übermitteln wird das kostenpflichtige Modul Handelsstücklisten benötigt. Ggf. ist auch das Modul Handelsstücklisten der WinLine zu lizenzieren.

Überblick

- Handelsstücklisten ermöglichen die Darstellung von Artikelbündeln (Sets) im Shopware-Shop
- Die Zusammensetzung der Stückliste wird automatisch übertragen und kann in einem Shopware-Zusatzfeld angezeigt werden
- Der Bestand eines Stücklistenartikels wird anhand der enthaltenen Einzelartikel berechnet
- Die Erkennung erfolgt automatisch über das Feld “Artikeltyp = Handelsstückliste” in den Lagereinstellungen

Workflow

1. Artikel im WinLine-Artikelstamm als “Handelsstückliste” kennzeichnen (Lagereinstellungen → Artikeltyp)
2. Stückliste im Stücklistenstamm pflegen (enthaltene Artikel und Mengen)
3. Mapping für die Stücklistendarstellung konfigurieren (Kategorie `Stueckliste`)
4. Artikelsync starten
5. Ergebnis im Shop prüfen: Stücklisteninformationen im Zusatzfeld des Artikels sichtbar

WinLine

Einrichtung

Es ist keine gesonderte Einrichtung innerhalb der WinLine notwendig.

Verwendung

Immer vom Lager

Wenn diese Option im Stücklistenstamm aktiviert ist, werden die Bestände aus der Handelsstückliste nicht zur Bestandsberechnung herangezogen.

Bedarfsvorschau

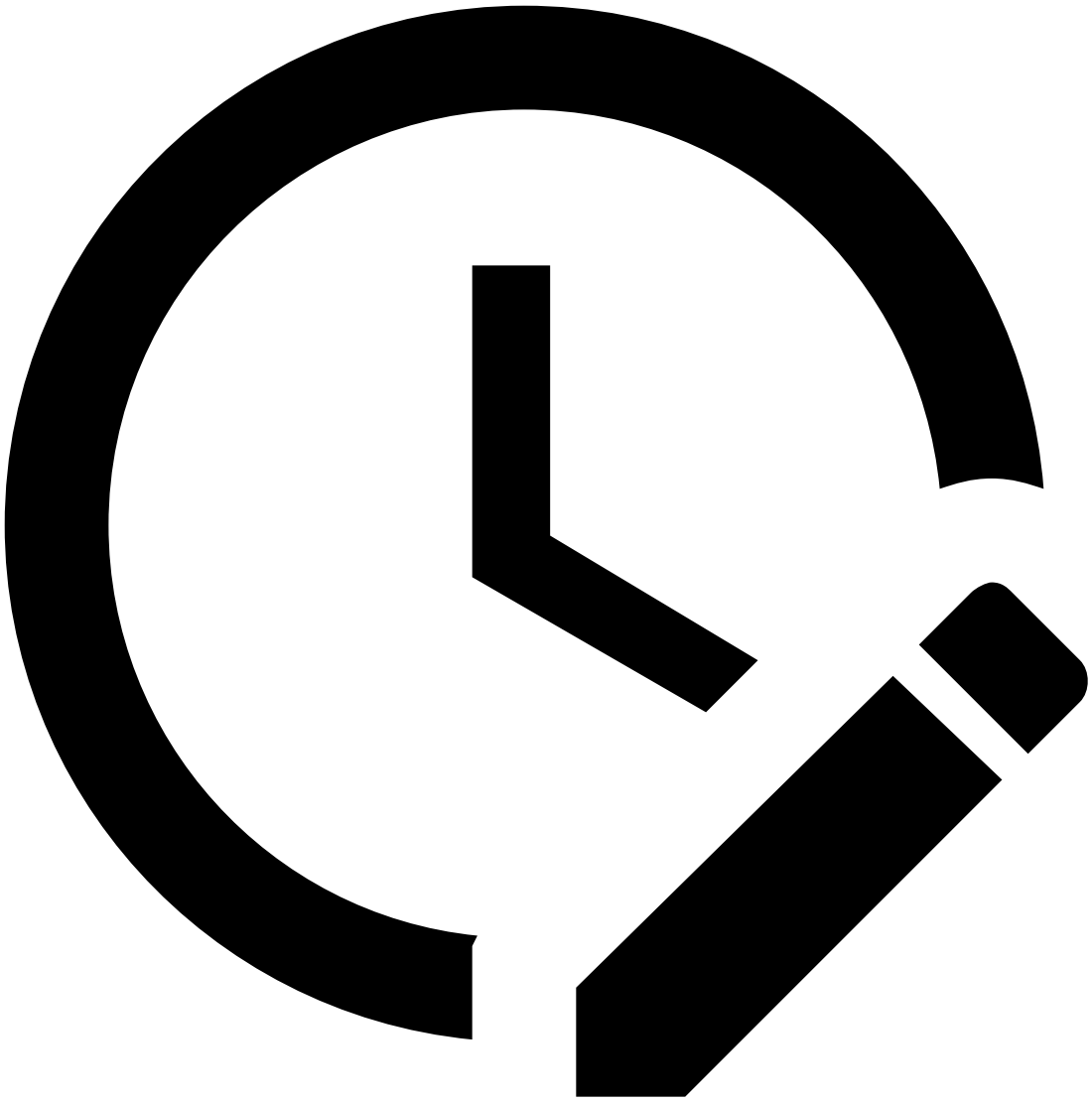
Ohne das Modul Bedarfsvorschau der Schnittstelle können Bestände für in der Stückliste enthaltene Artikel nicht korrekt in Shopware dargestellt werden wenn Belege als Angebot oder Auftrag eingelesen werden. Dies liegt daran, dass die WinLine Bestandsbuchungen für Stücklistenartikel erst in der Stufe Lieferschein durchführt.

Die Übermittlung von Stücklisten wird automatisch für Artikel aktiviert die das Feld Artikeltyp in den Lagereinstellungen des Artikelstamms auf `Handelsstückliste` gesetzt haben.

Schnittstelle

Einrichtung

Die Schnittstelle stellt im Mapping eine weitere Kategorie für die Handelsstücklisten zur Verfügung. Es kann die Zusammensetzung für einzelne Artikelzeilen konfiguriert werden. Diese werden dann zu einer Tabelle zusammengeführt und in ein Zusatzfeld von Shopware übermittelt



2026-03-25

Kategorien

Überblick

Der Kategorien-Sync überträgt die Kategoriestruktur von der WinLine an Shopware.

Sync-Richtung: WinLine → Shopware

Es stehen zwei Strategien zur Kategorieverwaltung zur Verfügung:

1. **Artikeluntergruppen**: Einfache Zuordnung über die bestehenden Artikeluntergruppen der WinLine. Ein Artikel kann jedoch nur einer Kategorie zugeordnet werden. Übersetzungen sind in dieser Variante nicht möglich.
2. **Mehrfachauswahl aus Eigenschaft** (kostenpflichtiges Modul): Flexible Zuordnung über ein Eigenschaftsfeld vom Typ M (Mehrfachauswahl). Beliebig viele Kategorien pro Artikel möglich. Hierarchische Kategorien werden über Backslash-Trennung abgebildet.

Workflow

Variante 1: Artikeluntergruppen

1. Artikeluntergruppen in der WinLine anlegen
2. In der `config.ini` die Einstellung `Kategorien_aus_Artikeluntergruppen = True` im Segment `WL` setzen
3. Kategorien-Sync starten
4. Kategoriebaum im Shop prüfen

Variante 2: Mehrfachauswahl aus Eigenschaft

1. Eigenschaft vom Typ M (Mehrfachauswahl) in der WinLine anlegen
2. Kategorien als Optionen hinterlegen im Format: `Kategorie\Unterkategorie\Unter-Unterkategorie`
3. Eigenschaftsnummer in der `config.ini` unter `[PROPERTY]` → `categories` eintragen
4. Artikel die gewünschten Kategorien zuweisen
5. Kategorien-Sync starten
6. Kategoriebaum und Zuordnungen im Shop prüfen

Komplettsync nach Änderungen

Nach Änderungen am Kategorie-Mapping oder der Kategoriestrategie ist immer ein Komplettsync aller Artikel notwendig.

Aus Artikeluntergruppen

Es können die Artikeluntergruppen als Webshop-Kategorien übernommen werden. Hierzu muss die Einstellung `Kategorien_aus_Artikeluntergruppen` im segment `WL` auf `True` gesetzt werden. ~~Auch die hinterlegten Grafiken der Artikeluntergruppen werden übermittelt.~~

In dieser Variante kann aber ein Artikel immer nur einer Kategorie zugeordnet sein.

Zudem kann die Übersetzung nicht gespeichert werden, daher sind alle Übersetzungen in der gleichen Sprache wie die Hauptsprache.

Filter über Eigenschaften Plugin

Es ist möglich nur bestimmte Artikeluntergruppen zu übertragen, wenn diese in den Eigenschaften eine Checkbox definiert haben und das Plugin `get_article_groups_filter_by_eigenschaften` installiert ist.

```
t070_Eigenschaften.C002_Eigenschaft == 1015 # hier die Eigenschaftenkennzahl eintragen

t309_Artikeluntergruppen.C000_Untergruppennummer > '00000-00000-00000-00000-00000' # hier die Untergruppennr eintragen
```

Mehrfachauswahl aus Eigenschaft

> **Modul: Multi-Kategorien**
>
> Um Kategorien aus Eigenschaften an Shopware zu übermitteln wird das kostenpflichtige Modul Multi-Kategorien benötigt.

Kategorien können auch aus einer Eigenschaft des Typs M (Mehrfachauswahl) übermittelt werden. Dazu müssen alle wählbaren Kategorien als Option der Eingeschaft hinterlegt werden. Der Aufbau sieht folgendermaßen aus:

```
Kategorie\Unterkategorie\Unter-Unterkategorie
```

Wobei eine neue Kategorie bei jedem ‘\’ erstellt und der vorangegangenen Kategorie untergeordnet wird.

In der Datei `config.ini` ist die verwendete Eigenschaftsnummer anzugeben.

```
[PROPERTY]
categories = [EIGENSCHAFTSNUMMER]
```

In dieser Variante können beliebig viele Kategorien pro Artikel zugeordnet werden.

Artikeluntergruppen in Eigenschaften übertragen

Um die bestehenden Artikeluntergruppen in Eigenschaftsfelder zu übertragen, müssen diese exportiert und per MMR eingespielt werden. Damit können die Artikeluntergruppen als eine Mehrfachauswahl verwendet werden.

Artikeluntergruppen exportieren:

FAKT > Auswertungen > Stichtagsbezogene Bestandsliste.

Per Excel über Artikel-Nr. + Bezeichnung + Artikeluntergruppen zusammenbauen =VERKETTEN (SPALTE1;"";SPALTE";"...). Daraus die bestehenden Artikeluntergruppen zusammenstellen.

Makro Autoanlage für Artikel-Kategorien

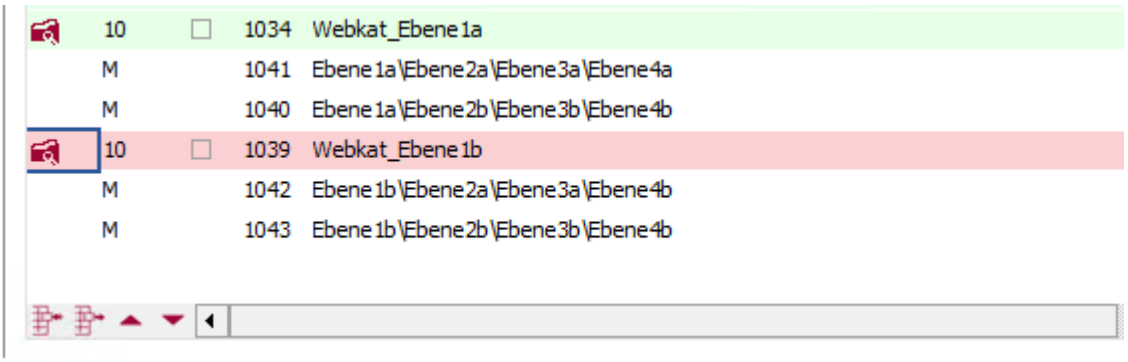
AUTOANLAGE_EIGENSCHAFTEN.MMR

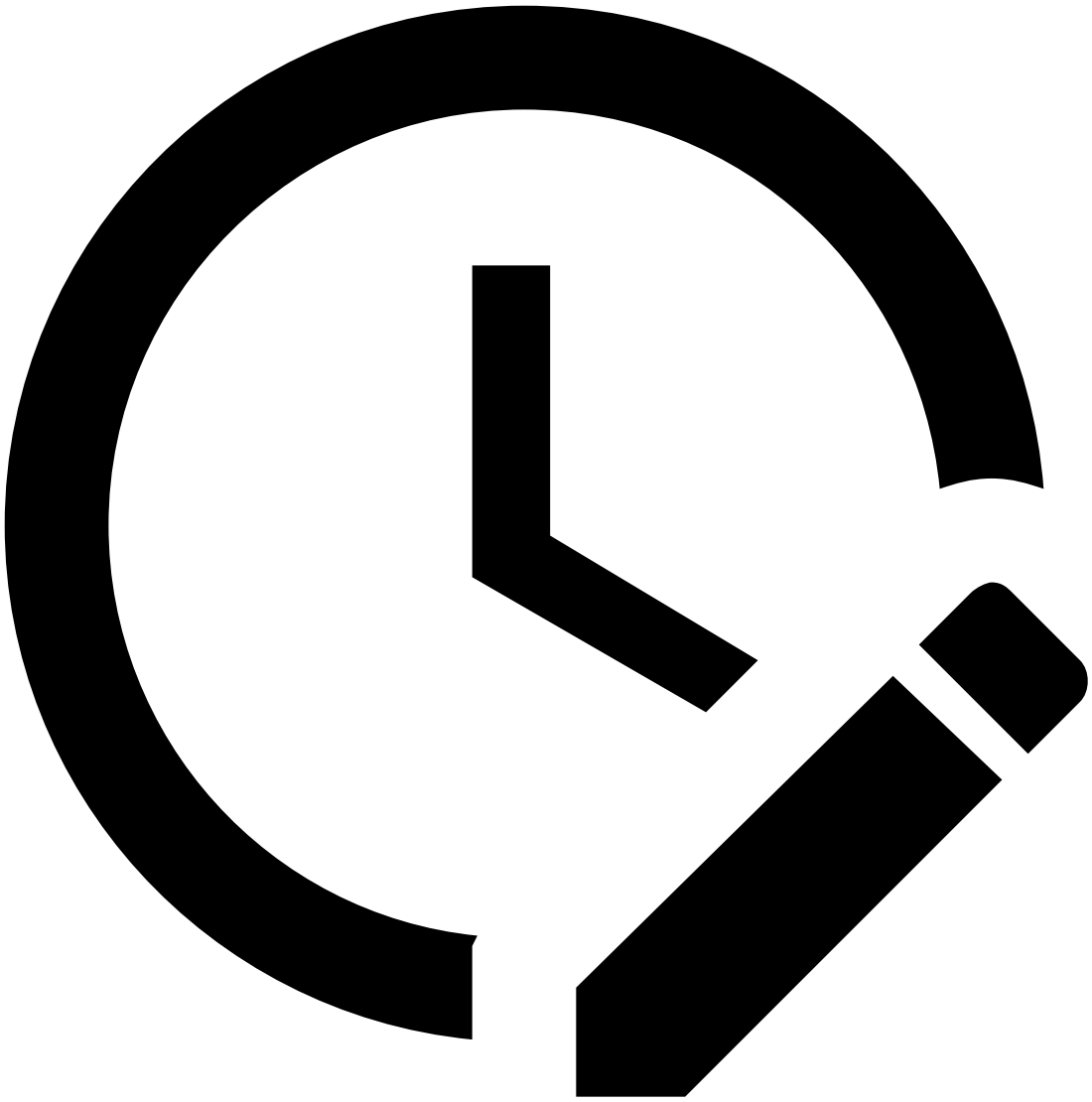
Einsatz des Makros

Die erste Ebene muss händisch angelegt werden. Um die Excel-Datei lesen zu können, muss auf dem Kundenrechner eine Office-Version installiert sein. Reihenfolge umdrehen.

Kategorien Artikel zuweisen

Anschließend per EXIM den Artikeln die Kategorien (Eigenschaften) zuweisen. Die Mehrfachauswahl muss anschließend durch den Kunden erfolgen.





2026-03-25

Konfigurator

Shopware 6: Custom Products ist nativ enthalten

Die Funktionalität des ehemaligen Custom-Products-Plugins ist in Shopware 6 bereits nativ enthalten. Es wird kein zusätzliches Plugin benötigt.

Überblick

- Der Produktkonfigurator ermöglicht die individuelle Zusammenstellung eines Artikels durch den Kunden im Shop
- Die Konfiguratorartikel werden **nicht** automatisch von der Schnittstelle synchronisiert, sondern müssen manuell in Shopware angelegt werden
- Beim Bestellimport liest die Schnittstelle den Konfiguratorartikel und alle ausgewählten Varianten als einzelne Positionen in den WinLine-Beleg ein
- Voraussetzung: Jedem Konfiguratorartikel und jeder Variante muss eine gültige WinLine-Artikelnummer zugeordnet sein

Workflow

1. Konfiguratorartikel in Shopware manuell anlegen
2. Jedem Artikel und jeder Konfigurationsoption eine WinLine-Artikelnummer (Feld: Bestellnummer) zuweisen
3. Kunde konfiguriert den Artikel im Shop und bestellt
4. Beim Bestellimport (syncOrders) werden alle Konfigurationsoptionen als einzelne Belegzeilen in der WinLine erfasst

Shopware

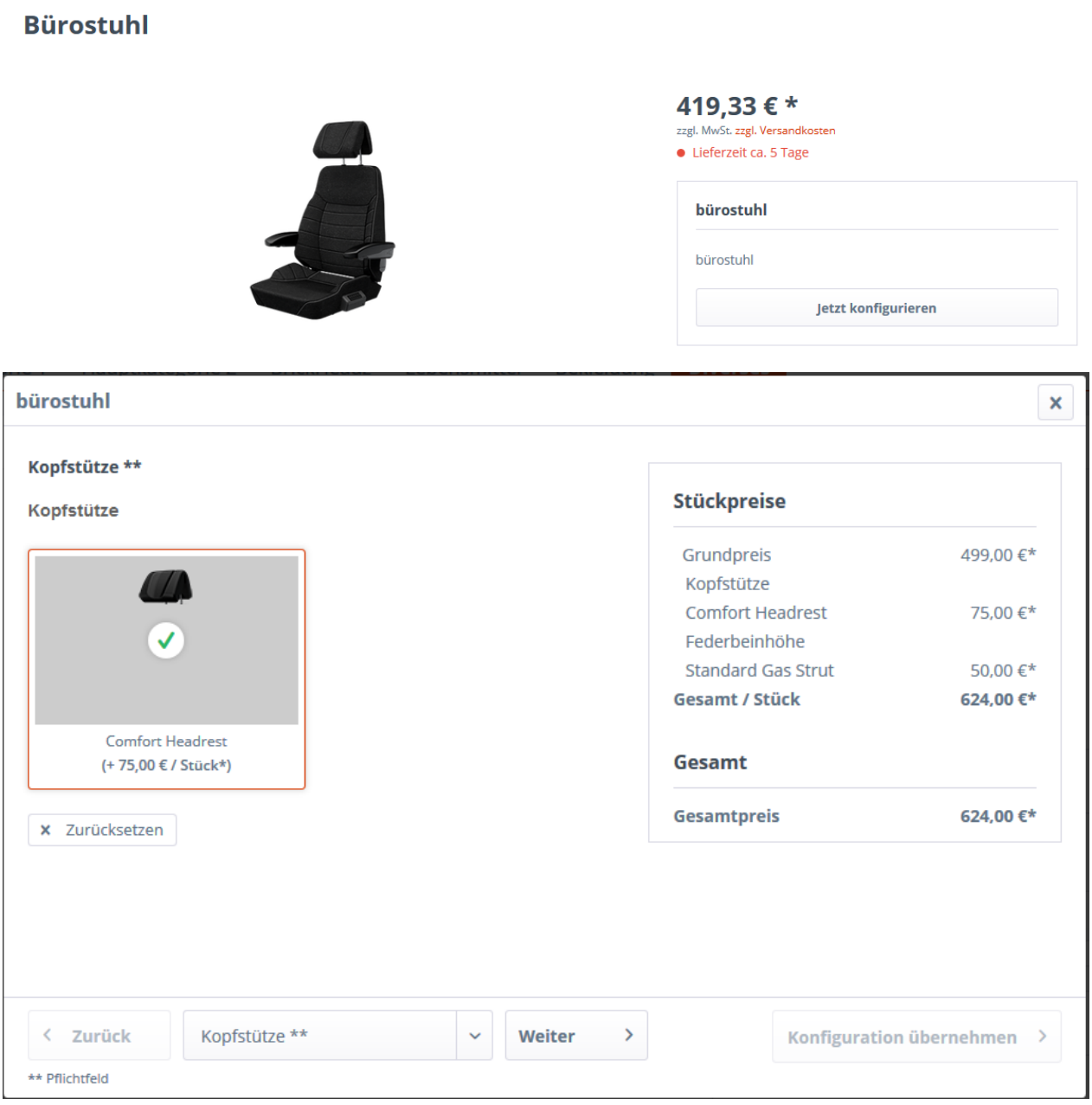
Einrichtung

Einrichtung

Konfiguratorartikel können nicht automatisch von der Schnittstelle synchronisiert werden. Sie müssen manuell in Shopware angelegt werden. Jedem Artikel muss eine gültige WinLine-Artikelnummer (Feld: Bestellnummer) zugeordnet werden. Die Schnittstelle liest den Konfiguratorartikel und alle ausgewählten Varianten als Einzelzeilen in einen WinLine-Beleg ein.

Screenshots

Die folgenden Screenshots stammen aus einer älteren Shopware-Version



Stückpreise

Grundpreis	499,00 €*
Kopfstütze	
Comfort Headrest	75,00 €*
Federbeinhöhe	
Standard Gas Strut	50,00 €*
Rollen	
Soft Tyre Castors	25,00 €*
Gesamt / Stück	**649,00 €***

Gesamt

Gesamtpreis	**649,00 €***

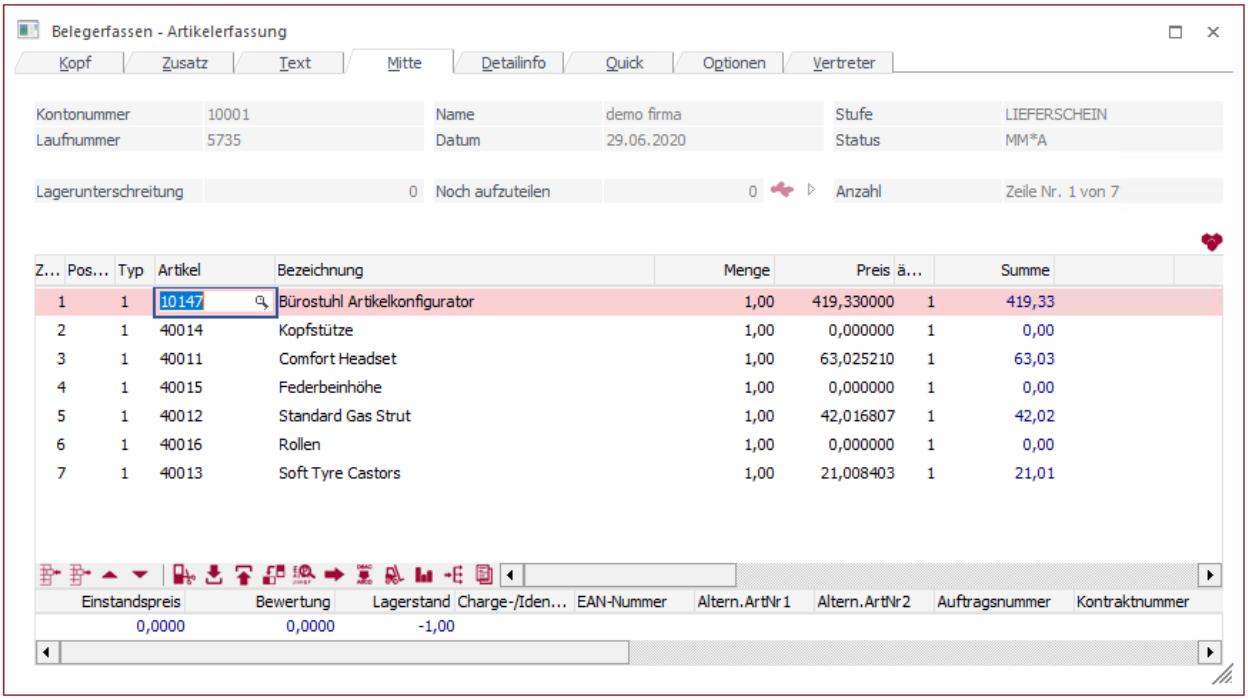
1 In den Warenkorb ›

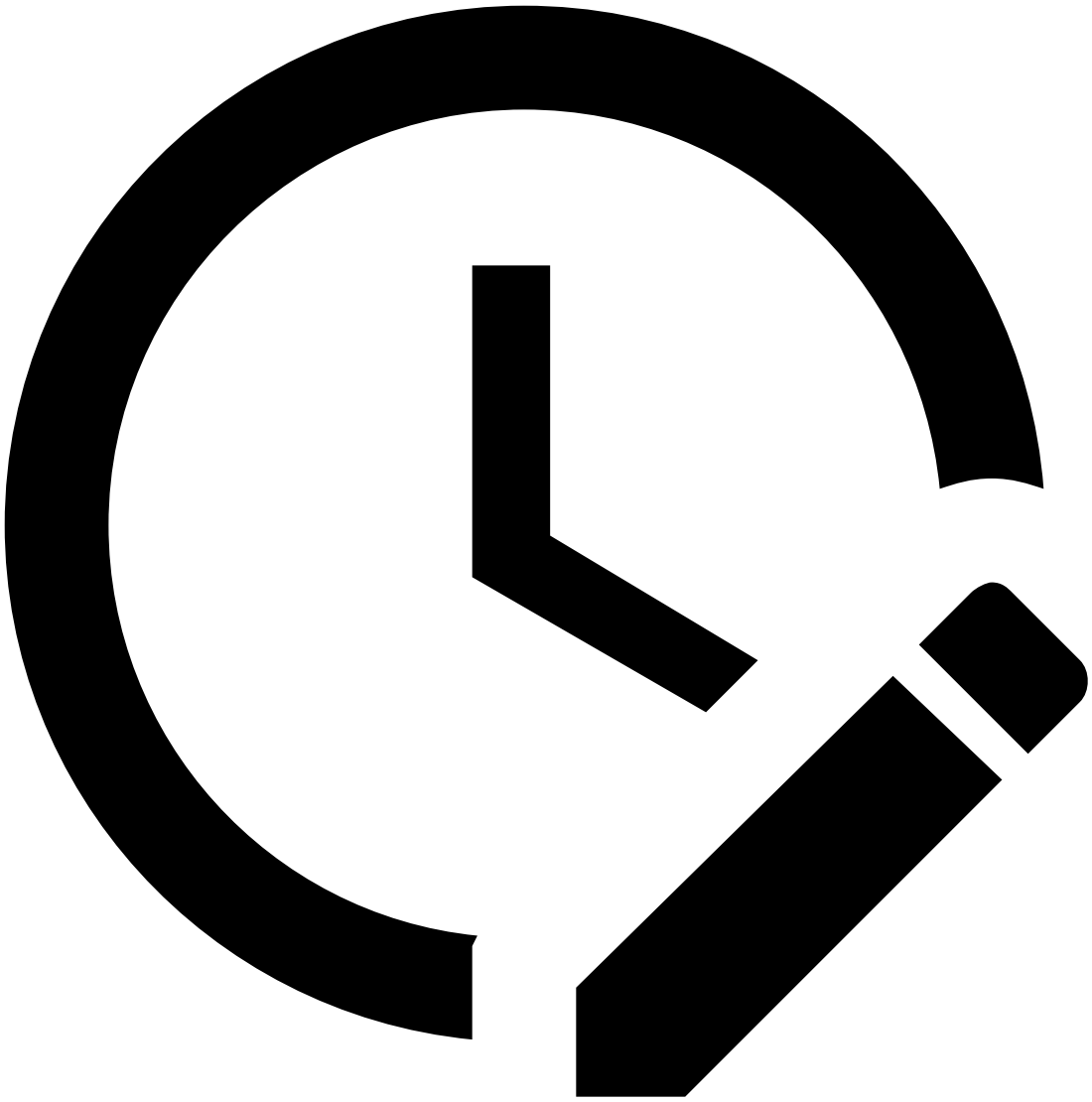
Artikel	Anzahl	zzgl. Mwst.	Summe
Bürostuhl Artikel-Nr.: 10147 Lieferzeit ca. 5 Tage	1	79,67 €	**419,33 €**
Aufschläge: Kopfstütze : Comfort Headrest (+ **63,03 € / Stück**, zzgl. Mwst. 11,97 €) Federbeinhöhe : Standard Gas Strut (+ **42,02 € / Stück**, zzgl. Mwst. 7,98 €) Rollen : Soft Tyre Castors (+ **21,01 € / Stück**, zzgl. Mwst. 3,99 €)			**126,06 €**
Konfiguration öffnen ›		**Artikelsumme gesamt:**	**545,39 €**

Gutschein-Code eingeben ›

Summe:	545,39 €*
Versandkosten:	0,00 €*
Gesamtsumme:	**649,01 €**
Gesamtsumme ohne MwSt.:	545,39 €
zzgl. 19 % MwSt.:	103,62 €

Zahlungspflichtig bestellen ›


Belegerfassen - Artikelerfassung
Kopf
Zusatz
Text
Mitte
Detailinfo
Quick
Optionen
Vertreter
Kontonummer 10001
Name demo firma
Stufe LIEFERSCHEIN
Laufnummer 5735
Datum 29.06.2020
Status MM*A
Lagerunterschreitung 0
Noch aufzuteilen 0
Anzahl Zeile Nr. 1 von 7
Z... Pos... Typ Artikel Bezeichnung Menge Preis ä... Summe
1 1 10147 Bürostuhl Artikelkonfigurator 1,00 419,330000 1 419,33
2 1 40014 Kopfstütze 1,00 0,000000 1 0,00
3 1 40011 Comfort Headset 1,00 63,025210 1 63,03
4 1 40015 Federbeinhöhe 1,00 0,000000 1 0,00
5 1 40012 Standard Gas Strut 1,00 42,016807 1 42,02
6 1 40016 Rollen 1,00 0,000000 1 0,00
7 1 40013 Soft Tyre Castors 1,00 21,008403 1 21,01
Einstandspreis Bewertung Lagerstand Charge-/Iden... EAN-Nummer Altern.ArtNr1 Altern.ArtNr2 Auftragsnummer Kontraktnummer
0,0000 0,0000 -1,00



2026-03-25

Kundenimport Shopware to WinLine

Überblick

- Die Schnittstelle überträgt Kundendaten aus Shopware in die WinLine
- Kunden können als Personenkonto oder als Kontakt/Ansprechpartner importiert werden
- Es gibt vier Varianten für den Umgang mit Kundennummern:
    1. Kundennummer wird von Shopware vergeben
    2. Kundennummer wird beim WinLine-Import vergeben
    3. Alle Bestellungen auf ein Sammelkonto buchen
    4. Sammelkonto nur als Fallback für unbekannte Kunden
- Der Import kann über XML-Dateien oder direkt per EXIM-Datenbank erfolgen
- Über Umsetzungstabellen können importierte Werte transformiert werden (z.B. Anreden)

Sync-Befehle

- **syncFilteredCustomers** / **syncCustomers -to wl**: Kundendaten aus dem Shop abrufen
- **syncCustomersWLNumber**: Kundennummern aus der WinLine zurückschreiben

Grundsätzlich können die Kundendaten aus Shopware als Personenkonto oder Ansprechpartner/Kontakt an die WinLine übergeben werden.
Es sollte vorab ein klar Workflow definiert werden, damit zum Zeitpunkt des Order-Imports die entsprechenden Kundendaten zur Verfügung stehen.
Kundendaten werden anhand des timestamps heruntergeladen. Auch geänderte Kundaten können auf diese Weise immer aktuell ins System gezogen werden.
!!! Vorsicht: Alle in der Vorlage definierten Vorbelegungen werden in diesem Fall durch den erneuten Import überschrieben. Um dieses Verhalten zu umgehen, kann mit dem Mapping `existingCustomer.yml` die Werte vorbelegt werden, sodass diese nicht überschrieben werden.

```
last_customer_sync = 2021-12-07T15:14:16.494653+01:00
```

Plugin zur Filterung von Kundenimporten

Das Plugin “getCustomersFiltered.py” aus “beispiel-plugins” lädt nur die Kunden, die noch nicht in der WinLine angelegt wurden und berücksichtigt darüber hinaus nur Kunden die eine Bestellung im Shop getätigt haben.

```
getCustomersFiltered.py
```

!!! Vorsicht: Leider werden nur abgeschlossene Bestellungen im OrderCount gezählt.

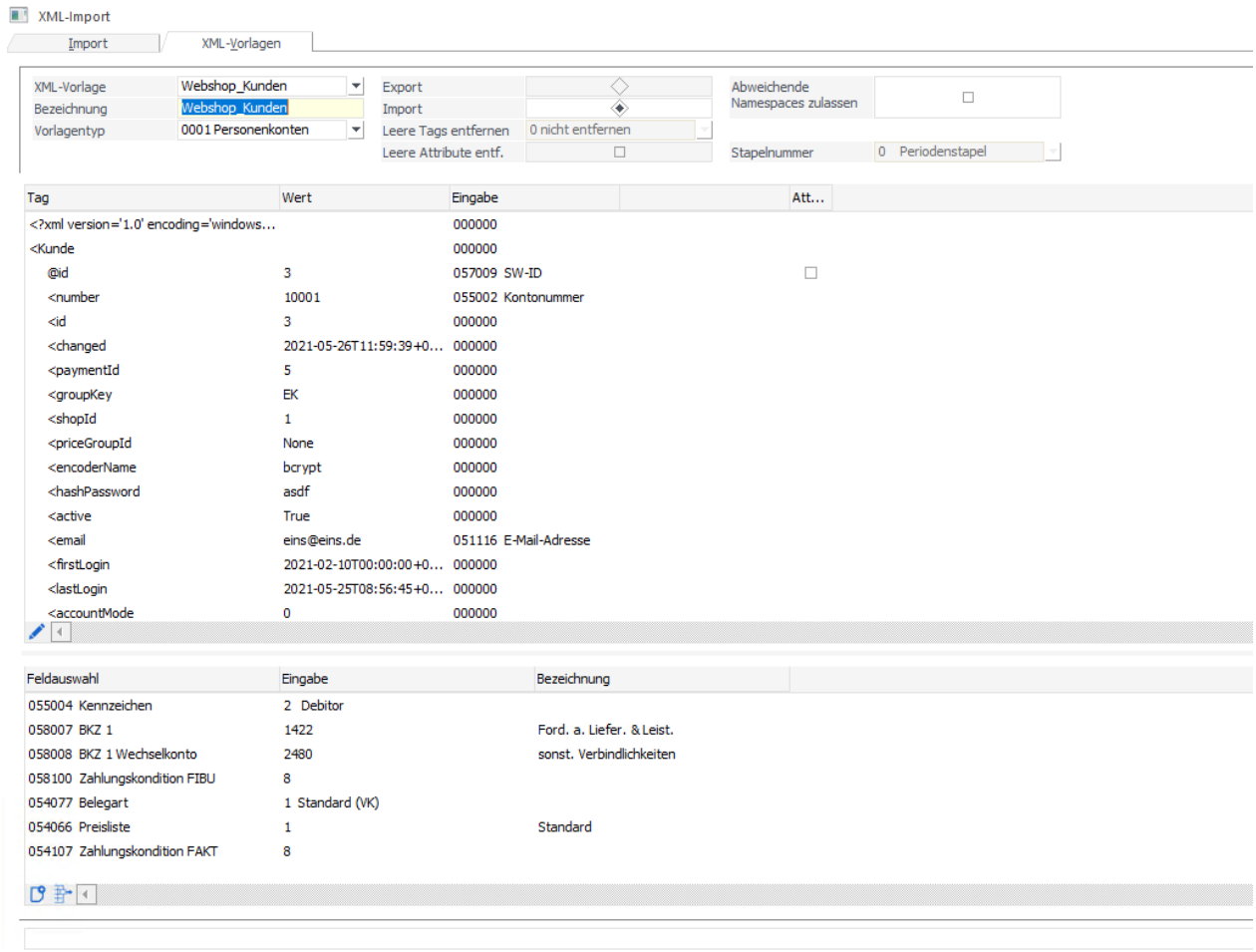
Es gibt vier Möglichkeiten mit den Kundendaten aus Shopware umzugehen.

1. Kunden erhalten von Shopware eine Kundennummer
2. Kunden erhalten beim Import von der WinLine eine Kundennummer
3. Alle Bestellungen werden auf ein Sammelkonto gebucht.
4. Alle Bestellungen einer in der WinLine nicht existierenden Kundennummer werden auf ein Sammelkonto gebucht.

1. Import mit Kundennummer aus SW

Es wird ein Kundennummern-Kreis für Shopkunden festgelegt und dieser in der WinLine und Shopware angelegt.
Bei einer Neuregistrierung eines Kunden werden die angegebenen Daten von der Schnittstelle (syncFilteredCustomers) als XML-File zur Verfügung gestellt.

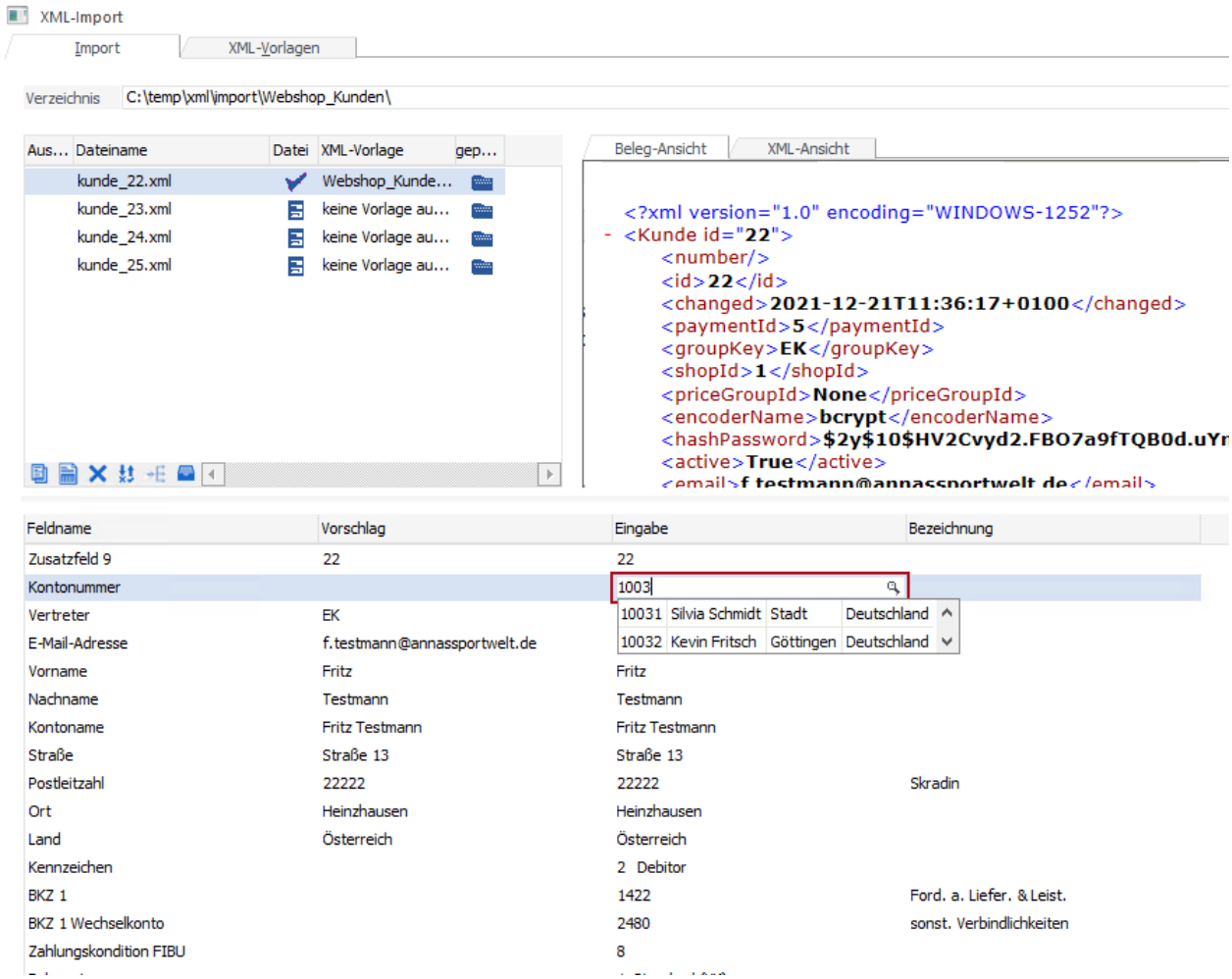
XML-Import Vorlage in der WinLine mit entsprechenden Vorbelegungen:



2. Import ohne Kundennummer

Beim Import ohne Kundennummer wird die Kundennummer beim Import in die WinLine vergeben.
Wie oben beschrieben werden die XML-Files mit den Kundendaten zur Verfügung gestellt.

Beim Import wird manuell eine Kundennummer vergeben und das Personenkonto angelegt.
Mit dem Sync (syncCustomersWLNumber) wird diese Kundennummer anschließend in den Shop zurückgeschrieben, damit diese bei weiteren Bestellungen direkt zur Verfügung steht.



3. Sammelkonto

Im nachfolgenden Fall werden alle Orders auf das Sammelkonto 50000 gebucht.

config:

```
; Kundennummer für Orders ohne oder mit in der WinLine nicht hinterlegten Kundenn
sammelkonto = 50000
always_sammelkonto = True
```

4. Sammelkonto fallback

Im nachfolgenden Fall werden alle Orders ohne WinLine Kundennummer das Sammelkonto 50000 gebucht.

config:

```
; Kundennummer für Orders ohne oder mit in der WinLine nicht hinterlegten Kundenn
sammelkonto = 50000
always_sammelkonto = False
```

Umsetzungstabellen für Kundenimport

Es ist darüber hinaus möglich Umsetzungen zu realisieren, um von SW zur Verfügung gestellte Daten für den Import anzupassen.
Es können auch ergänzende Werte hinzugefügt werden.

Der nachfolgende Ordner muss in der Config angegeben werden.
config:

```
xml_preferences = C:\Program Files (x86)\energy-shop-sync-sentry\mapping\shopware
```

Dort werden die Dateien add_tags.json & refactor_xml.json abgelegt. (siehe ..\Beispiel-Konfiguration\mapping)

Beispiele zur Umsetzung

1. Anrede von “mr” zu “Herr” und von “ms” zu “Frau”

refactor_xml.json:

```
"salutation": {
   "ms": "Frau",
   "mr": "Herr"
},
```

2. Der Wert aus “groupKey” wird zu einem neuen Wert im neuen tag “Preisliste”

refactor_xml.json:

```
  "groupKey": {
    "EK": "40|Preisliste",
    "H": "40|Preisliste"
  },
```

add_tags.json:

```
"Preisliste": {}
```

3. Der Wert aus “email” wird in den neuen neuen tag “email-copy” kopiert

refactor_xml.json:

```
   "email-copy": "<email>",
```

add_tags.json:

```
  "email-copy": {}
```

4. Die Werte aus parent “ShippingAddress” “areaId” & “vatId” werden für den tag “Belegart” zur Verfügung gestellt.

Diese stehen pipegetrennt (|) zur Vergügung. Das Umsetzungergebnis landet im tag “Belegart”. Mit dem “*” vor “vatId” in add_tags wird ein beliebiger Wert zu “*” - ein “None” bleibt als Option kombinierbar.

refactor_xml.json:

```
   "Belegart": {
       "1|*": "1",
       "1|None": "3"
   }
}
```

add_tags.json:

```
  "Belegart": {
    "ShippingAddress": {
      "areaId": "",
      "*vatId": ""
    }
  }
```

Existierende Kunden

über das Mapping ‘existingCustomer’ können die Pflicht-Felder bei erneutem import mit den werten aus der WinLine überschreiben werden. Da beim ersten Import höchstwahrscheinlich ein Default gesetzt wurde kann so im nachhinein der wert in der WinLine gesetzt werden, ohne dass dieser wieder Überschrieben wird.

Kundenimport Per Mapping

Es ist nun auch möglich importierte Kunden durch eine Mappingdatei so zu transformieren, dass sie von EXIM akzeptiert werden.

Kunden als Kontakt/Ansprechpartner

Über das Feature Flag `CUSTOMERS_AS_CONTACT = True` können die Accounts als Kontakt/Ansprechpartner in der WinLine angelegt werden.

Über die Mapping-Dateien `customer6_kontakt` und `customer6_kontakt_exisiting` können die Daten für den Import angepasst werden.

In der Konfiguration-Datei muss die WinLine Vorlage für die Kontakte in dem Wert `wlvorlageKontakt` im Segment `[SHOPFIELDS]` gesetzt werden. Im Standard heißt diese `Shopware6Kontakt`

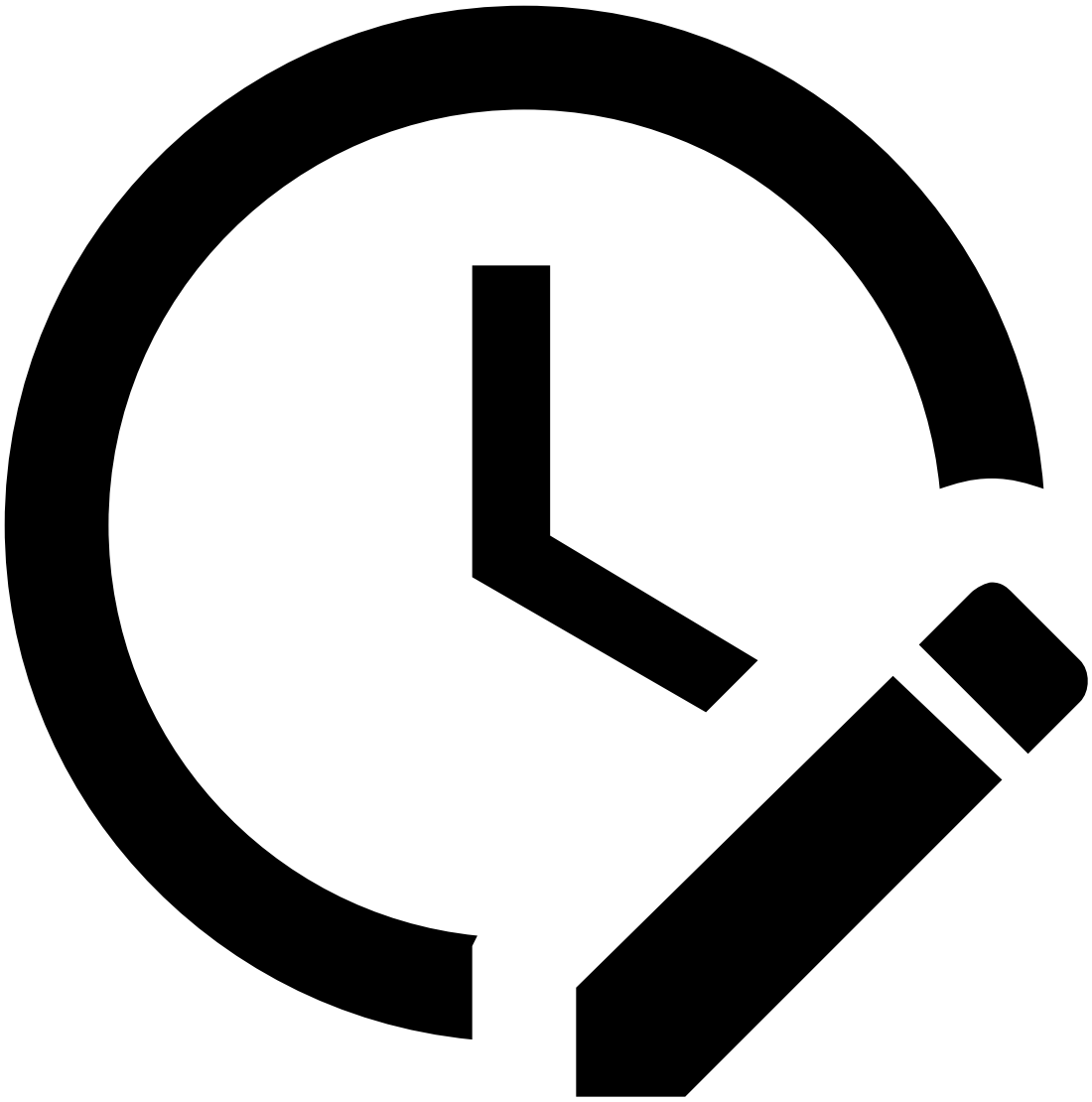
Zum Abrufen der Kunden muss der Befehl `syncCustomers -to wl` aufgerufen werden. SObald die Kontakte in der WinLine zu Personenkonten gewandelt oder zu einem Personenkonto hinzugefügt wurden, müssen über `syncCustomersWLNumber` die Accounts im Shop mit den Daten aus der WinLine synchronisiert werden. Vor jedem `syncCustomers` wird einmal `syncCustomersWLNumber` ausgeführt, dies vermeidet doppelte Abrufe.

Kunden Datenbankimport

Personenkonten und Kontakte können über die EXIM-Datenbank importiert werden.

Dazu muss in der Konfiguration die Flag gesetzt werden und die Datenbank-Tabelle angegeben werden:

```
[WL]
customers_via_exim = True
exim_customers = EXIM_KONTAKT
```



2026-04-29

Kundensync

Überblick

Die Kundensynchronisation ermöglicht den Abgleich von Kundendaten zwischen Shopware und WinLine. Zwei Synchronisationsrichtungen sind möglich:

1. **Shop → WinLine**: Neuregistrierungen aus dem Shop werden als Personenkonten oder Kontakte an die WinLine übergeben.
2. **WinLine → Shop**: Kontakte/Ansprechpartner aus der WinLine werden als Kundenkonto im Shop angelegt (derzeit nur für Shopware 5 verfügbar).

Die Kundennummernvergabe kann wahlweise durch Shopware oder durch die WinLine erfolgen.

Typischer Ablauf

Neukunde registriert sich im Shop:

1. Kunde registriert sich im Shopware-Shop (ohne Kundennummer)
2. Sync **syncFilteredCustomers** oder **syncCustomers** starten
3. Die Schnittstelle exportiert die Kundendaten als XML-Datei
4. XML-Datei über den WinLine-Import einlesen und Personenkonto anlegen
5. Sync **syncCustomersWLNumber** starten – die vergebene Kundennummer wird in den Shop zurückgeschrieben
6. Bei zukünftigen Bestellungen wird die Kundennummer automatisch mitgeliefert

> **Sync-Befehle**
>
> - **syncCustomers**: Kundendaten synchronisieren (Richtung per `-to` Parameter steuerbar)
> - **syncFilteredCustomers**: Nur Kunden mit Shopbestellungen synchronisieren (Shop → WinLine)
> - **syncCustomersWLNumber**: Kundennummern aus der WinLine in den Shop zurückschreiben
> - **syncCustomersAndOrders**: Kunden und Bestellungen in einem Durchlauf synchronisieren

---

Einstellungen Config

Um die Kundensynchronisation zu ermöglichen müssen nachfolgende Eigenschaftsfelder und Zusatzfelder in Kontakten und Personenkonten angelegt werden.

config.ini

```
[WL]
; ID Eigenschaftsfeld um Personenkonten als Shopkonto zu markieren
```

```
webkonto = <Nummer Eigenschaftsfeld>
; ID Eigenschaftsfeld um Kontaktkonten als Shopkonto zu markieren
webkonto_kontakt = <Nummer Eigenschaftsfeld>
; Zusatzfeld für die ShopwareID der Kontakte
swid_kontakt = <Nummer Zusatzfeld>
; Zusatzfeld für die ShopwareID der Personenkonten
swid_pkonto = <Nummer Zusatzfeld>

; Optional: Default Benutzer-Password für Anlage (Sektion: [SHOPPROPERTY])
defaultpw = **********************************
```

Weitere relevante Konfigurationsoptionen:

config.ini

```
[WL]
; Kundenimport über EXIM
customers_via_exim = True
; EXIM-Tabelle für Kunden
exim_customers = EXIM_KONTAKT
; Feld für Shopware-Kunden-ID in WinLine
customer_shopware_id_field = C002_Kontonummer

[SHOPFIELDS]
; WebService-Vorlage für Kunden
wlvorlagekunden = Shopware6Kunden
; WebService-Vorlage für Kontakte
wlvorlagekontakt = Shopware6Kontakt
```

Einstellungen Interface (Button):

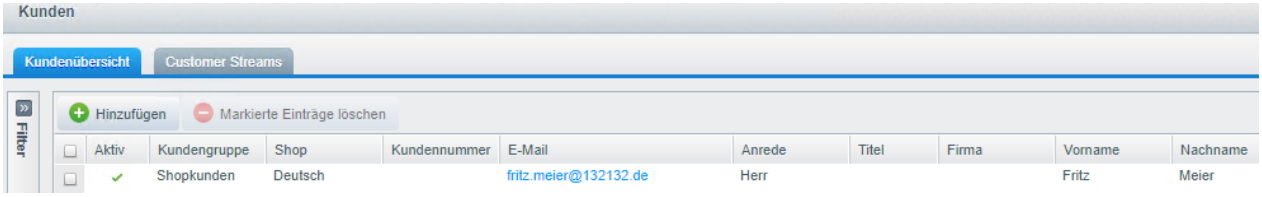
Management.ini

```
syncFilteredCustomers = Kunden SW-WL
```

Export/Import

Der neuregistrierte Kunde wird in Shopware ohne Kundennummer angelegt.

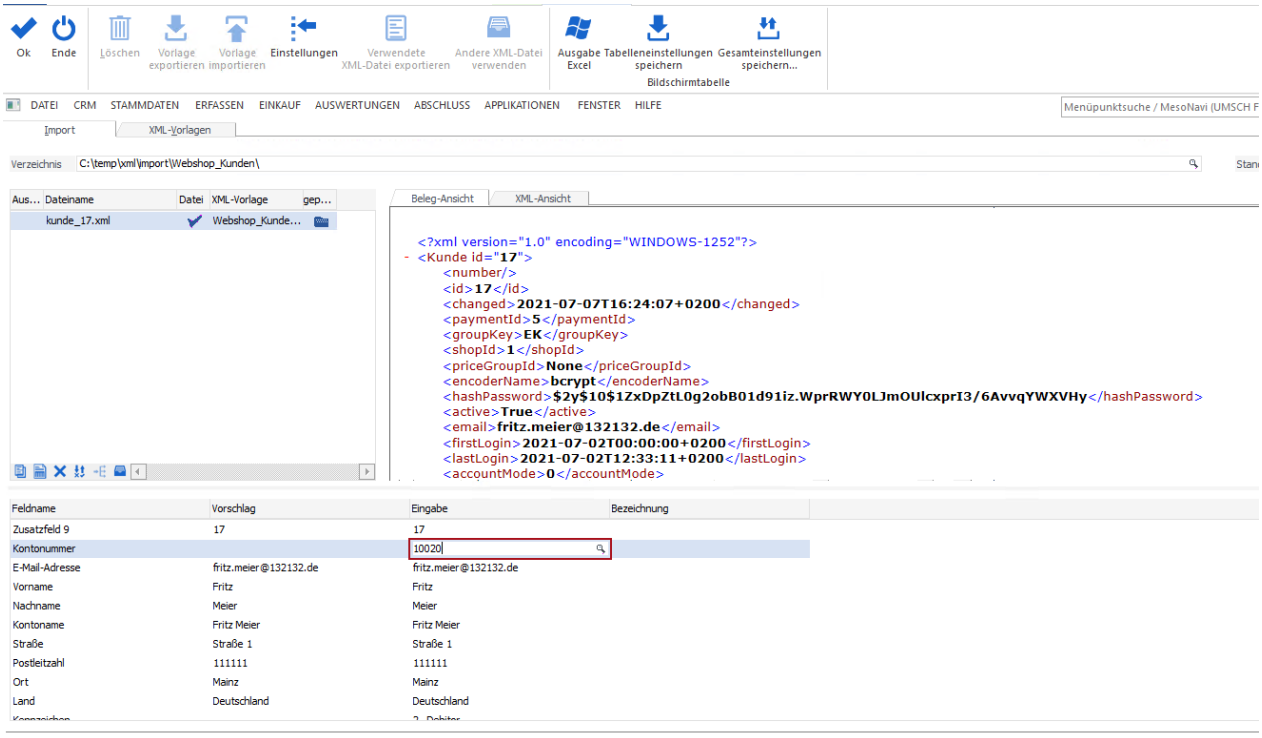
> **Screenshot aus einer älteren Shopware-Version**



Anschließend übergibt die Schnittstelle die Kundendaten des neuregistrierten Kunden als XML an einen lokalen Ordner. **config.ini**

```
; Verzeichnis in welches die Importdaten für den Kundenimport abgelegt werden
xmlpath_customers = C:\temp\xml\import\Webshop_Kunden
```

Import der Kundendaten über den WinLine XML-Import.



Drei Möglichkeiten: 1. Neuanlage Kundenkonto - Vergabe einer Kundennummer 2. Zuordnung Personenkonto - Kontaktdaten werden mit Shopdaten überschrieben 3. Zuordnung Personenkonto als Ansprechpartner - weitere XML-Vorlage (Hauptkontozuordnung über “Fibu Kontonummer”)

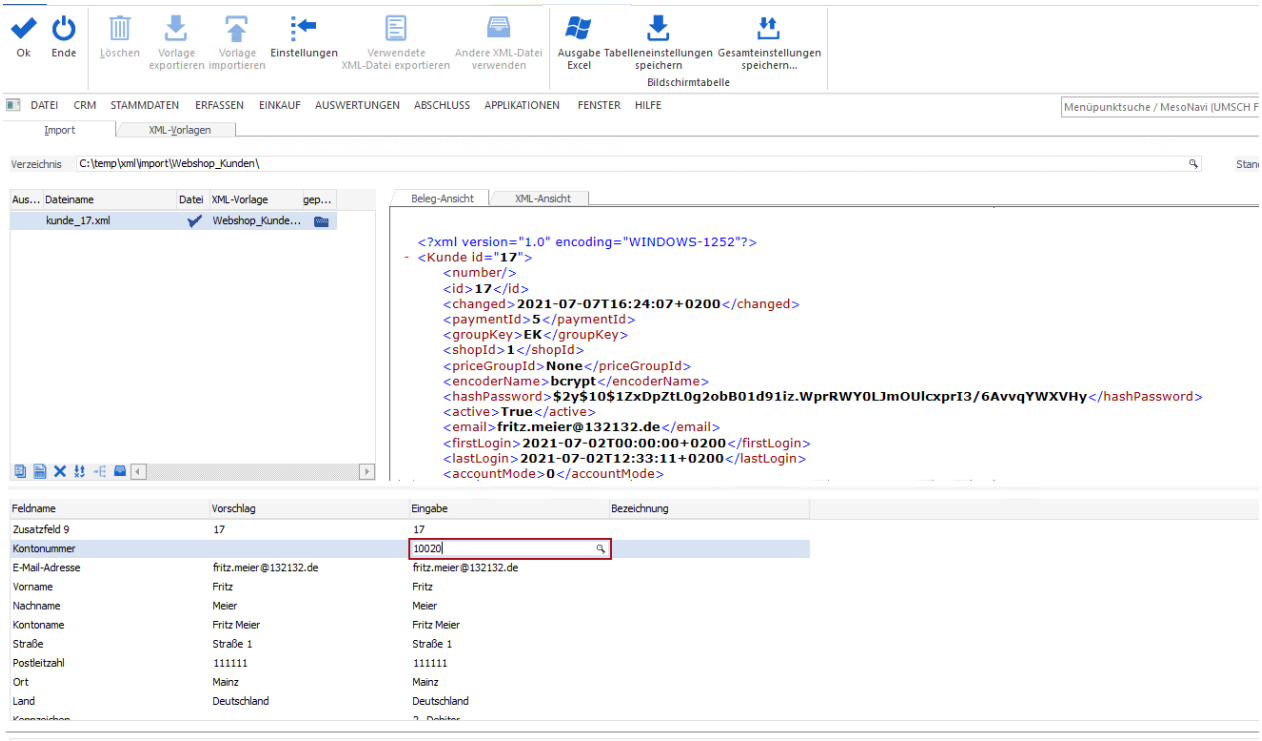
Die XML-Vorlagen müssen entsprechend in der WinLine angelegt werden. (Ein Import der Vorlagen ist nicht möglich.)

KdNr.-Vergabe in Shopware

Darüber hinaus gibt es auch die Möglichkeit die Kundennummern in Shopware direkt vergeben zu lassen und diese anschließend so in die WinLine zu importieren. Dazu ist es lediglich erforderlich den in Shopware gewählten Nummernkreis in der WinLine zu reservieren, damit es zu keinen Überschneidungen kommt. Eine Existenz-/Dublettenprüfung ist nicht vorgesehen.

WL-KundenNr. -> Shopware

Nach der Personenkontenanlage in der WinLine können die vergebenen Kundennummern nach Shopware zurückgeschrieben werden. (syncCustomersWLNumber)



Für aktuelle und zukünftige Bestellungen werden diese Kundennummern bei einer Bestellung übergeben.

WL-Kontakt -> Shopware

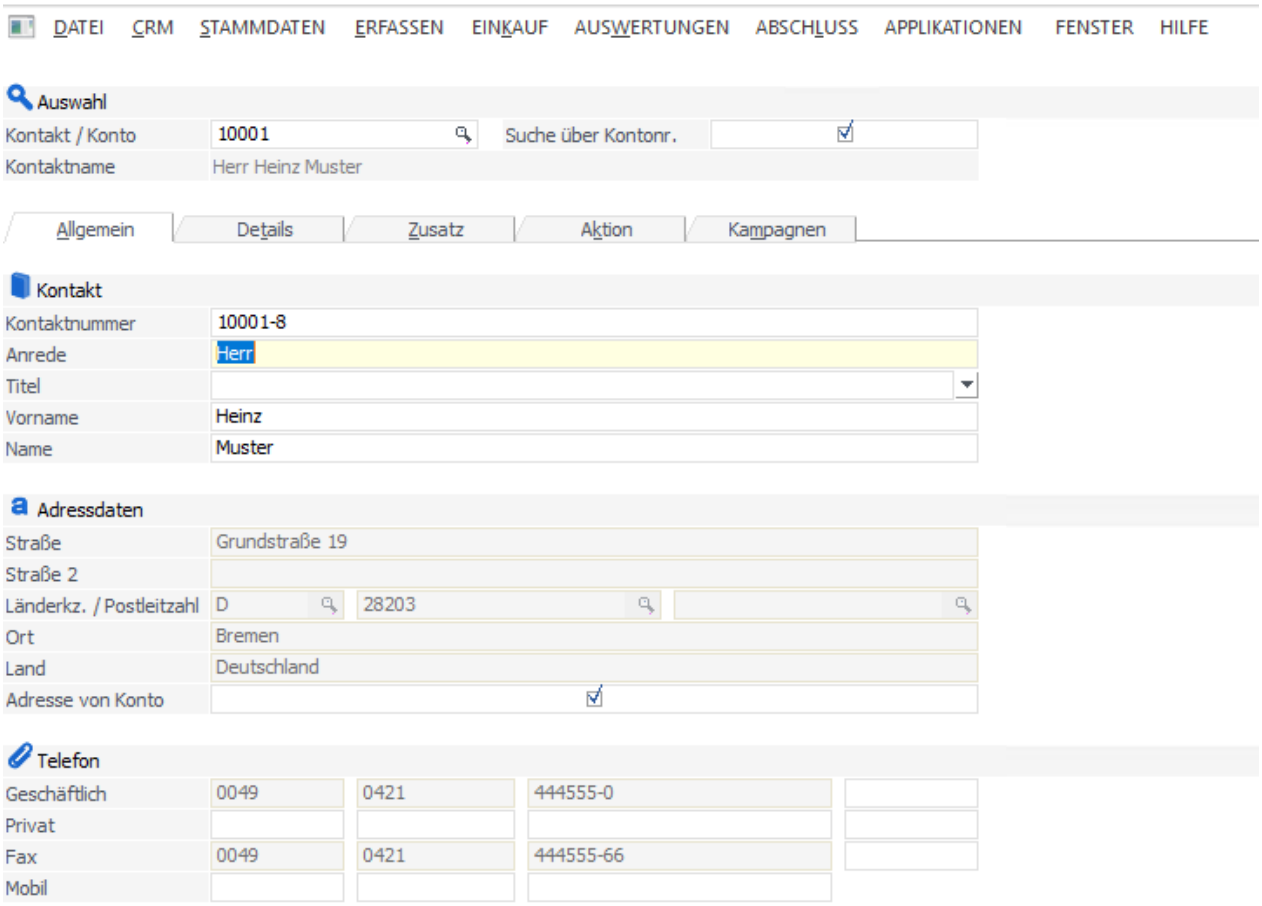
> **Nicht für Shopware 6 verfügbar**
>
> Diese Funktion ist derzeit nur für Shopware 5 implementiert. Für Shopware 6 wird beim Aufruf ein `NotImplementedError` ausgelöst. Die Implementierung für SW6 ist in Planung.

WinLine Kontakte/Ansprechpartner können als Kundenkonto an den Shop übergeben werden. In diesem Fall können auch mehrere Accounts unter einer KdNr. laufen, sodass unterschiedliche Ansprechpartner eines Unternehmens selbstständig Bestellungen im Shop vornehmen können. Die Bestellungen laufen anschließend alle auf ein Personenkonto.

Markierung über Eigenschaftsfeld

config.ini

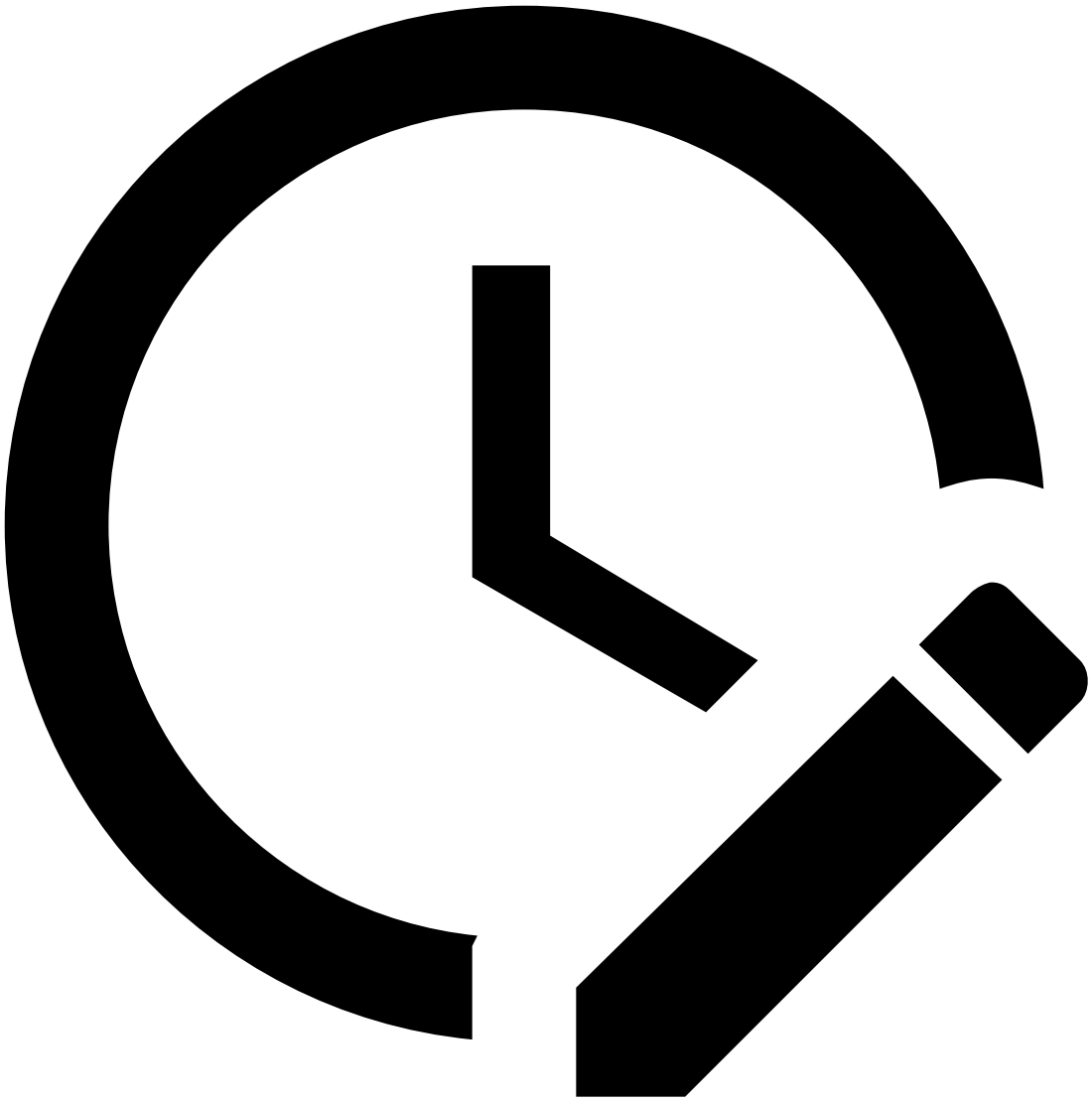
```
webkonto_kontakt = <Nummer Eigenschaftsfeld>
```



Kunden müssen anschließend über die Anlage ihres Kontos informiert werden und zur Vergabe eines Passworts aufgefordert werden (manueller/persönlicher Prozess). Änderungen der Kontaktdaten (Adressdaten) kommen anschließend als regelmäßige Aufgabe aus der WinLine. Um den Benutzer in Shopware identifizieren zu können wird die Shopware-ID in den Kontakt geschrieben.

config.ini

```
swid_kontakt = <Nummer Zusatzfeld>
```



2026-03-27

Losgröße

Überblick

Die Losgröße (auch Staffelung genannt) bestimmt, in welchen Schritten ein Artikel im Shop bestellt werden kann. Bei einer Losgröße von 10 kann der Kunde nur in 10er-Schritten bestellen (10, 20, 30, ...). Die Losgröße wird aus dem Feld “Losgröße VK” der WinLine ausgelesen und beim Artikelsync an Shopware übertragen.

Zusammenspiel mit Mindestbestellmenge

Die Losgröße hat keine Auswirkung auf die Mindestbestellmenge. Ohne gesetzte Mindestbestellmenge wird ab Menge 1 in Losgrößen-Schritten bestellt (1, 11, 21, ...). Um z.B. eine Staffelung ab 10 zu erreichen, muss die Mindestbestellmenge ebenfalls auf 10 gesetzt werden.

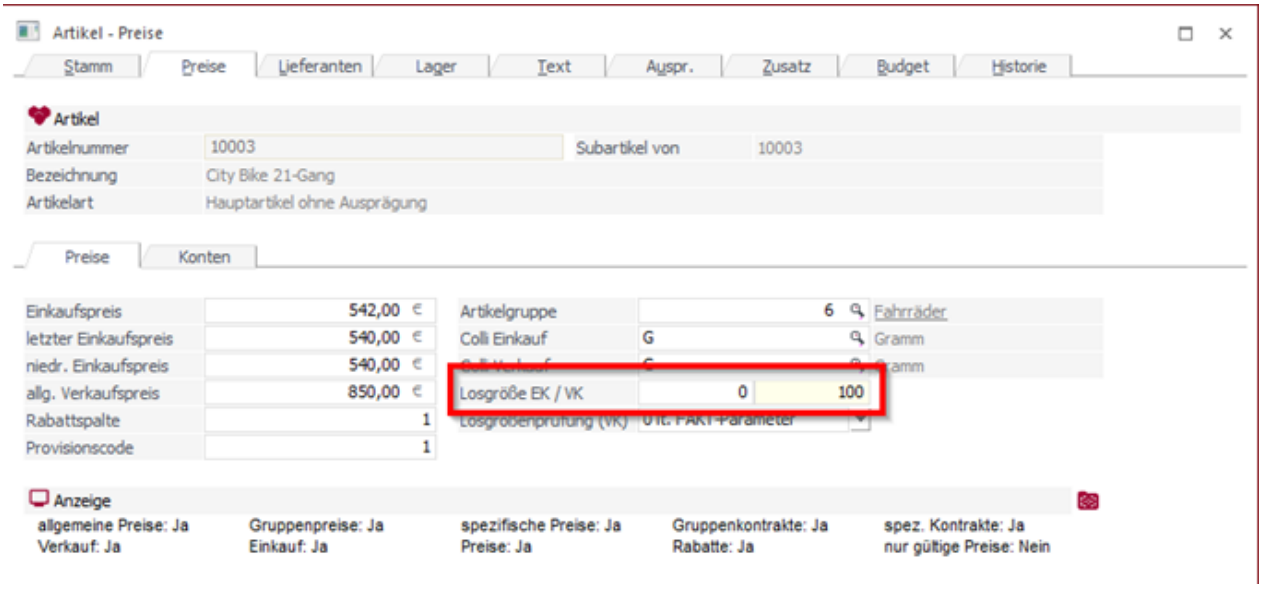
WinLine

Einrichtung

In der WinLine ist keine Einrichtung notwendig sofern das Standardfeld „Losgröße VK“ verwendet wird.

Verwendung

Die Losgröße kann in dem Feld „Losgröße VK“ hinterlegt werden.



Shop

Bei einer Losgröße von 10 kann im Shop nur in 10er Schritten bestellt werden.

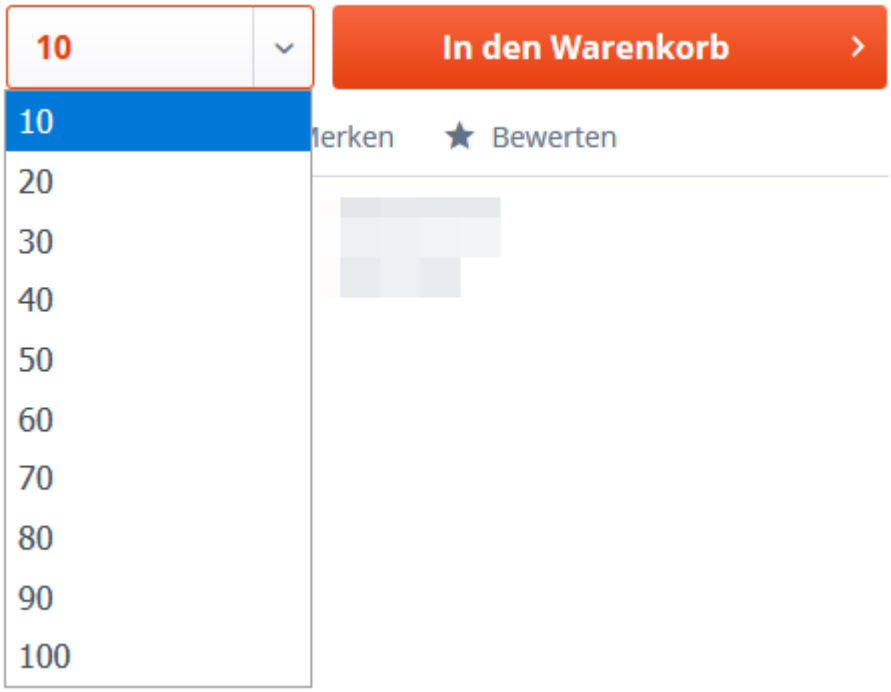
Mindestbestellmenge

Die Losgröße hat keine Auswirkung auf die Mindestbestellmenge! Um ein Ergebnis wie in diesem Beispiel zu erreichen muss die Mindestbestellmenge daher auf 10 gesetzt werden. Wird die Mindestbestellmenge nicht gesetzt wird in 10er Schritten ab Menge 1 bestellt. (1, 11, 21, 31 ...)

Screenshots aus einer älteren Shopware-Version

Backend Shopware - Staffelung:

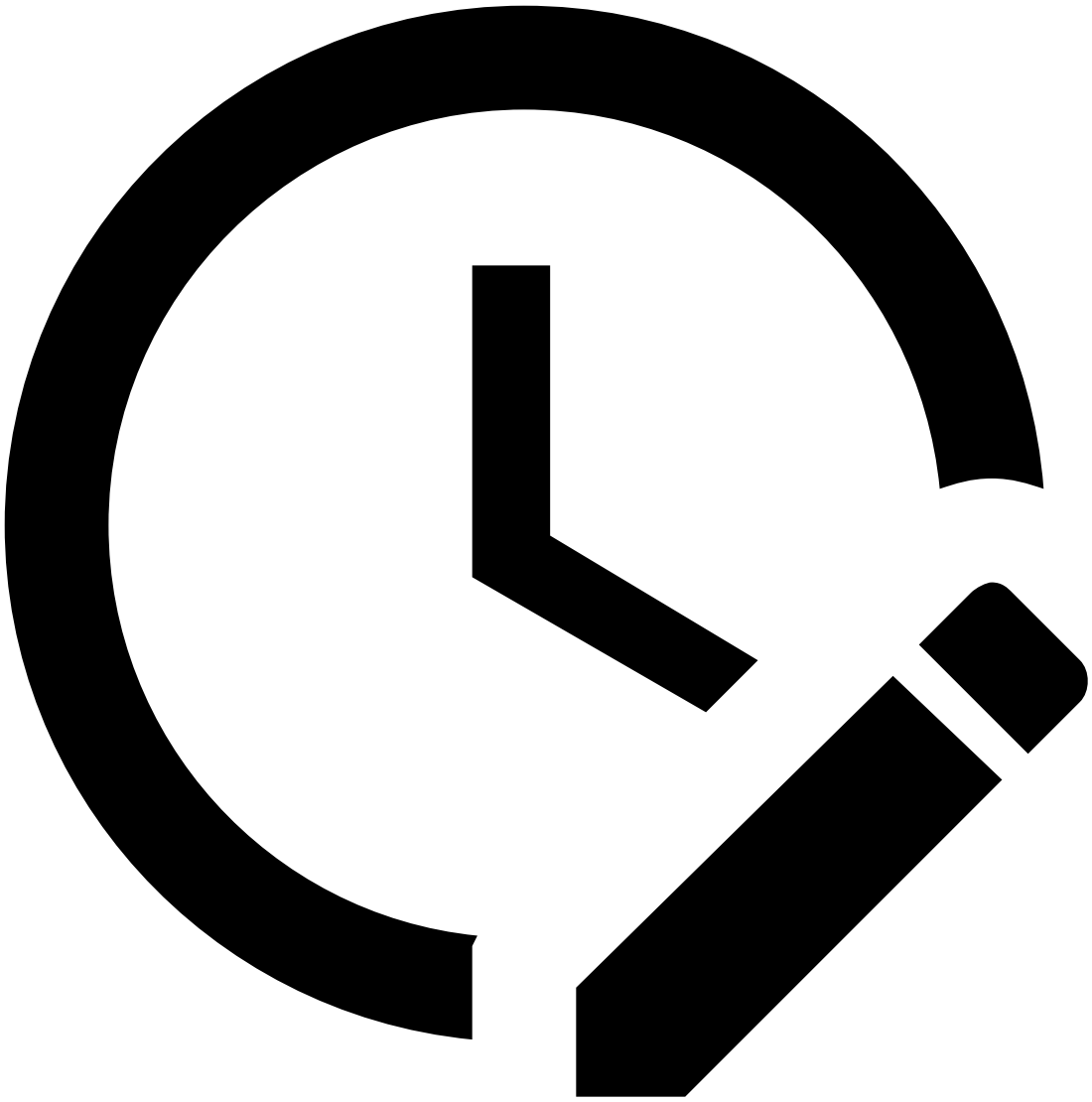
Frontend Shopware:



Schnittstelle

article6.yml (oder article6_overwrite.yml)

```
Artikel_Details:
  Losgroesse:
    winline: C146_LosgreVerkauf
    shop: purchaseSteps
    default: 1
```



2026-03-25

Shop Management Interface

Web-Applikation zur Ausführung und Planung von Synchronisationen.

Rahmendaten

Zur Installation wird die *ess.exe* benötigt. Für diese Datei ist die Config-Datei **management.ini** essenziell. Dieser liegt in dem gleichen Ordner wie die *ess.exe* Datei.

Die Datei wird beim Ausführen von `ess.exe --init` automatisch mit Standardwerten erstellt, sofern sie noch nicht existiert. Vor dem ersten Start müssen in der **management.ini** Datei einige Änderungen vorgenommen werden. Diese Config-Datei ist folgendermaßen aufgebaut.

management.ini

Die Kategorien *ACTIVE* und *SHOPS* bilden den Grundbaustein der zu ansteuerenden Shops. Je nach Wunsch können neue Shops als Keys aufgenommen oder gelöscht werden.

In der *SHOPS* Kategorie wird jedem Shop der Pfad zu seiner **config.ini** zugewiesen. Die Shop-URL wird automatisch aus der jeweiligen config.ini (`[SHOP]` → `url`) ausgelesen. Der Launcher-Pfad (ess.exe bzw. Python) wird automatisch bestimmt.

Beispiel: Einen Shop hinzufügen

```
[ACTIVE]
default = mostwanted

[SHOPS]
mostwanted = C:\Program Files\energy shop-sync\config.ini
```

Die URL des Shops (z.B. `https://shop.example.de/`) wird automatisch aus der hinterlegten config.ini (`[SHOP]` → `url`) ausgelesen und im Interface angezeigt.

Beispiel: Mehrere Shops

```
[ACTIVE]
default = local

[SHOPS]
local = conf\config.ini
mostwanted = C:\Program Files\energy shop-sync\config.ini
```

Weitere Kategorien sind für die Funktionsweise der Syncs notwendig und können angepasst werden.

[TIMER]

Parameter	Defaultwert	Beschreibung
wakeup	1300	Timer zum Abchecken der Sync-Warteschlange in Sekunden-Intervall
overtimekill	85500	Maximale Laufzeit eines Syncs in Sekunden bevor er abgebrochen wird

Synceinstellungen sind Shop-abhängig, d.h. für jeden Shop ist eine Config-Kategorie notwendig bei der der Sektionsname dem Shopnamen in Großbuchstaben entspricht.

Vom obigen Beispiel also:

[MOSTWANTED]

Parameter	Defaultwert /Wert wenn Aktiv	Beschreibung
debuglvl	-d	Debug-Level (-d=ERROR, -dd=WARNING, -ddd=INFO, -dddd=DEBUG)
-wc	-wc	Sync ohne Kategorien. Value frei lassen falls mit Kategorien.
-wa	-wa	Sync ohne Attribute. Value frei lassen falls mit Attributen.
-to	-to wl	Target-Option für den Sync
-we	-we	Sync ohne Eigenschaften. Value frei lassen falls mit Eigenschaften.

Beispiel: Shop-Einstellungen

```
[LOCAL]
debuglvl = -d
-wc = -wc
-wa = -wa
-we = -we
-m =
-ai =
-to = -to wl

[MOSTWANTED]
```

```
debuglvl = -dddd
-wc = -wc
-wa = -wa
-we = -we
-m =
-ai =
-to = -to wl
```

Zusätzlich ist es möglich einzelne Synchronisationsarten auszublenden. Der Defaultwert, wenn der Sync aktiv sein soll: Key = Value. Falls diese ausgeblendet werden sollen, muss der jeweilige Value einer Syncart freigelassen werden.

[SYNCRENAME]

Steuert, welche Sync-Kacheln auf dem Dashboard angezeigt werden und wie sie beschriftet sind. Die Reihenfolge der Zeilen bestimmt zugleich die Default-Reihenfolge der Kacheln, sofern keine `[DASHBOARD]`-Sektion (siehe unten) definiert ist.

Parameter	Defaultwert	Beschreibung
syncProductsAndPrices	Artikel & Preise	Combo-Sync: Artikel und anschließend Preise
syncCustomersAndOrders	Kunden & Bestellungen	Combo-Sync: Kunden und anschließend Bestellungen
syncProducts	Artikel	Der Artikel-Sync
syncProductsAll	Artikel[–all]	Artikel-Sync mit –all Flag
syncPrices	Preise	Synchronisiert die Preise
syncOrders	Bestellungen	Der Bestellungen-Sync
syncCustomers	Kunden	Synchronisiert die Kunden-Daten
syncCategories	Kategorien	Der Kategorien-Sync
syncAttr	Eigenschaften	Synchronisiert die Eigenschaften
syncCross	Cross-Artikel	Synchronisiert die Cross-Selling-Artikel
syncBestand	Bestand	Synchronisiert die Bestände
syncProperties	Properties	Synchronisiert die Properties
syncImages	Bilder	Synchronisiert die Bilder
syncDownloads	Downloads	Synchronisiert die Downloads
syncTranslations	Übersetzungen	Synchronisiert die Übersetzungen
syncFilteredCustomers	Kunden SW-WL	Synchronisiert gefilterte Kundendaten

Parameter	Defaultwert	Beschreibung
syncOrderStatus	Bestellstatus	Synchronisiert den Bestellstatus
syncShipmenttype	Versandart	Synchronisiert die Versandart
syncAll	Alles	Synchronisiert alles

Beispiel: Sync-Bezeichnungen

```
[SYNCRENAME]
# "" = inaktiv. Reihenfolge dieser Zeilen = Default-Reihenfolge der Kacheln.
syncProductsAndPrices = Artikel & Preise
syncCustomersAndOrders = Kunden & Bestellungen
syncProducts = Artikel
syncProductsAll = Artikel[--all]
syncPrices = Preise
syncOrders = Bestellungen
syncCustomers = Kunden
syncCategories = Kategorien
syncAttr = Eigenschaften
syncCross = Cross-Artikel
syncBestand = Bestand
syncProperties = Properties
syncImages = Bilder
syncDownloads = Downloads
syncTranslations = Übersetzungen
syncFilteredCustomers = Kunden SW-WL
syncOrderStatus = Bestellstatus
syncShipmenttype = Versandart
syncAll = Alles
```

[DASHBOARD] und [DASHBOARDGROUP:*]

Optional. Erlaubt es, die Kacheln auf dem Dashboard in mehrere Bereiche zu gruppieren und die Reihenfolge frei festzulegen. Die erste Gruppe wird als *Featured* gerendert (größere Kacheln, dezent hervorgehoben). Fehlt die Sektion, werden alle aktiven Kacheln aus `[SYNCRENAME]` als eine flache Liste in INI-Reihenfolge angezeigt.

Sektion / Parameter	Beschreibung
`[DASHBOARD] groups`	Komma-getrennte Liste der Gruppen-IDs in Anzeigereihenfolge. Jede ID verweist auf eine Sektion `[DASHBOARDGROUP:<id>]`.
`[DASHBOARDGROUP:<id>] title`	Überschrift über der Gruppe. Leer = keine Überschrift (z. B. für die Featured-Zeile). Nur Gruppen mit Titel sind aufklappbar.

Sektion / Parameter	Beschreibung
`[DASHBOARDGROUP:<id>]` `keys`	Komma-getrennte Liste der Sync-Keys in der gewünschten Reihenfolge. * als alleiniger oder zusätzlicher Eintrag steht für „alle übrigen aktiven Keys aus `[SYNCRENAME]`, die nicht von einer **anderen** Gruppe explizit beansprucht werden". Unbekannte oder inaktive Keys werden ignoriert. Bei mehreren *-Gruppen gewinnt die erste in `groups`-Reihenfolge.
`[DASHBOARDGROUP:<id>]` `collapsed`	Optional, Default `false`. Bei `true` wird die Gruppe initial zugeklappt dargestellt; ein Klick auf den Gruppen-Header klappt sie auf.

Beispiel: Featured-Zeile + Restbereich

```
[DASHBOARD]
groups = featured, weitere

[DASHBOARDGROUP:featured]
title =
keys = syncProductsAndPrices, syncOrders

[DASHBOARDGROUP:weitere]
title = Weitere Synchronisationen
keys = *
collapsed = false
```

Mit dieser Konfiguration erscheinen oben in größeren Kacheln "Artikel & Preise" und "Bestellungen", darunter durch eine horizontale Linie abgegrenzt alle übrigen Sync-Kacheln in `[SYNCRENAME]`-Reihenfolge. Mit `collapsed = true` lässt sich die Restgruppe initial zuklappen.

Beispiel: Drei Gruppen mit Wildcard in der Mitte

```
[DASHBOARD]
groups = featured, weitere, footer

[DASHBOARDGROUP:featured]
title =
keys = syncProductsAndPrices, syncOrders

[DASHBOARDGROUP:weitere]
title = Weitere Synchronisationen
keys = *
```

```
[DASHBOARDGROUP:footer]
title = Bestand & Bilder
keys = syncBestand, syncImages
```

`syncBestand` und `syncImages` landen verlässlich im Footer-Block, weil die Wildcard erst danach ausgewertet wird und nur Tiles aufnimmt, die nirgendwo explizit beansprucht sind.

Vollständige Beispiel management.ini

```
[ACTIVE]
default = local

[SHOPS]
local = conf\config.ini
mostwanted = C:\Program Files\energy shop-sync\config.ini

[TIMER]
# in Sekunden
wakeup = 1300
overtimekill = 85500

[LOCAL]
debuglvl = -d
-wc = -wc
-wa = -wa
-we = -we
-m =
-ai =
-to = -to wl

[MOSTWANTED]
debuglvl = -dddd
-wc = -wc
-wa = -wa
-we = -we
-m =
-ai =
-to = -to wl

[SYNCRENAME]
# "" = inaktiv. Reihenfolge dieser Zeilen = Default-Reihenfolge der Kacheln.
syncProductsAndPrices = Artikel & Preise
syncCustomersAndOrders = Kunden & Bestellungen
syncProducts = Artikel
```

```
syncProductsAll = Artikel[--all]
syncPrices = Preise
syncOrders = Bestellungen
syncCustomers = Kunden
syncCategories = Kategorien
syncAttr = Eigenschaften
syncCross = Cross-Artikel
syncBestand = Bestand
syncProperties = Properties
syncImages = Bilder
syncDownloads = Downloads
syncTranslations = Übersetzungen
syncFilteredCustomers = Kunden SW-WL
syncOrderStatus = Bestellstatus
syncShipmenttype = Versandart
syncAll = Alles

[DASHBOARD]
groups = featured, weitere

[DASHBOARDGROUP:featured]
title =
keys = syncProductsAndPrices, syncOrders

[DASHBOARDGROUP:weitere]
title = Weitere Synchronisationen
keys = *
collapsed = false
```

Installation

Die management.ini wird beim Ausführen von `ess.exe --init` automatisch mit Standardwerten erstellt. Anschließend müssen die Shops manuell hinzugefügt werden. Dies erfolgt dadurch, dass jeder Shop in der Kategorie [SHOPS] mit seinem Namen als Key und dem Pfad zur config.ini als Value aufgenommen wird. Die Shop-URL und der Launcher-Pfad werden automatisch bestimmt.

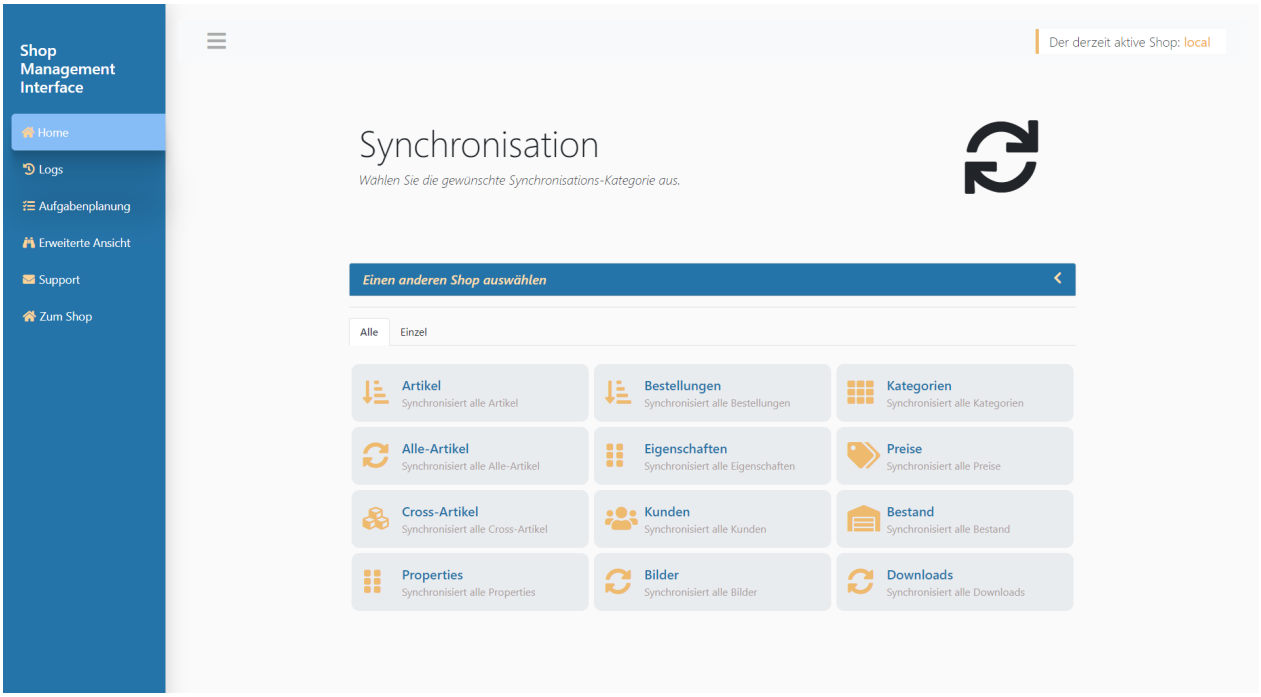
Ist die Management.ini entsprechend erstellt worden, reicht die Ausführung von *ess.exe manage runserver* aus. Die Web-Applikation läuft standardmäßig auf dem Hostnamen mit dem Port **8000** und kann über jeden beliebigen Web-Browser erreicht werden. Der Port kann beim Start angepasst werden (z.B. `ess.exe manage runserver 0.0.0.0:9000`).

Bspw. : *nerdpool:8000*

Ausführung von Synchronisationen

Das Management-Interface bietet 2 verschiedene Möglichkeiten Synchronisationen auszführen bzw. 3, letzteres ist aber über die erweiterte Ansicht erreichbar.

1. Live-Sync



Live-Syncs sind über die Startseite des Interfaces erreichbar. Mit einem Klick auf die Schaltflächen lassen sich einzelne Syncs startet. Wird eine Sync gestartet erscheint dazu ein Live-Log der solange bleibt bis man eine andere Seite lädt. Zusätzlich kann der Sync über den Progressbar oberhalb der Seite verfolgt werden. Der Progressbar wird auf jeder Seite neugeladen.

2. Geplante Syncs (Scheduler)

Über den Scheduler im erweiterten Bereich können Synchronisationen zeitlich geplant werden. Die geplanten Jobs werden über APScheduler in einer separaten SQLite-Datenbank (`apevents.db`) persistiert.

Scheduler verwalten

Job erstellen:

1. Im Menü auf “Scheduler” klicken
2. Bezeichnung für den Job vergeben
3. Sync-Art auswählen (z.B. Artikel, Bestellungen)
4. Frequenz wählen:
    - **Einmalig**: Einmaliger Sync zu einem bestimmten Zeitpunkt
    - **Minütlich/Stündlich/Täglich**: Wiederkehrender Sync im gewählten Intervall

    - **Wöchentlich**: Sync an bestimmten Wochentagen
5. Optional: **Parameter** als freie CLI-Argumente angeben (z. B. `-a` für Vollsync, `-wc` ohne Kategorien). Mehrere Argumente werden per Leerzeichen getrennt, Quoting wird unterstützt. Eine Übersicht aller Flags liefert `ess.exe --help`.
6. Startzeit festlegen und Job speichern
7. Optional: **Verpasste Aufgaben nachholen** ankreuzen, wenn die Aufgabe auch dann noch laufen soll, wenn ihr planmäßiger Zeitpunkt während einer Down-Zeit (z. B. Wochenend-Reboot) verpasst wurde. Mehrfach verpasste Ausführungen werden zu einer einzelnen zusammengefasst (`coalesce`). Ohne diese Option verwirft APScheduler Ausführungen, die mehr als 15 Minuten zu spät dran sind.

Job verwalten:

- **Pausieren**: Einen laufenden Job vorübergehend anhalten
- **Fortsetzen**: Einen pausierten Job wieder aktivieren
- **Bearbeiten**: Bezeichnung, Zeitplan, Parameter oder die Catchup-Option eines Jobs ändern. Im Bearbeiten-Dialog stehen die Intervalle Einmal / Minütlich / Stündlich / Täglich zur Auswahl; wöchentliche Cron-Jobs werden grundsätzlich neu angelegt statt editiert.
- **Löschen**: Einen Job dauerhaft entfernen

Alle geplanten Jobs werden persistent gespeichert und überleben einen Neustart des Management Interfaces.

Standard-Zeitplan laden

Über den Button **“Standard-Zeitplan laden”** im Kopfbereich der Aufgabenplan-Tabelle lassen sich drei typische Hintergrundaufgaben für den aktuellen Shop mit einem Klick anlegen:

Aufgabe	Sync-Art	Parameter	Zeitplan	Catchup
Artikel & Preise Mo-Fr 01:00	`syncProductsAndPrices`	–	Mo–Fr um 01:00	–
Artikel & Preise So 01:00	`syncProductsAndPrices`	`-a`	Sonntag um 01:00	ja
Bestellungen Mo-Fr 06-18 Uhr alle 15 Min	`syncOrders`	–	Mo–Fr zwischen 06:00 und 18:00 alle 15 Minuten	–

Unter der Woche läuft ein kombinierter Artikel- und Preissync inkrementell (nur Änderungen). Der Sonntags-Vollsync läuft mit `-a` (Artikel & Preise ohne Berücksichtigung des Änderungsdatums), damit übers Wochenende alle Daten garantiert synchron sind. Bei einem Wochenend-Reboot wird der Vollsync dank `Catchup` beim nächsten Start des Dienstes nachgeholt.

Bereits vorhandene Aufgaben mit identischem Namen bleiben unangetastet – das erneute Laden des Presets fügt lediglich fehlende Jobs hinzu. Soll ein bestehender Sonntags-Job auf `-a` umgestellt werden, muss er manuell gelöscht und das Preset erneut geladen werden.

3. Einzelsync

Über die erweiterte Ansicht kann ein einzelner Artikel per Artikelnummer synchronisiert werden.

Warteschlange

Wird ein Sync gestartet während bereits ein anderer Sync für denselben Shop läuft, wird der neue Sync automatisch in eine Warteschlange eingereiht. Sobald der laufende Sync abgeschlossen ist, wird der nächste Sync aus der Warteschlange automatisch gestartet.

> **Ausnahme: Bestellungen**
>
> Der Bestellungssync (syncOrders) kann unabhängig von anderen Syncs parallel laufen, da er keine Artikeldaten verändert.

Sync-Fortschritt

Der Fortschritt einer laufenden Synchronisation wird in Echtzeit über einen Progressbar angezeigt. Der ProcessManager parst die Stdout-Ausgabe des Sync-Prozesses im Format:

`PROGRESS: {process}: {step} von {max} - {nachricht}`

Der Fortschritt wird in zwei Phasen unterteilt:

- **Vorbereitung** (0-50%): Daten werden aus der WinLine gelesen
- **Übertragung** (50-100%): Daten werden an den Shop übermittelt

Der Status wird über AJAX-Polling (`/api/sync-progress/{shop}`) abgefragt und auf jeder Seite im Progressbar angezeigt.

Mögliche Status: `running` → `finished` / `error` / `overtime`

Ein Overtime-Schutz beendet Syncs automatisch nach der in `[TIMER]` konfigurierten `overtimekill`-Zeit (Standard: 85500 Sekunden).

Sync-Fehler-Management

Unter `/{shop}/errors` können Sync-Fehler eingesehen und verwaltet werden.

Jeder Fehler enthält folgende Informationen:

Feld	Beschreibung
Shop-URL	Betroffener Shop
Sync-Typ	Art der Synchronisation
Entity-Typ	Betroffener Datentyp
Entity-ID	ID des betroffenen Datensatzes
Fehlermeldung	Beschreibung des Fehlers
Status	Aufgelöst / Offen

Funktionen:

- Filtern nach Shop-URL, Entity-Typ und Auflösungsstatus
- Bulk-Aktionen: Fehler auflösen oder wieder öffnen
- Automatische Auflösung bei Full-Sync (nicht bei Differential-Sync)

Scheduler-Events-Dashboard

Unter `/advanced/apdashboard` kann der Status aller geplanten Scheduler-Events eingesehen werden.

Angezeigte Events:

- Hinzugefügt
- Entfernt
- Geändert
- Erfolgreich ausgeführt
- Fehlgeschlagen
- Verpasst

Die Events werden in einer separaten SQLite-Datenbank (`apevents.db`) gespeichert.

Konfigurationseditor

Über den erweiterten Bereich können Konfigurationsdateien direkt im Browser bearbeitet werden:

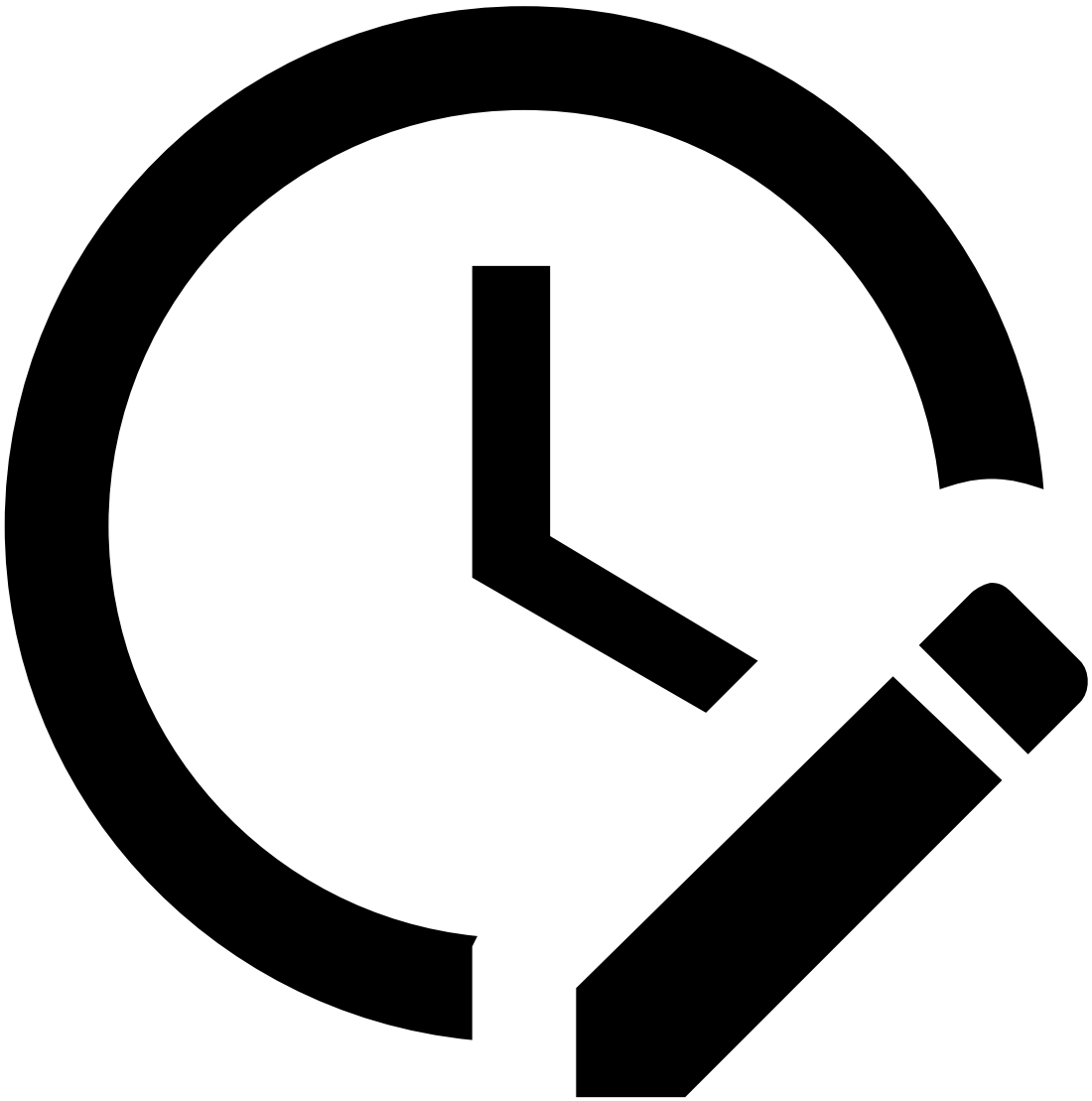
Route	Beschreibung
`/advanced/management/`	management.ini bearbeiten
`/advanced/settings/{shop}`	config.ini eines Shops bearbeiten
`/advanced/mappings/{shop}`	Mapping-Dateien eines Shops bearbeiten

Zusätzlich können Debug-Level und Sync-Timestamps pro Shop eingestellt werden.

Log-Ansicht

Logdateien können direkt im Browser eingesehen und heruntergeladen werden:

Route	Beschreibung
`/logs`	Liste aller verfügbaren Logdateien
`/logs/view/{file}`	Logdatei im Browser anzeigen
`/logs/download/{file}`	Logdatei herunterladen



Mehrsprachigkeit

Überblick

- Die Schnittstelle ermöglicht die Übertragung von Artikelinformationen in mehreren Sprachen an Shopware
- Übersetzbare Felder: Artikelbezeichnung, Artikelbeschreibung, Kategorien und Eigenschaften (Gruppenbezeichnungen)
- Jede Sprache wird einem Shopware Sales Channel zugeordnet
- Die Übersetzungen werden in der WinLine über die Textfelder und den Eigenschaftenstamm gepflegt

Workflow

1. In Shopware einen Sales Channel mit der gewünschten Sprache anlegen
2. In der config.ini die Sprachen konfigurieren (Sektion `[LANGUAGES]` und je Sprache eine eigene Sektion)
3. Übersetzungen in der WinLine pflegen:
    - Artikelbezeichnung und -beschreibung über Text ½ oder Zusatzfelder
    - Kategorien im Eigenschaftenstamm in der entsprechenden Sprachspalte
    - Eigenschaften (Gruppenbezeichnungen) ebenfalls in der jeweiligen Sprachspalte
4. Mapping für die Übersetzungsfelder anpassen (falls abweichende Felder verwendet werden)
5. Übersetzungssync starten (**syncTranslations**)
6. Ergebnis im Shop prüfen: Sprache über Sales Channel/Sprachumschalter wechseln

Sync-Befehl

Der Übersetzungssync wird über den Befehl **syncTranslations** (Übersetzungen) im Management Interface ausgeführt.

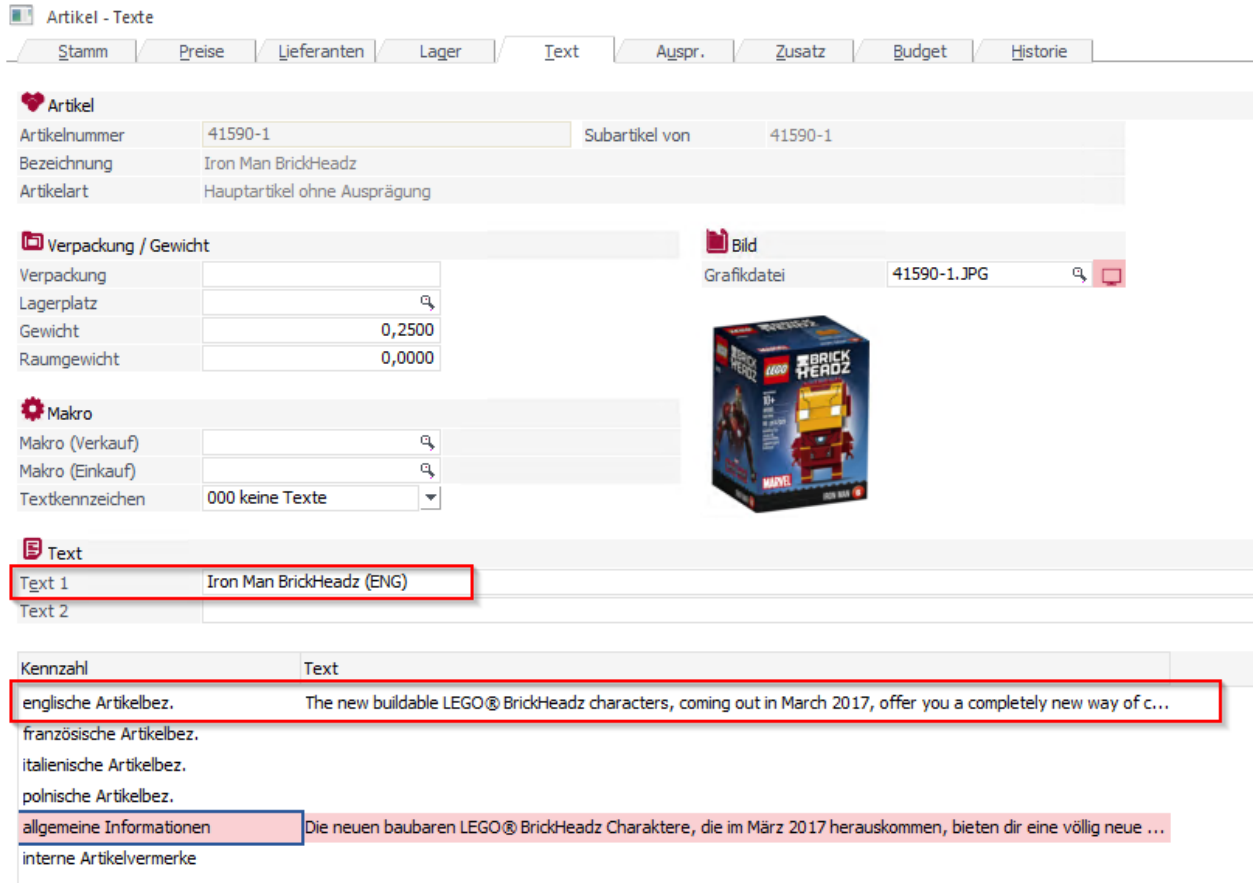
Eigenschaftswerte

Die Werte der Eigenschaften (nicht die Gruppenbezeichnungen) können derzeit nicht automatisch übersetzt werden. Diese müssen im Eigenschaftenstamm von Shopware manuell hinterlegt werden.

Die Schnittstelle ermöglicht die Übertragung der Artikelbezeichnungen, -beschreibungen, -kategorien und -eigenschaften, die in der WinLine gepflegt sind.

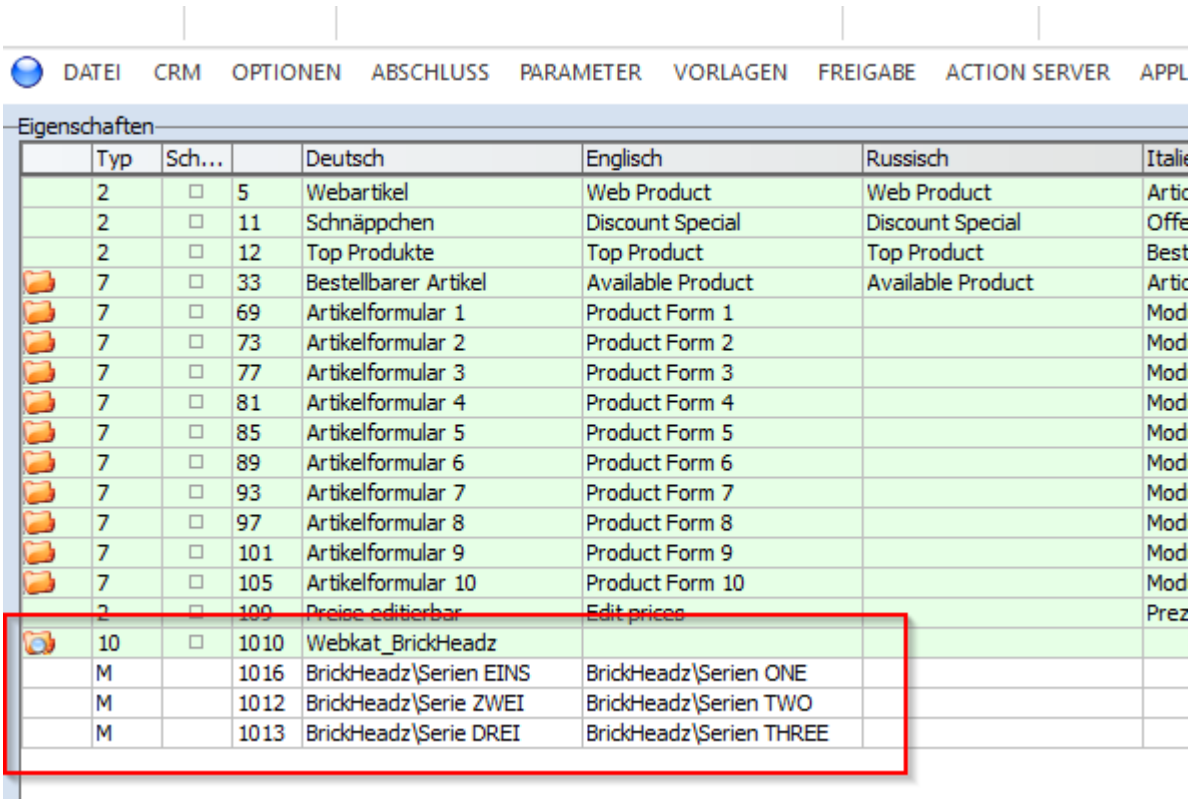
Bezeichnung und Beschreibung

Die Artikelbezeichnung und -beschreibung können über Text ½ und Textfelder für die Beschreibung in weiteren Sprachen hinterlegt werden. Falls die vorhandenen Felder nicht ausreichen, können auch Zusatzfelder zum Einsatz kommen. Diese werden über ein angepasstes Mapping angesprochen.



Kategorie und Eigenschaften

Die Artikelkategorien werden im Eigenschaftsstamm in der Spalte “English” angelegt.



	Typ	Sch...		Deutsch	Englisch	Russisch	Itali
	2	□	5	Webartikel	Web Product	Web Product	Artic
	2	□	11	Schnäppchen	Discount Special	Discount Special	Offe
	2	□	12	Top Produkte	Top Product	Top Product	Best
	7	□	33	Bestellbarer Artikel	Available Product	Available Product	Artic
	7	□	69	Artikelformular 1	Product Form 1		Mod
	7	□	73	Artikelformular 2	Product Form 2		Mod
	7	□	77	Artikelformular 3	Product Form 3		Mod
	7	□	81	Artikelformular 4	Product Form 4		Mod
	7	□	85	Artikelformular 5	Product Form 5		Mod
	7	□	89	Artikelformular 6	Product Form 6		Mod
	7	□	93	Artikelformular 7	Product Form 7		Mod
	7	□	97	Artikelformular 8	Product Form 8		Mod
	7	□	101	Artikelformular 9	Product Form 9		Mod
	7	□	105	Artikelformular 10	Product Form 10		Mod
	2	□	109	Preise editierbar	Edit prices		Prez
	10	□	1010	Webkat_BrickHeadz			
	M		1016	BrickHeadz\Serien EINS	BrickHeadz\Serien ONE		
	M		1012	BrickHeadz\Serie ZWEI	BrickHeadz\Serien TWO		
	M		1013	BrickHeadz\Serie DREI	BrickHeadz\Serien THREE		

Falls eine große Anzahl von Kategorien angelegt werden muss, kann das folgende Makro verwendet werden:

AUTOANLAGE_EIGENSCHAFTEN_ENG.MMR

Diese Version ist für die englische Sprache gemacht. Für weitere Sprachen müsste im Makro die Spaltennummer angepasst werden.

Eigenschaften

Auch die Eigenschaften werden gleichermaßen in der jeweiligen Spalte übersetzt. Die Werte der Eigenschaften können in ihrer übersetzten Form nicht an Shopware übertragen werden. Diese müssen im Eigenschaftenstamm von Shopware händisch hinterlegt werden.

Shopware

In Shopware 6 muss ein entsprechender Sales Channel mit der gewünschten Sprache angelegt werden.

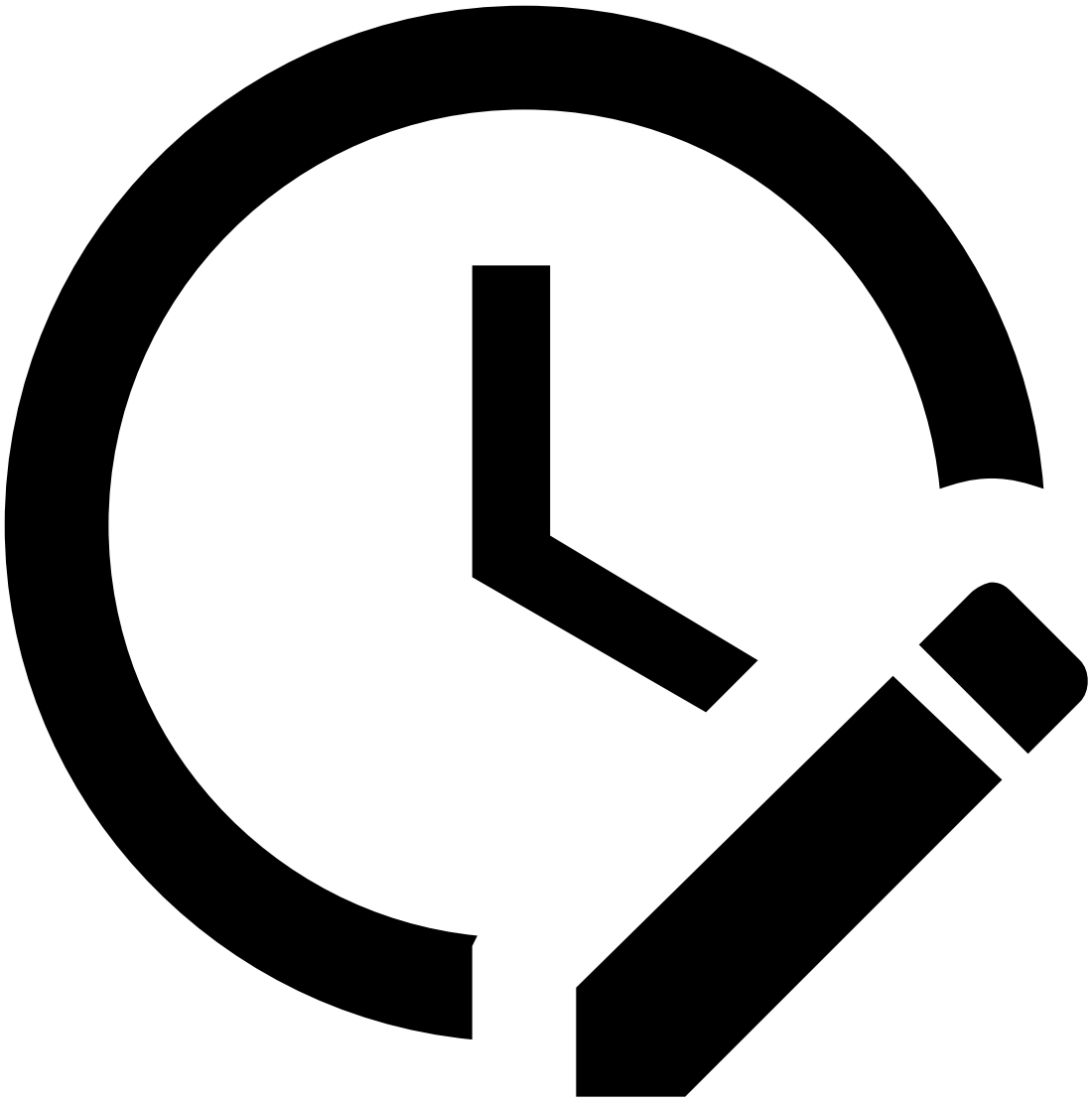
Config

```
[LANGUAGES]
languages = ["Deutsch", "English"]
```

```
[Deutsch]
salesChannelName = Storefront
descriptionfield = 5
position = 1
separate_categorie = False
primary_shop = True
local_code = de-DE
encoding = utf-8

[English]
salesChannelName = English Store
descriptionfield = 1
position = 2
separate_categorie = False
primary_shop = False
local_code = en-GB
encoding = utf-8
```

Mapping

```
"Translations":{
        "English":{
            "Article_translation":{
                "Bezeichnung":{
                    "winline": "C073_Langtext1",
                    "shop": "name"
                },
                "Beschreibung":{
                    "winline": "C080_Notiz1",
                    "shop": "descriptionLong"
                }
            },
            "Property_translation":{
                "Bezeichnung":{
                    "winline": "C011_Text2",
                    "shop": "optionName"
                }
            }
```



2026-03-27

Mindestbestellmenge / Maximalbestellmenge

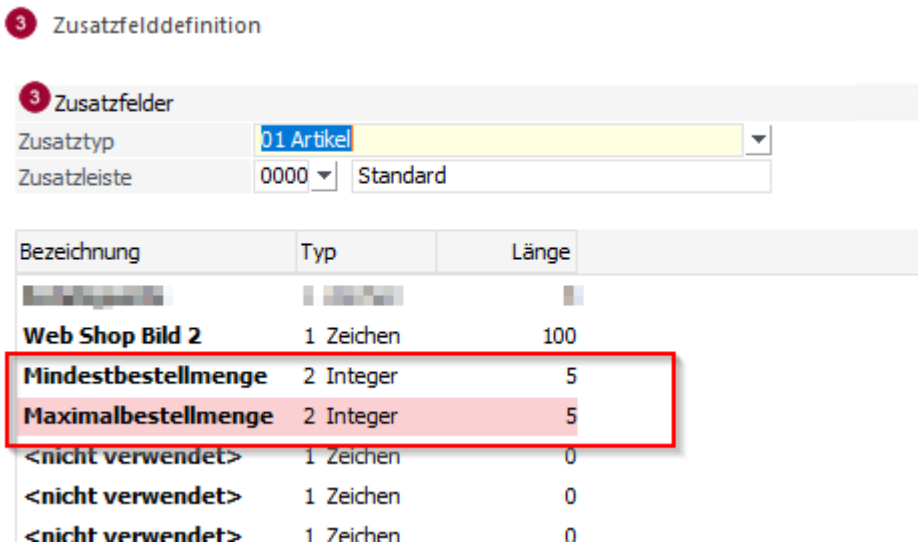
Überblick

Über die Mindest- und Maximalbestellmenge kann gesteuert werden, welche Mengen ein Kunde im Shop bestellen darf. Die Werte werden über frei definierbare Zusatzfelder in der WinLine gepflegt und beim Artikelsync an Shopware übertragen. Ein leeres Feld bei der Mindestbestellmenge entspricht dem Standardwert 1. Ein leeres Feld bei der Maximalbestellmenge bedeutet keine Beschränkung.

WinLine

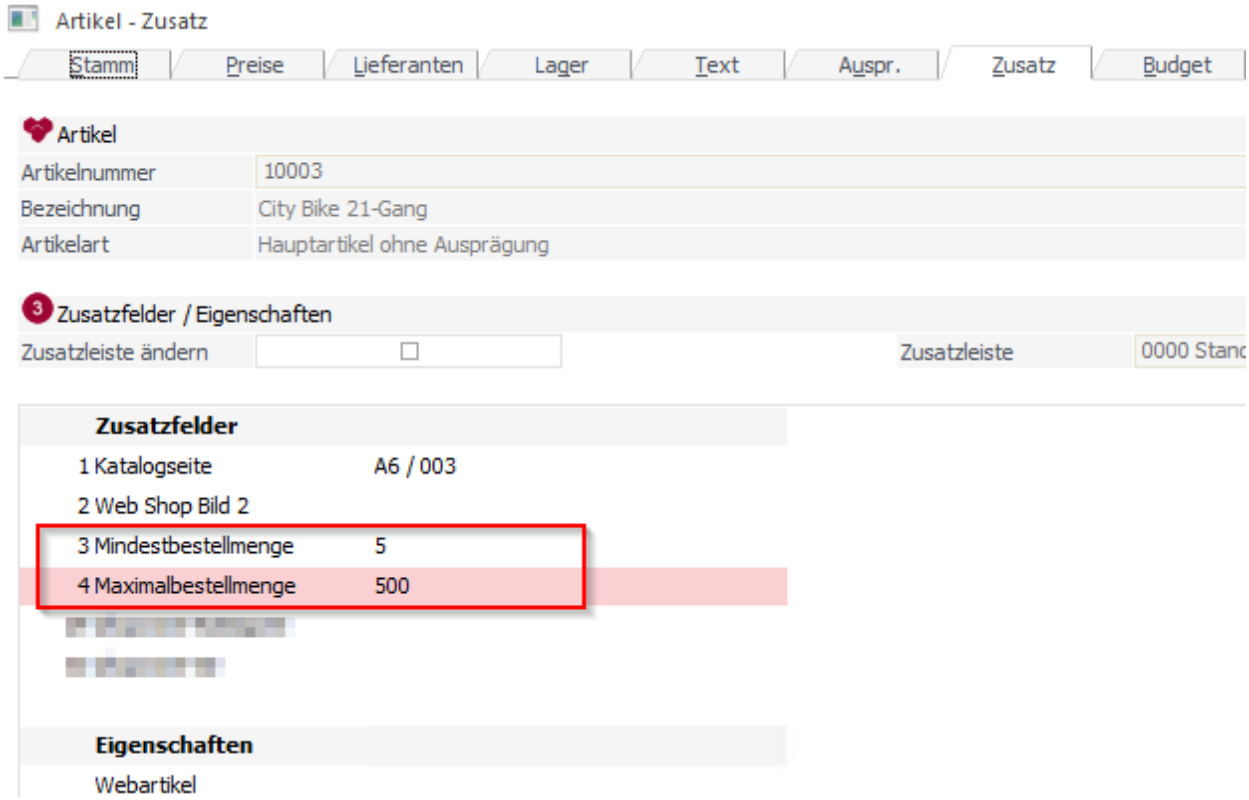
Einrichtung

Die Mindestbestellmenge wird über ein frei definierbares Zusatzfeld bestimmt. Dieses muss in der WinLine mit dem Typ Integer angelegt werden.



Verwendung

Im Artikelstamm kann die Mindestbestellmenge über das konfigurierte Zusatzfeld hinterlegt werden. Ein leeres Feld bedeutet, dass die Mindestbestellmenge 1 ist.



Shop

Im Shop muss mindestens die eingestellte Menge bestellt werden.

Schnittstelle

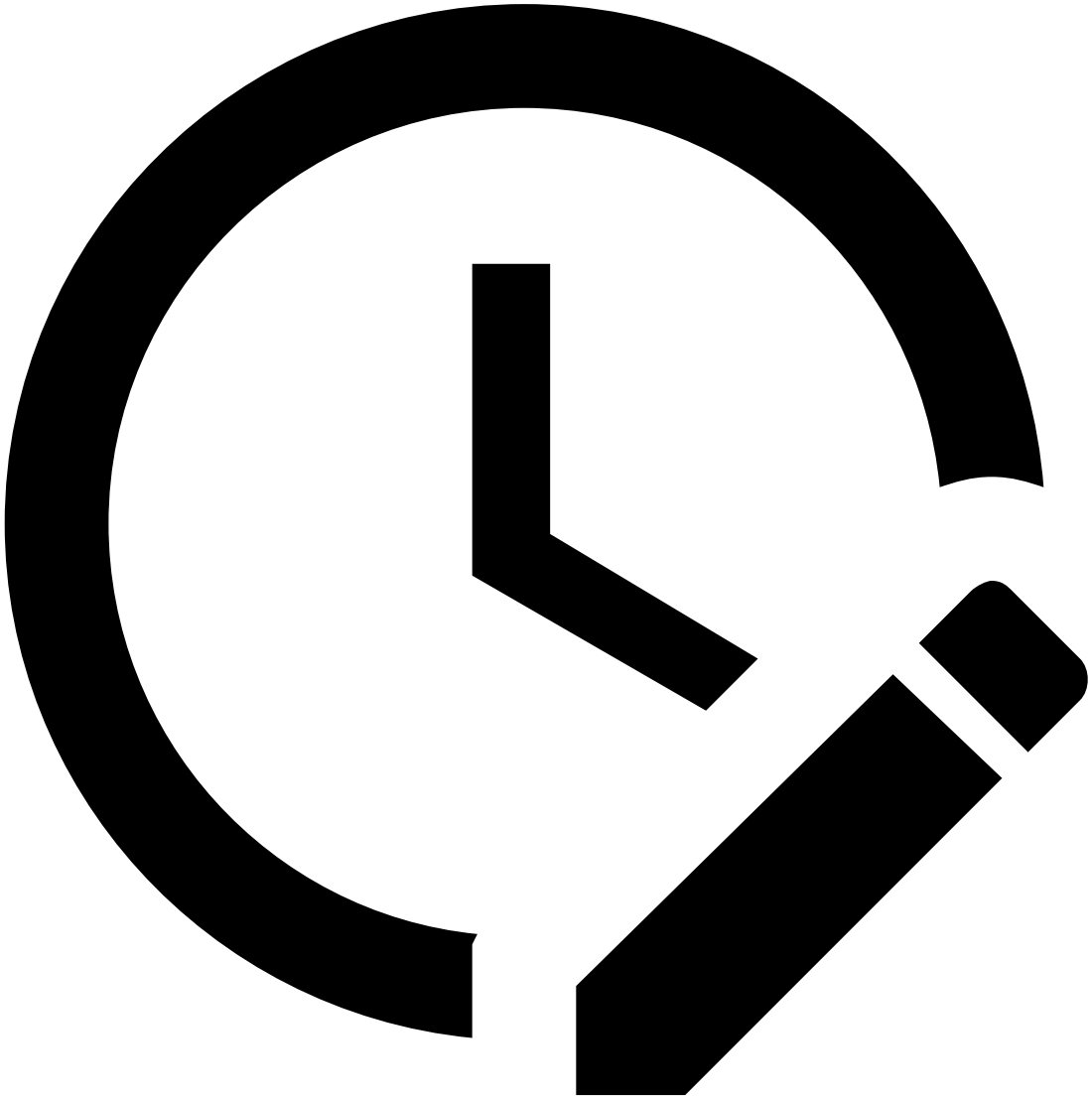
article6.yml (oder article6_overwrite.yml)

```
Artikel_Details:
  Mindesabnahme:
    winline: C228_Zusatzfeld28
    shop: minPurchase
    default: 1
  Maximalabnahme:
    winline: C229_Zusatzfeld29
    shop: maxPurchase
    default: 0
```

Kundenspezifische Feldnummern

Die Zusatzfeldnummern (hier C228/C229) sind kundenspezifisch und können in der Mapping-Datei angepasst werden. Das Standard-Mapping verwendet diese Felder — in älteren Installationen können andere Felder (z.B. C203/C204) konfiguriert sein.

Die Mindestbestellmenge kann natürlich über das Mapping auch an die Losgröße VK gebunden werden.



Preise

Überblick

Der Preissync überträgt Artikelpreise von der WinLine an Shopware. Die Synchronisation erfolgt **einseitig** in Richtung WinLine → Shopware.

Folgende Preisarten werden übermittelt:

- Allgemeine Verkaufspreise
- Kundengruppenpreise
- Kundenindividuelle Preise
- Staffelpreise (Mengenrabatte)
- Zeitlich begrenzte Preise (mit Gültigkeitszeitraum)

Der Preissync kann sowohl separat als auch als Teil des Gesamtsyncs ausgeführt werden.

Workflow

1. **Preise in der WinLine pflegen** – Preislisten, Staffelpreise und kundenspezifische Preise anlegen bzw. aktualisieren
2. **Preissync starten** – über das Management Interface oder einen geplanten Job
3. Die Schnittstelle **liest alle relevanten Preise** aus der WinLine
4. **Allgemeine Preise** werden als Standardpreise im Shop hinterlegt
5. **Kundenindividuelle Preise** werden über Shopware-Regeln (Rules) mit dem jeweiligen Kundenkonto verknüpft
6. **Kundengruppenpreise** werden der entsprechenden Kundengruppe zugeordnet
7. **Ergebnis im Shop prüfen**: Preise ohne Login (allgemeiner VK) und mit Kundenlogin (individueller Preis)

Verwendung

> **Sync-Befehl**
>
> Der Preissync wird über den Befehl **syncPrices** im Management Interface oder als geplanter Job ausgeführt.

- Nach Preisänderungen in der WinLine den Preissync starten
- Kundenspezifische Preise: Erst sicherstellen, dass der Kunde im Shop existiert und eine Kundennummer hat
- Bei der Ersteinrichtung oder nach Änderungen an der Preiskonfiguration empfiehlt sich ein Vollsync

Preisliste

Die Schnittstelle kann beliebig viele Preislisten übertragen. Hierfür muss in der Konfiguration jede Preislistennummer hinterlegt werden (mit Komma getrennt). Die Preisliste an erster Stelle erhält die niedrigste Priorität und ist im Shop als Standardpreisliste hinterlegt.

Alle anderen Preislisten werden im Shop ausgewählt, indem dem Kundenkonto das Tag “Preisliste X” hinzugefügt wird.

```
[WL]
; interne WinLine Nummer für die Webshop Preisliste
preislistennr = 5, 6
```

Generelles zur Preisübermittlung

Die Schnittstelle übermittelt die Preise so wie sie in der WinLine hinterlegt sind. Das bedeutet, es werden Staffeln, Rabatte, kundenspezifische Preise, kundengruppen Preise, allgemeine Preise und die Rabattmatrix berücksichtigt.

Kundenindividuelle Preise in Shopware 6

In Shopware 6 werden kundenindividuelle Preise über **Regeln (Rules)** realisiert. Für jeden Kunden mit individuellen Preisen wird eine Regel mit der Kundennummer erstellt. Die Preise werden dann als “Erweiterte Preise” mit dieser Regel verknüpft.

Darüber können auch datumsbezogene, rabattierte oder mengenspezifische Preise übermittelt werden.

Rabattmatrix

Bei der Rabattmatrix, im Fall eines allgemeinen Preises mit fixer Rabattspalte vom Artikel oder Rabattspalte, wird für jeden Shopware Account, der ein WinLine-Konto besitzt, eine extra Preisregel in den Erweiterten Preisen angelegt, außer für das Konto existiert ein kundengruppen oder kundenspezifischer Preis.

Rabattmatrix zusammenfassen

Für die Rabattmatrix kann die Schnittstelle alle Preise in der Preisart `1 - Verkaufspreis` zusammenfassen, um Preiseinträge im Shop zu reduzieren und in diesem die Performance zu verbessern. Weicht die Preisart einer Preiszeile in der WinLine ab, werden die Preise im Shop pro Kunden erstellt. Dieses Feature wird über die Featureflag `SUMMARIZE_ACCOUNT_MATRIX = True` aktiviert.

Kundengruppen Preise

Kundengruppenpreise werden dem Kunden in Shopware zugeordnet. Bei der WinLine-Anzeige darauf achten, dass die Ansicht vollständig ist!

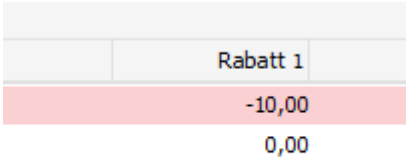
Brutto/Netto-Flag Preisliste

Logik für Brutto oder Netto Berechnung wird durch das Brutto/Netto-Flag aus Preisliste eines Preislisteneintrags ausgelesen.

Pseudo- /rabattierter Preis

Der in der WinLine hinterlegte Rabatt 1 wird als errechneter Pseudopreis an Shopware übergeben.

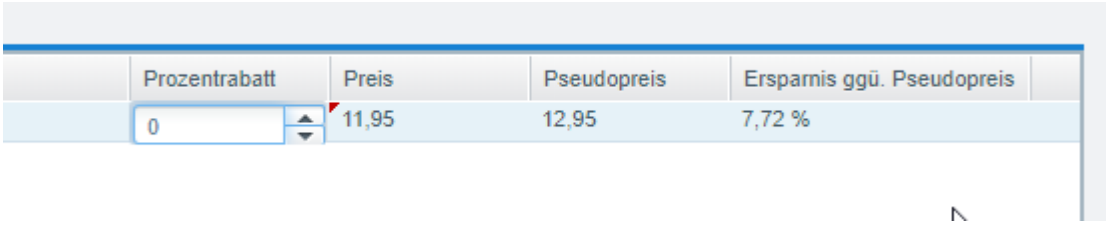
WinLine Rabatt 1



Screenshots aus einer älteren Shopware-Version

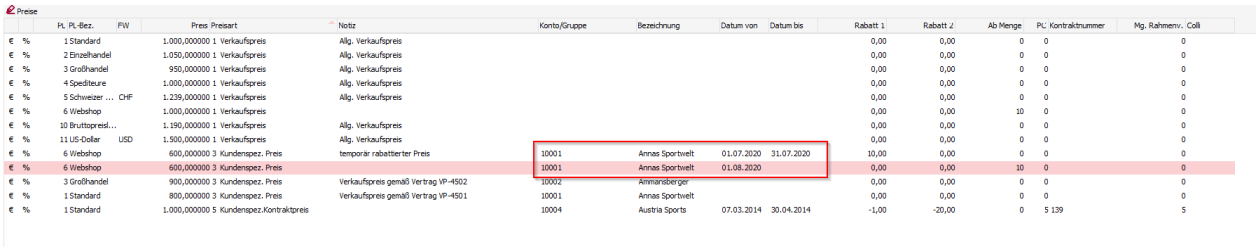
Shopware Frontend

Shopware Backend



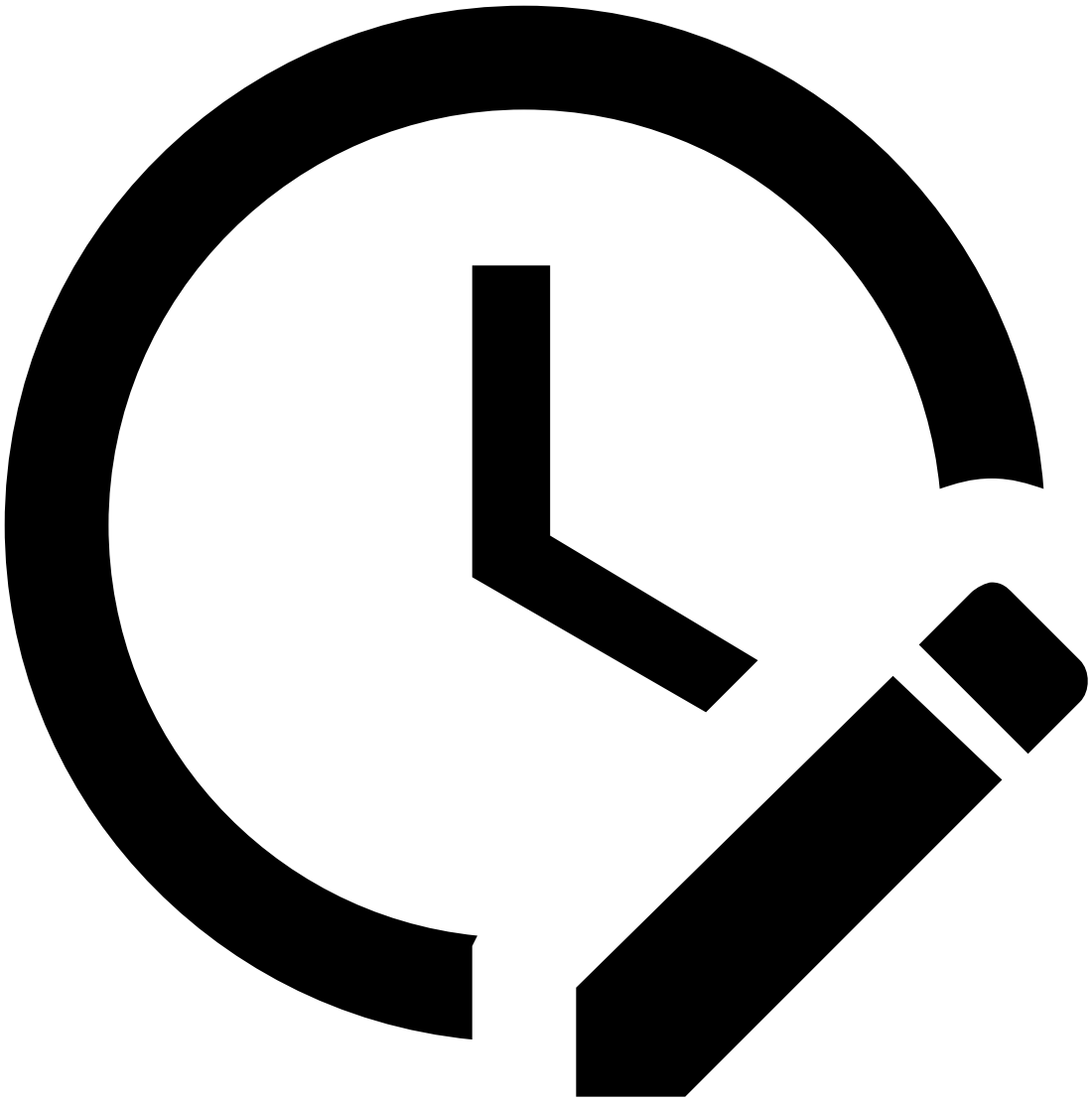
Preise mit überschneidender Gültigkeit

Wenn ein Preis zusätzlich zum existierenden Preis (kundenindividuell, gruppenspezifisch, allgemein) mit “Datum von” und “Datum bis” belegt wird, werden für die Zeit der Gültigkeit beide Preise an Shopware übergeben. Um dies zu umgehen, muss der für den Zeitraum invalide Preis mit einen “Datum von” belegt werden. Nur auf diese Weise kann garantiert werden, dass nur der gültige Preis übergeben wird.



Preisduplikate entfernen

Über das Feature `REMOVE_PRICE_DUPLICATES = True` können Preisduplikate automatisch entfernt werden, z.B. bei Überschneidungen in der Gültigkeit oder wenn kein Enddatum für den Preis gesetzt wurde. Es wird im Duplikatfall der Preis mit dem größeren Von-Datum genommen. Ist das Von-Datum identisch wird der höhere Preis übermittelt.



2026-03-27

Statusupdates

Überblick

Die Schnittstelle importiert Bestellungen aus dem Shopware-Shop in die WinLine und aktualisiert den Bestellstatus.

- **Sync-Richtung Bestellimport:** Shopware → WinLine
- **Sendungsverfolgung:** Kann in beide Richtungen übermittelt werden (konfigurierbar über `trackingNrToWinline`)
- **Importsteuerung:** Der Bestellimport wird über `import_sequences` gesteuert – damit können mehrere Belegarten (z.B. Bestellung, Lieferung) mit unterschiedlichen Statusübergängen definiert werden

Workflow

Bestellimport

1. Kunde gibt Bestellung im Shopware-Shop auf
2. Sync "Bestellungen" (`syncOrders`) starten – manuell oder als geplanter Job
3. Die Schnittstelle sucht Bestellungen anhand der in `status_fetch` konfigurierten Stati
4. Für jede gefundene Bestellung wird ein Beleg in der WinLine erstellt (Belegstufe konfigurierbar: Angebot, Auftrag, Lieferschein oder Rechnung)
5. Bei Erfolg wird der Bestellstatus in Shopware auf `status_success` gesetzt
6. Bei Fehler wird der Status auf `status_fail` gesetzt – die Bestellung wird beim nächsten Sync erneut verarbeitet

Sendungsverfolgung

- **Shop führend** (`trackingNrToWinline = True`): Sendungsnummer wird aus Shopware in die WinLine übertragen
- **WinLine führend** (`trackingNrToWinline = False`): Sendungsnummer wird aus der WinLine an Shopware übertragen

Sync-Befehle

- **syncOrders**: Bestellungen importieren und Sendungsnummern synchronisieren
- **syncOrderStatus**: Nur den Bestellstatus aktualisieren (ohne Belegimport)
- **syncShipmenttype**: Versandart synchronisieren

Tracking

Es gibt zwei Möglichkeiten die Sendungsnummer zu übermitteln. 1. Sendungsnummer wird in der WinLine vergeben und an die Shopbestellung übergeben. 2. Sendungsnummer wird von Shopware vergeben und an den WinLine-Beleg übergeben.

Beide Übergaben erfolgen im “syncOrders”.
Nachfolgend die Einstellungsoptionen in der Config
config:

```
[WL]
; Winline Feld für Sendungsverfolgungsnummer
; Hinweis: Darf nicht dasselbe Feld wie shoporderid sein (Standard: C063_Textzeil
trackingNrField = C063_Textzeile1
; Richtung der TrackingNr Übermittlung
; True - (Shop führend) Sendungsverfolgungsnummer wird aus dem Shop in die Winlin
; False - (Winline führend) Sendungsverfolgungsnummer wird aus der Winline in der
trackingNrToWinline = True
; Shopware Feld für Sendungsverfolgungsnummer
shoptrackingnrfield = trackingCodes
; Feld für die Shop-Belegnummer im WinLine-Beleg
shopbelegnummernfeld = C064_Textzeile2
```

Statuswechsel

Im Segment `ORDERS` im Feld `import_sequences` wird die Reihenfolge der Belegabrufe aus dem Webshop definiert. Der Wert im Beispiel `Bestellung` ist dann der Segmentname in der dieser Abruf konfiguriert wird.

Im Feld `import_belegstufe` (Segment hängt von der `import_sequences` ab) kann definiert werden in welcher WinLine Belegstufe, die vom Webshop abgerufenen Bestellungen, erfasst werden. Mögliche Belegstufen: `nicht_drucken`, `Angebot`, `Auftrag`, `Lieferschein` und `Rechnung`.

In den drei Feldern `status_*` kann eine Liste der Stati angegeben werden. - status_fetch: hier wird eine Verundung der Stati für die Suche in Shopware erstellt - bsp.: `status_fetch = OrderState.open, OrderTransactionState.paid` Es werden alle Bestellungen abgerufen, die Offen sind und die Bezahlt wurden. - status_success: Hier wird die Liste der Stati hinterlegt, in die eine Bestellung gesetzt werden soll, wenn sie erfolgreich abgerufen wurde. - status_fail: Hier wird die Liste der Stati hinterlegt, in die eine Bestellung gesetzt werden soll, wenn ein Fehler aufgetreten ist.

Statusarten in Shopware 6

Statusart	Config-Bezeichnung	Mögliche Werte
Bestellstatus	`OrderState`	`open`, `in_progress`, `cancelled`, `completed`
Bezahlstatus	`OrderTransactionState`	`open`, `failed`, `authorized`, `refunded_partially`, `refunded`, `in_progress`, `paid`, `paid_partially`, `reminded`, `cancelled`
Lieferstatus	`OrderDeliveryState`	`open`, `shipped`, `shipped_partially`, `cancelled`, `returned`, `returned_partially`

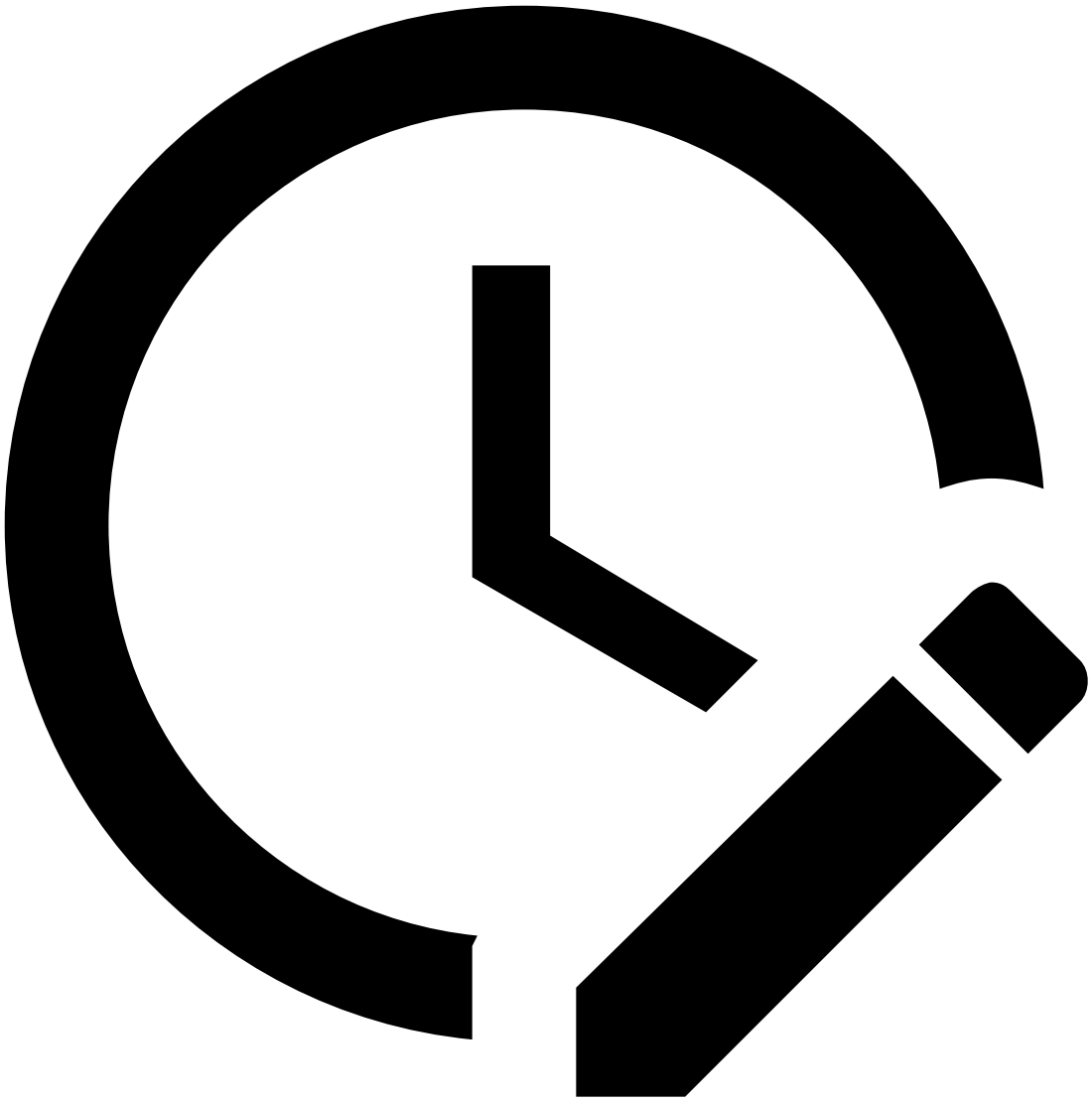
Beispielkonfiguration

```
[ORDERS]
import_sequences = ["Bestellung"]

[Bestellung]
; Optional: Statustyp für die Suche (OrderState, order_delivery, order_transactio
status_type = OrderState
status_fetch = OrderState.open
status_success = OrderState.inProgress
status_fail = OrderState.open
import_belegstufe = Lieferschein
check_existing = True
```

Weitere Belegarten

Es können weitere Segmente für verschiedene Bestellarten definiert werden:

```
[ORDERS]
import_sequences = ["Bestellung", "Lieferung"]

[Lieferung]
; Optional: Statustyp für die Suche (OrderState, order_delivery, order_transactio
status_type = order_delivery
status_fetch = OrderDeliveryState.open
status_success = OrderDeliveryState.shipped
status_fail = OrderDeliveryState.open
status_tracking = OrderDeliveryState.shipped
import_belegstufe = Lieferschein
```



2026-03-25

Umsetzungstabellen

Überblick

- Umsetzungstabellen werden verwendet, um Werte beim Datenaustausch zwischen Shopware und WinLine zu transformieren.
- Typische Anwendungsfälle: Anreden umsetzen (mr → Herr), Zahlungsarten zuordnen, Versandarten mappen.
- Die Tabellen werden sowohl beim Bestellimport (Order-Mapping) als auch bei der Feldzuordnung (Artikel-Mapping) eingesetzt.
- Umsetzungstabellen werden als JSON- oder YAML-Dateien im Unterordner `Umsetzungstabellen/` des Mapping-Verzeichnisses abgelegt.
- Es wird ein Wildcard-Zeichen (*) unterstützt für flexible Zuordnungen.

Ziel ist es mit Umsetzungstabellen die aus Shopware kommenden Werte beim Belegimport, auf WinLine Werte zu mappen.

> **Dateiformat**
>
> Umsetzungstabellen können sowohl im JSON-Format (`.json`) als auch im YAML-Format (`.yml`) angelegt werden. Beide Formate werden unterstützt.

Beispiel: Aus einem englischen Anredekürzel aus Shopware soll eine deutsche Anrede für den WinLine Beleg generiert werden.

Einrichtung - einfache Umsetzung

mapping:

```
"Anrede":{
    "winline": "Lief.Anrede",
    "shop": "salutation",
    "eval": "self.umsetzungstabelle('Umsetzung_Anrede', wl_data)"
    },
```

Order-Mapping\Umsetzungstabellen\Umsetzung_Anrede.json (oder .yml)

```
{
    "mr": "Herr",
    "mrs": "Frau"
}
```

zweifache Umsetzung

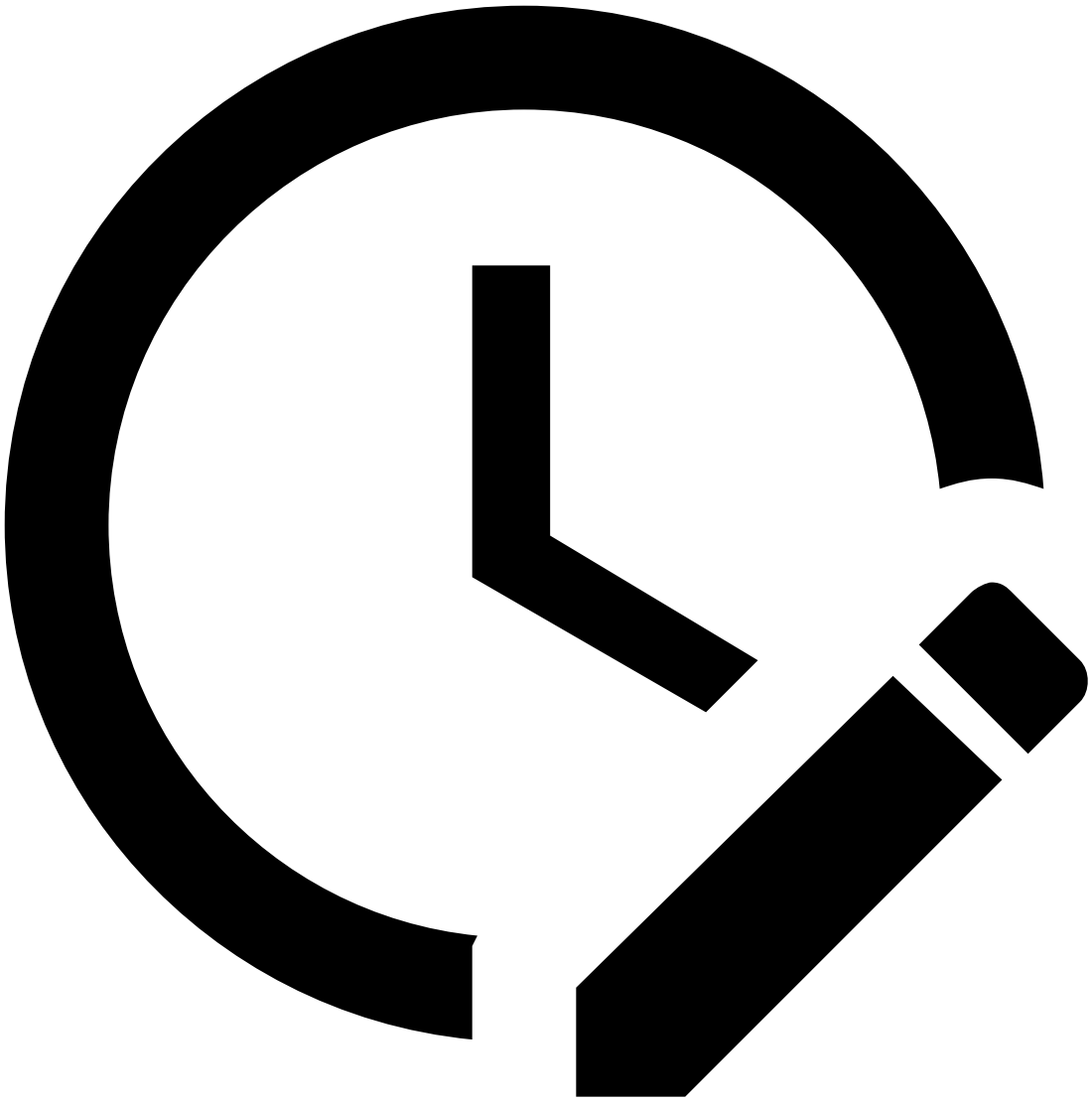
Im nachfolgenden Beispiel werden aus der Kombination aus payment.id & paymentStatus.id ein entsprechender Belegart-Wert in der WinLine generiert.
mapping:

```
"OP-Kennzeichen": {
        "winline": "Belegart",
        "shop": ["payment.id", "paymentStatus.id"],
        "concat_string_shop": "|",
        "eval": "self.umsetzungstabelle('Umsetzung_OP', wl_data)"
        },
```

Order-Mapping\Umsetzungstabellen\Umsetzung_OP.json (oder .yml)
“*” - dient als Füllwert für nicht explizit genannte Kombinationsoptionen

```
{
    "5|*": "B",
    "7|12": "P",
    "7|17": "O",
    "7|18": "O",
    "7|*": "O"
}
```

Siehe auch: Feldzuordnung für die Verwendung von Umsetzungstabellen im Mapping.



2026-03-25

Ausprägungen/Varianten

Überblick

- Ausprägungen in der WinLine entsprechen Varianten in Shopware
- Die Schnittstelle überträgt bis zu zwei Ausprägungsgruppen pro Artikel
- Varianten werden automatisch beim Artikelsync (syncProducts) mit übertragen
- Einzelne Ausprägungsartikel können eigene Bilder über das Archiv II erhalten
- Inaktive Varianten können über die Konfiguration ausgeblendet oder als “nicht verfügbar” angezeigt werden
- Lagerausprägungen können vom Sync ausgeschlossen werden

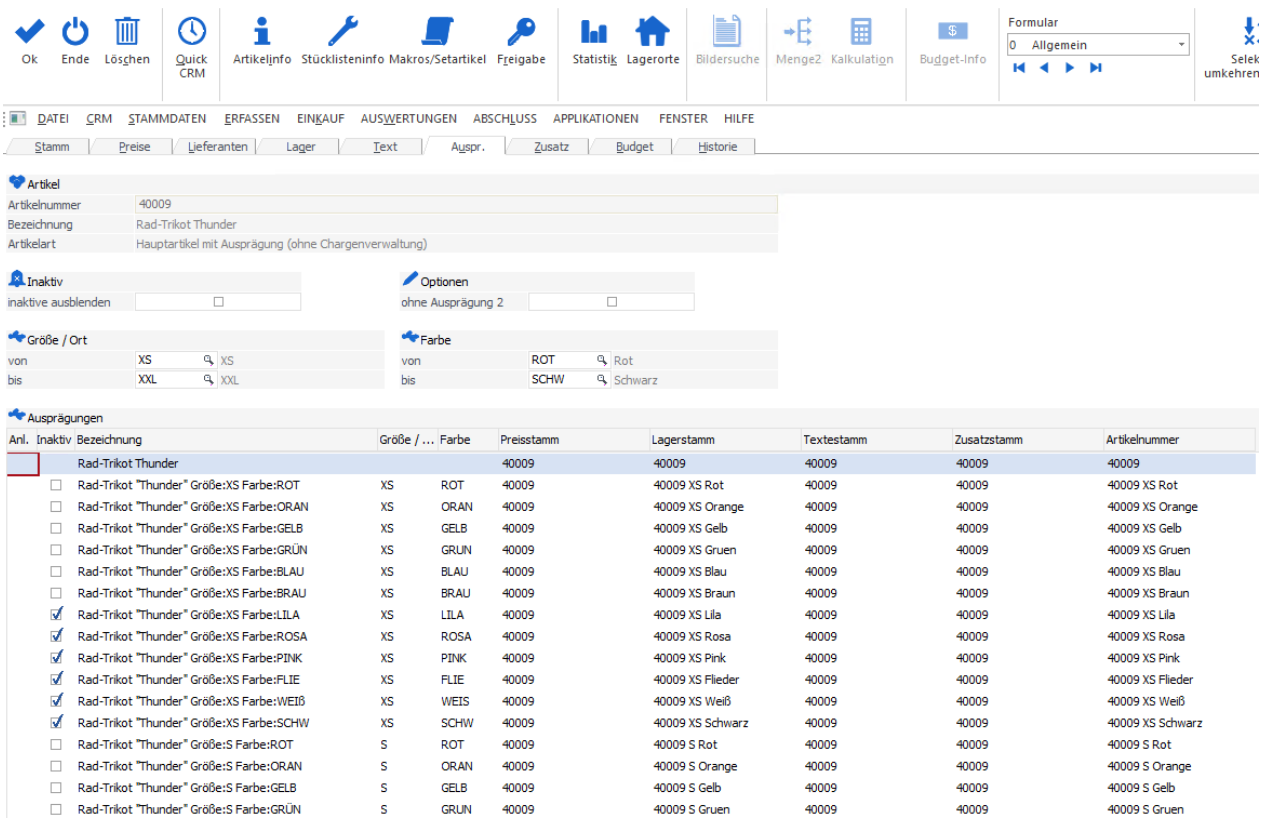
Workflow

1. Ausprägungen im Artikelstamm der WinLine anlegen (bis zu 2 Ausprägungsgruppen)
2. Optional: Bilder für einzelne Varianten über Archiv II zuordnen
3. Artikelsync starten (**syncProducts** oder **syncProductsAll**)
4. Die Schnittstelle erstellt den Hauptartikel und alle aktiven Varianten im Shop
5. Ergebnis im Shop prüfen: Varianten sind als Auswahloptionen auf der Produktseite sichtbar

Sync-Befehl

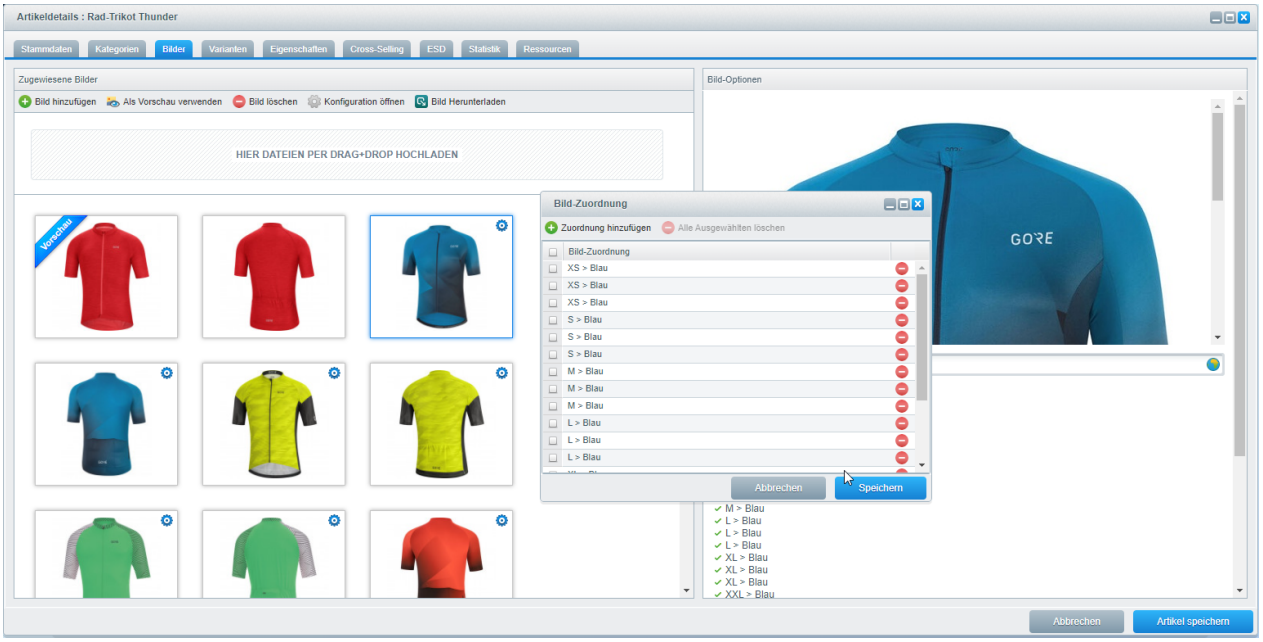
Varianten werden automatisch beim **Artikelsync** (syncProducts) übertragen. Ein separater Sync-Befehl ist nicht erforderlich.

Das was die WL als “Ausprägungen” bezeichnet, heißt in SW “Varianten”. Grundsätzlich gleichen sich die Systeme in diesem Punkt. In der WL lassen sich zwei Ausprägungen anlegen. Diese können nach dem angelegten Schema an SW übergeben und angelegt werden.

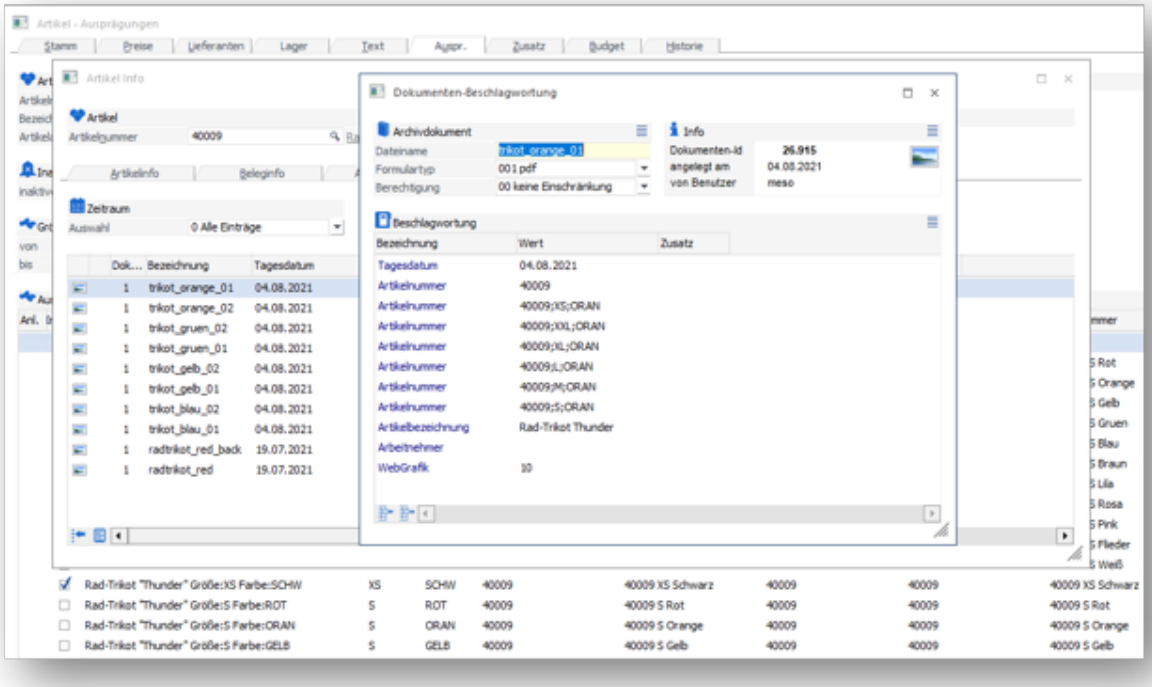


Ausrägungsartikel mit eigenen Bilder

SW verfügt über die Möglichkeit einzelnen Varianten zugehörige Bilder zuzuordnen.



Über die WL lassen sich diese Bildzuordnungen mit Hilfe des Archiv II realisieren. Dazu werden die Ausprägungsartikel per Beschlagwortung dem jeweiligen Bild zugeordnet. Diese Bilder werden im “syncProducts” mit übernommen und können zusätzlich über den “syncImages” übertragen werden.



inaktive Ausrägungsartikel

Über den config-Parameter “skip_inaktiv_variants = True” können die inaktiv gesetzen Varianten bei der Variantenanlage ausgeschlossen werden. Dadurch werden diese Varianten gar nicht angelgt.

Im False-Fall würden die Varianten in SW angelegt und auf inaktiv gesetz werden. Das führt dazu, dass diese Variante von SW als “Nicht verfügbar” im Frontend angezeigt wird.

config.ini

```
;Inaktive Ausprägungen werden nicht übermittelt wenn diese Option auf True steht.
skip_inaktiv_variants = True
```

Mapping (EANs Ausrägungsartikel)

Über das Mapping lassen sich einzelne Bereiche innerhalb der Variante ansprechen. Im nachfolgenden Beispiel wird der, in der Variante gepflegte, EAN-Code an den Shop übergeben. Über einen eigenen Zusatzstamm lassen sich natürlich weitere variantenspezifische Details mappen.

```
Variant:
  EAN:
    shop: ean
    winline: C075_EANCode
```

WL Lagerausprägungen

Durch die folgende Einstellung in der Config lassen sich angelegte Lagerausprägungen im Shop igonieren. Diese sollen in der Regel nicht als Variante im Shop verfügbar gemacht werden.

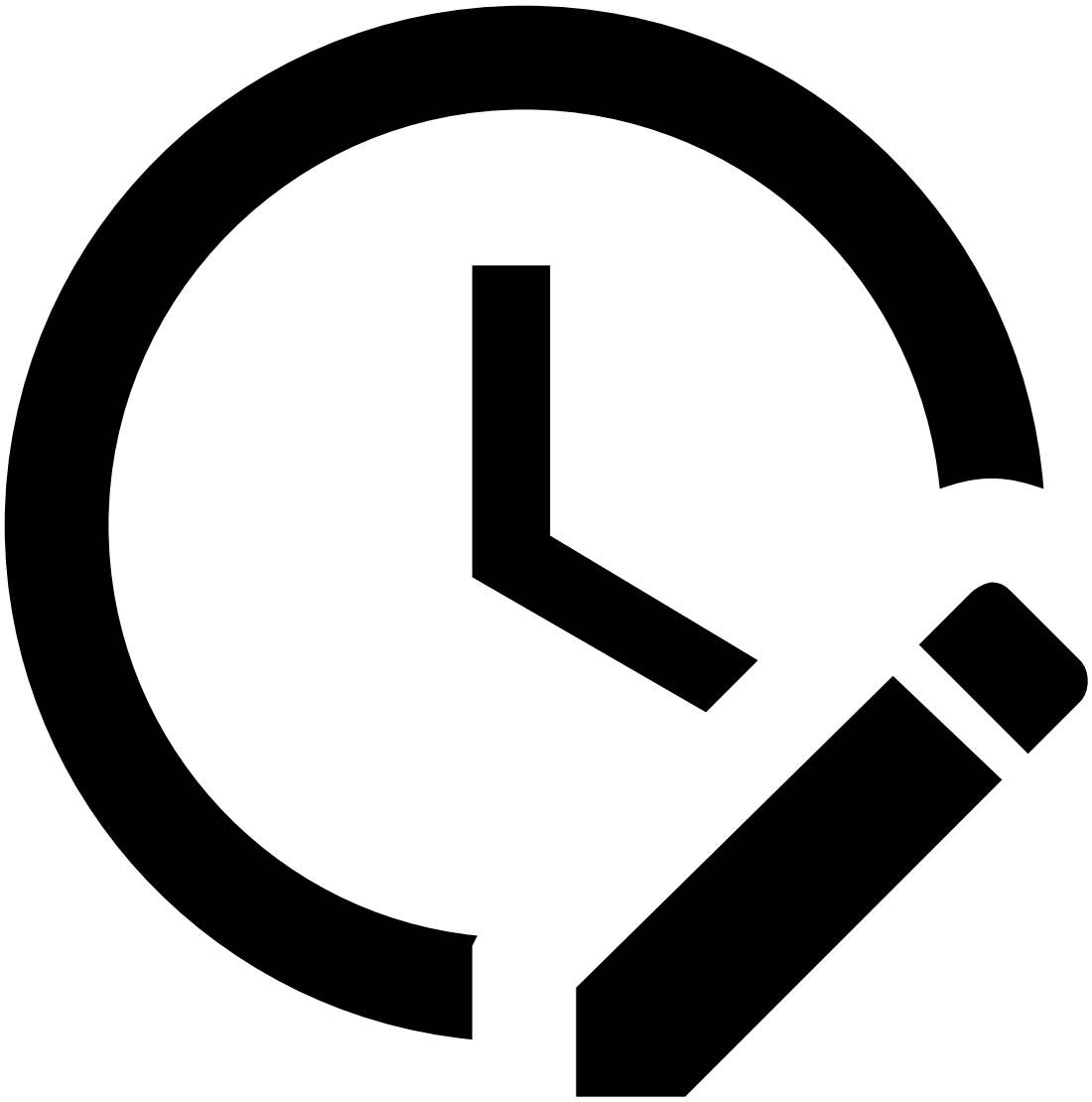
config.ini

```
;Alle hier aufgeführten (Kommagetrennt) Ausprägungsgruppennummern werden nicht al
ignoreauspraegungsgruppen = 3,6
```

Ausprägungen durch Eigenschaften

Die an Shopware zu übertragenen Eigenschaften der WinLine können direkt in der WinLine Konfiguriert werden. Shopware 6 kann diese für die Varianten verwenden. Hierzu wird ein Modul/Plugin benötigt.

Siehe: Eigenschaften WinLine Einrichtung



2026-03-25

Sonstiges

Jahreswechsel

Folgende Schritte sollten Sie während dem Jahreswechsel in der WinLine berücksichtigen

1. Pausieren / beenden automatischer Datenabgleiche in der Windows-Aufgabenplanung

Die Windows-Aufgabenplanung kann über das Startmenü und der Suche nach Aufgabenplanung gestartet werden. Die Aufgaben sind in der Regel auf dem gleichen System wie der WinLine-Server installiert. Die Aufgaben sollten für den Zeitraum des Jahreswechsels deaktiviert werden. Sofern der WinLine Action-Server für den Belegimport verwendet wird, den Action-Server-Dienst beenden.

2. Sofern der WinLine-WebService nicht verwendet wird

Import aller noch ausstehenden Bestellungen in die WinLine per EXIM oder Action-Server.

3. Jahresabschluss in der WinLine

Der Haken **“Kopie der Stammdaten erzeugen”** muss **angehakt** sein!

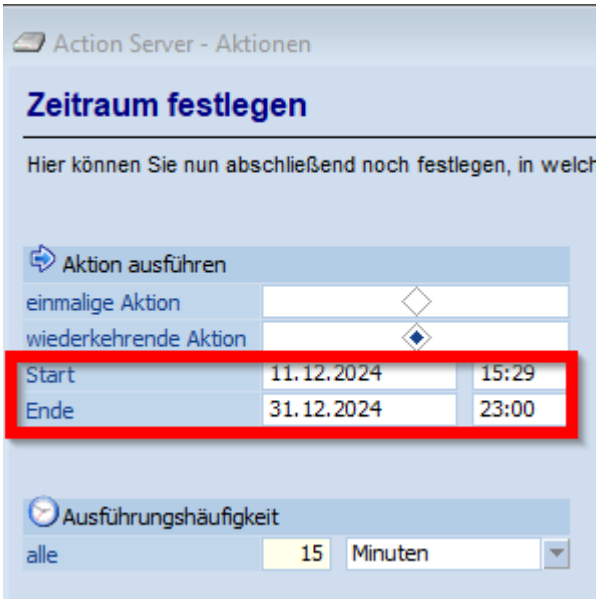
4. Nach dem Jahreswechsel

Im Fall von Import per Actionserver

In diesem Fall muss der Import bis zum 31.12. mit dem alten Wirtschaftsjahr erfolgen. Dies kann im Menüpunkt Action Server -> Report Definition der entsprechenden Aktion in WinLine hinterlegt werden.

Ab dem 01.01. muss dieser Wert manuell auf das neue Wirtschaftsjahr umgestellt werden.

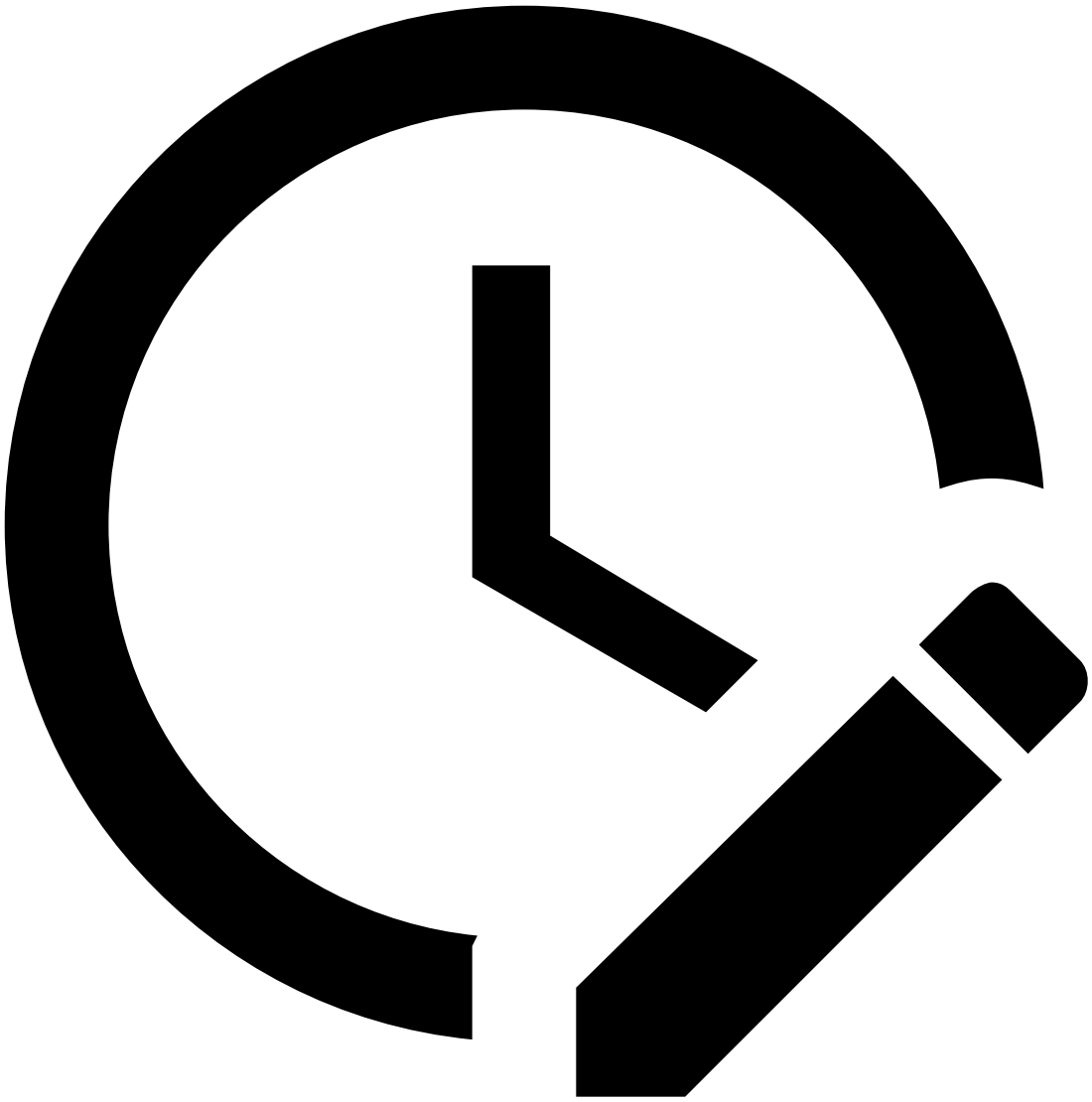
Alternativ kann eine zweite Aktion erstellt werden, welche als Startdatum den 01.01. erhält. Die Aktion für das alte Jahr muss dann mit einem Enddatum versehen werden (siehe Screenshot).



Im Fall von Import per WebService

Wenn das Feature `WEBSERVICE_COMPANYYEAR` aktiv gesetzt ist und es sich nicht um einen Mandanten mit abweichendem Wirtschaftsjahr handelt, werden Bestellungen automatisch in das korrekte Wirtschaftsjahr importiert.

In allen anderen Fällen kann der Import nach dem Jahreswechsel erst wieder ab dem 01.01. des neuen Jahres erfolgen.



2026-02-09

Logging

Überblick

Die Schnittstelle protokolliert alle Sync-Vorgänge in Logdateien. Die Logdateien können über das Management Interface eingesehen und heruntergeladen werden.

Debug-Stufen:

Flag	Level	Beschreibung
`-d`	ERROR	Nur Fehlermeldungen (Standard)
`-dd`	WARNING	Fehler und Warnungen
`-ddd`	INFO	Fehler, Warnungen und Informationen
`-dddd`	DEBUG	Alle Meldungen inkl. Debug-Ausgaben

Die Debug-Stufe kann pro Shop in der management.ini oder beim manuellen Start über die erweiterte Ansicht im Management Interface eingestellt werden.

Manuelle Konfiguration

Die Logausgaben können optional manuell konfiguriert werden. Eine Beispielkonfiguration findet sich in der Datei `logging.cfg`.

Wird diese Datei im Programmverzeichnis gefunden, wird die Logkonfiguration aus dieser gelesen. Der Dateiaufbau entspricht dabei dem Python-Standard *(https://docs.python.org/3/library/logging.config.html#module-logging.config)*.

Es stehen folgende Variablen zur Verfügung.

Variable	Beschreibung
%(logfolder)s	Das Verzeichnis für Logdateien für den Mandanten
%(filename)s	Der Dateiname für die Übertragungsart (syncProducts, syncPrices etc.)

Existiert in der Konfiguration ein Logger mit dem Namen `baseHandler`, so werden in diesen alle Lognachrichten für die laufende Übertragung geschrieben.

E-Mail-Benachrichtigungen

Über die `[MAIL]`-Sektion in der config.ini können Fehlerbenachrichtigungen per E-Mail versendet werden. Die Schnittstelle verwendet einen `BufferingSMTPHandler`, der Log-Nachrichten puffert und gesammelt per E-Mail versendet.

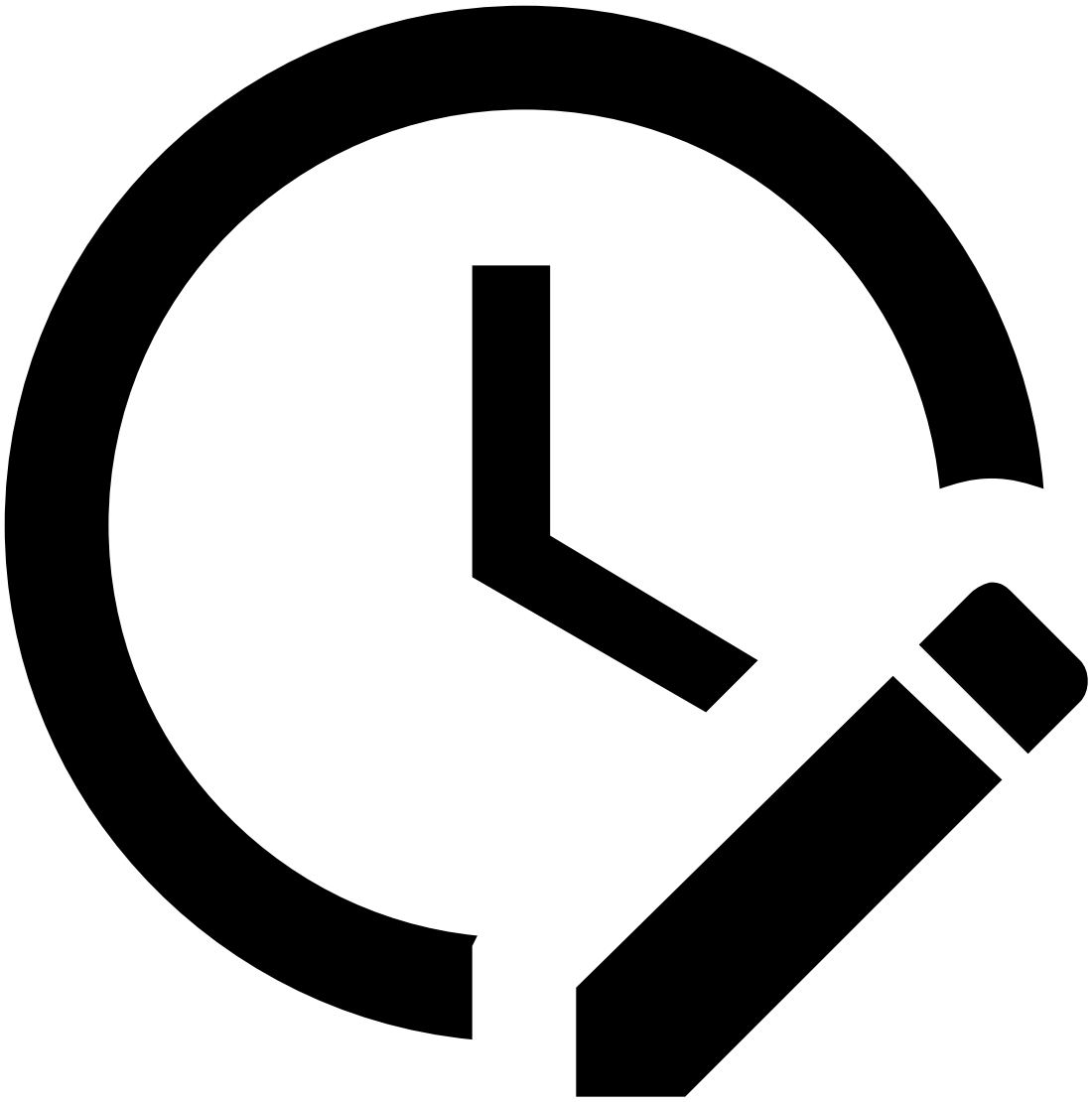
```
[MAIL]
; SMTP-Server
server =
; SMTP-Port
port = 0
; STARTTLS verwenden
use_tls = False
; SMTP-Benutzername
username =
; SMTP-Passwort
password =
; Betreff-Prefix
subject_prefix = [SHOPWARE FEHLERLOG]
; Absender-Adresse
sender =
; Empfänger-Adresse
receiver =
; Maximale Anzahl gepufferter Nachrichten
capacity = 1000
; Log-Level für E-Mail-Versand
level = DEBUG
; E-Mail-Versand aktivieren
tosend = False
```

Um den E-Mail-Versand zu aktivieren, muss `tosend = True` gesetzt und ein gültiger SMTP-Server konfiguriert werden. Bei aktiviertem `use_tls` wird STARTTLS für die Verbindung verwendet.

Plugin-Logger / Debug-Modus

Um das Logging für die Plugins zu aktivieren, muss in die logging.cfg folgender Block eingefügt werden bzw. der Log-Level muss auf DEBUG gesetzt werden:

```
[logger_shop.plugins]
level=DEBUG
handlers=consoleHandler,baseHandler
qualname=shop.plugins
propagate=0
```



2026-03-25

FAQ

Problembehebung

Die automatische Synchronisierung funktioniert nicht mehr

Webservice bzw. WinLine Actionserver neustarten

- Windows-Dienste öffnen
- `Mesonic WinLine Service Manager` neustarten

Aufgabenplanung prüfen

(Nur bei Verwendung der Aufgabenplanung)

- Windows Aufgabenplanung öffnen
- Datum letzte Ausführung prüfen
- Ggf. Rechtsklick auf den Dienst und Beenden wählen

Taskmanager prüfen

- Task-Manager öffnen
- ess.exe beenden

Logdateien prüfen

Wenn kein abweichender Pfad angegeben wurde befinden sich die Logdateien direkt im Ordner des Installationspfades oder im Ordner `scripts`. Der Inhalt der Datei kann auf Fehlermeldungen geprüft werden.

Bestellungen werden nicht importiert

Prüfen Sie folgende Punkte:

1. Ist der korrekte Bestellstatus in der Konfiguration unter `status_fetch` hinterlegt?
2. Ist die WinLine WebService-Vorlage korrekt eingerichtet?
3. Stimmen die Felder in der WebService-Vorlage mit dem Mapping überein?
4. Prüfen Sie die Logdatei auf Fehlermeldungen.

Sync-Fehler nachverfolgen

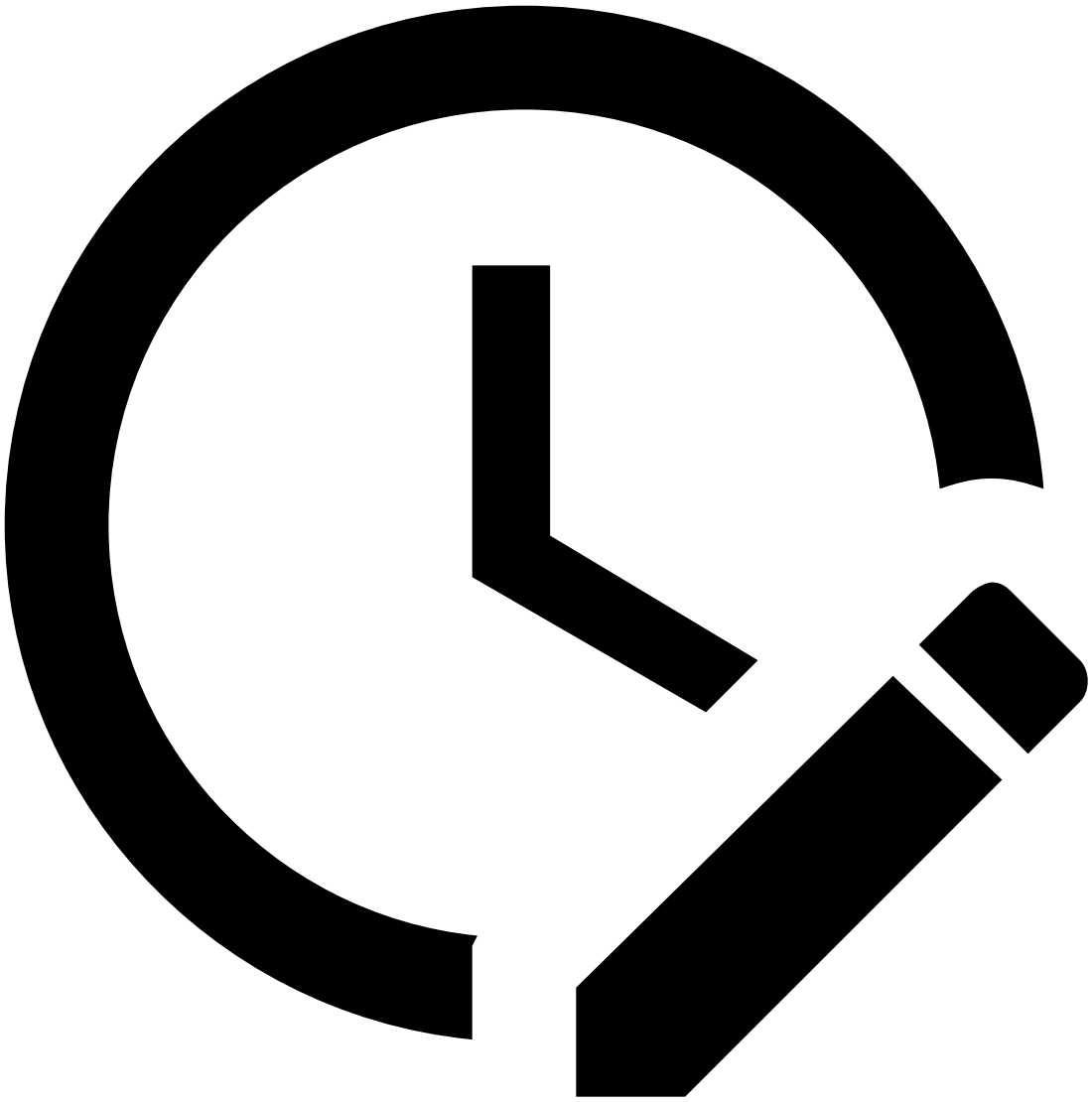
Über das Management Interface können unter `/{shop}/errors` alle aufgetretenen Sync-Fehler eingesehen werden. Die Fehler können nach Shop-URL, Entity-Typ und Auflösungsstatus gefiltert werden. Bei einem Full-Sync werden vorhandene Fehler automatisch als aufgelöst markiert.

Artikel werden nicht synchronisiert

1. Ist der Artikel in der WinLine als Webartikel markiert (Eigenschaftsfeld)?
2. Hat der Artikel einen gültigen Steuersatz der in Shopware angelegt ist?
3. Prüfen Sie ob das Änderungsdatum des Artikels neuer ist als der letzte Sync-Timestamp. Alternativ `-a` Parameter verwenden um alle Artikel zu übertragen.

Bild in Shopware ist defekt / fehlerhaft

1. Im Shopware-Backend den Artikel öffnen und unter dem Reiter `Allgemein` bis zu `Medien` scrollen.
2. Mit der Maus das defekte Bild hovern und den Dateinamen kopieren.
3. In den Bereich `Inhalte -> Medien wechseln` und nach dem kopierten Dateinamen suchen.
4. Das gefundene Bild im Backend löschen.
5. Eine neue Übertragung für den Artikel z.B. als Einzelsync starten.
6. Das Bild sollte im Shop korrekt angelegt werden.



2026-03-25

Release Notes

v3.2.0

Release Date: 2026-04-29

Neue Funktionen

- SH-385 Umfangreiche Ueberarbeitung des Managementinterfaces ("Webinar Nice-To-Haves"):
  - Kompakter Sticky-Header mit globalem Shop-Switcher und Sync-Status loest den wuchtigen Hero-Bereich auf dem Dashboard ab; die Sync-Kacheln liegen dadurch rund 70 % hoeher im sichtbaren Bereich.
  - Haeufig genutzte Combo-Syncs ("Artikel & Preise", "Kunden & Bestellungen") werden prominent in einer eigenen Featured-Zeile auf dem Dashboard angezeigt; Reihenfolge und Gruppierung sind ueber die neuen Sektionen `[DASHBOARD]` und `[DASHBOARDGROUP:*]` der `management.ini` konfigurierbar.
  - Die wuchtigen Page-Header-Banner (Bootstrap-Jumbotrons) auf allen Unterseiten wurden entfernt; die Topbar zeigt nun Seitentitel mit Icon, Beschreibung sowie Zurueck- und Aktions-Buttons.

  Weitere Verbesserungen: eindeutige, semantisch passende Icons je Sync-Kachel, einheitliches Inline-Banner-Styling fuer Status- und Fehlermeldungen, kompakte Live-Log-Card waehrend laufender Syncs.

- SH-388 Im Dashboard zeigt der Featured-Bereich jetzt nur noch die Kacheln "Artikel & Preise" und "Bestellungen". Die kombinierte Kachel "Kunden & Bestellungen" rutscht in den Bereich "Weitere Synchronisationen", da Bestellungen typischerweise deutlich haeufiger gesynct werden als Kunden.

- SH-390 Dashboard-Gruppen lassen sich pro Gruppe in der `management.ini` optional standardmaessig zugeklappt darstellen (`collapsed = true` in einem `[DASHBOARDGROUP:*]`-Block). Per Klick auf den Gruppen-Header laesst sich die Gruppe weiterhin auf- und zuklappen.

- SH-392 Im Aufgabenplaner gibt es eine neue Option "Verpasste Aufgaben nachholen" pro Aufgabe. Ist sie aktiv, wird eine Ausfuehrung, die z. B. waehrend einer Wochenend-Downtime ausgefallen ist, beim naechsten Start des Dienstes nachgeholt; mehrfach verpasste Ausfuehrungen werden zu einer einzelnen zusammengefasst. Das mitgelieferte Standard-Zeitplan-Preset nutzt die Option fuer den Wochenend-Vollsync "Artikel & Preise So 01:00".

- SH-395 Im AP-Dashboard ([/advanced/apdashboard]{.title-ref}) gibt es eine neue Karte „Alle geplanten Jobs (shop-uebergreifend)" mit Spalten Shop, Bezeichnung, Sync-Art, Trigger und naechstem Lauf. Verwaiste Aufgaben, deren Shop nicht mehr in [[SHOPS]] {.title-ref} der [management.ini]{.title-ref} steht, werden mit einem Badge markiert und lassen sich direkt aus dieser Liste entfernen.

Update-/Installationshinweise

- SH-395 Die Tabellen „Executions" und „Events" im AP-Dashboard zeigen jetzt die letzten 100 Eintraege. Die Stats-Counter (Hinzugefuegt/Entfernt/ Submitted/Erfolgreich/Verpasst) bleiben Gesamtzahlen ueber alle Eintraege.

Fehlerbehebungen

- SH-380: Timestamp wird nun auf das höchste Updated/Created Timestamp gesetzt wie bei der funktion syncFilteredCustomers

- SH-384:
    - healthCheck kann über management ausgeführt werden
    - Gastbestellungen über Sammelkonto
    - Versand und Zahlungsart im Default

- SH-385 Stabilisierung und Konsistenz des Managementinterfaces:

    - Shop-Switcher und Statusbar (letzter Sync-Status / laufender Sync-Indikator) funktionieren jetzt auf allen Seiten - vorher nur auf dem Dashboard.
    - Das AP-Dashboard laesst sich wieder oeffnen (`NoReverseMatch` durch veralteten `logpath`-Parameter behoben).

  Weitere Korrekturen: Configs oeffnen direkt den aktiven Shop statt hartkodiertem `local`; Django-Error-Messages werden konsistent dargestellt; AP-Dashboard-Executions als klickbares Detail-Modal statt ueberbreiter Tabelle (fehlgeschlagene Ausfuehrungen rot hinterlegt); stille Fehler beim Sync-Start werden sichtbar geloggt; Sync-Timestamps werden auch bei `consumer_key` aus `minimal.ini` korrekt angezeigt; Inhalt vergroesserter Featured-Kacheln wird vertikal zentriert.

- SH-387 Build-Fehler im PyInstaller-Skript behoben: Quellpfad fuer WinLineVorlagen in ess.spec korrigiert (Repo-Root statt bin/).

- SH-389 Im Dashboard verbraucht der Wildcard-Eintrag (`keys = *`) einer Gruppe nur noch die Tiles, die nicht von einer anderen Gruppe explizit beansprucht werden. Damit lassen sich Gruppen hinter "Weitere Synchronisationen" platzieren, ohne dass die Wildcard-Gruppe ihre Inhalte vorab schluckt.

- SH-391 Im Aufgabenplaner kann eine Aufgabe nachtraeglich auf "Minuetlich" gestellt werden. Die Intervall-Optionen im Bearbeiten-Dialog wurden mit der Anlegen-Maske

angeglichen und ein Tippfehler in den Choice-Keys ("Minunten") behoben, der die Edit-Logik bisher inkonsistent machte.

- SH-392 Tippfehler `coalsce` in den Scheduler-Defaults wurde zu `coalesce` korrigiert.
- SH-393 Im Aufgabenplaner werden gleichnamige Aufgaben in unterschiedlichen Shops nicht mehr gegenseitig blockiert. Die Existenz-Pruefung beim Standard-Zeitplan-Laden und beim manuellen Anlegen ist jetzt pro Shop. Vorher wurden Preset-Aufgaben fuer einen Shop stillschweigend uebersprungen, wenn sie unter einem anderen Shop schon einmal angelegt waren — die Tabelle blieb in diesem Fall leer. Beim manuellen Anlegen erschien fuer denselben Fall die irrefuehrende Meldung "Aufgabe existiert bereits".
- SH-393 `removejob`/`pausejob`/`resumejob` zeigen eine Meldung an, wenn die uebergebene Task-ID nicht (mehr) existiert, statt mit `AttributeError` abzustuerzen.
- time_to_safe checkt jetzt auch ob es None ist bevor es genutzt wird.

Sonstiges

- SH-375:
    - Neue Vorlagen für Beleg und Kunden/Kontakte Import
    - PyInstaller/Wix angepasst für neu Vorlagen

v3.1.0

Release Date: 2026-04-28

Neue Funktionen

- SH-260 Neues Feature-Flag REMOVE_ORPHAN_MEDIA entfernt beim Bilder-/Download-Sync automatisch alle Shopware-6-Artikelmedien, deren WinLine-Schlagwort (Webshop Grafik bzw. Webdokument) zwischenzeitlich entfernt wurde. Mit Sicherheitsnetz gegen versehentliches Loeschen bei leerem WinLine-Abruf.
- SH-371 Im Aufgabenplaner steht ein neuer Button "Standard-Zeitplan laden" bereit. Ein Klick legt drei typische Hintergrundaufgaben für den aktuellen Shop an: kombinierter Artikel- und Preissync (Mo-Fr 01:00), Vollsync für Artikel und Preise (So 01:00) sowie Bestellungen (Mo-Fr 06-18 Uhr alle 15 Minuten). Bereits vorhandene Aufgaben mit gleichem Namen bleiben unangetastet.
- SH-371 Jede geplante Aufgabe im Aufgabenplaner kann nun optionale CLI-Parameter erhalten (z. B. "-a" für Vollsync, "-wc" ohne Kategorien). Damit lässt sich derselbe Sync an unterschiedlichen Wochentagen mit verschiedenen Flags ausführen. Der Sonntags-Eintrag des Standard-Zeitplans läuft jetzt mit "-a", sodass über das Wochenende garantiert alle Artikel und Preise vollständig synchronisiert werden.
- SH-372 Management Interface: Neuer Navigationseintrag "Onlinedokumentation" verlinkt direkt auf die offizielle Doku.

- SH-378 Im Installationsverzeichnis liegen jetzt zwei Helfer-Skripte `init.bat` (Erstkonfiguration via `ess.exe init`) und `install_service.bat` (Windows-Service registrieren und starten). Beide fordern automatisch Adminrechte an (UAC) und ersetzen den bisherigen manuellen Aufruf in einer Admin-Konsole.
- SH-378 Das Management Interface läuft jetzt als Windows-Service mit Waitress als Produktions-WSGI-Server (Port 8001) hinter einem WinSW-Wrapper. Autostart bei Systemstart, automatischer Neustart nach Fehler und optionale Reverse-Proxy-Authentifizierung (REMOTE_USER via X-Forwarded-For) werden unterstützt. Der Dienst wird aus dem Installationsverzeichnis mit `management_service.exe install` registriert.

Fehlerbehebungen

- SH-370: Mandant und Saleschanel in Default/Minimal aufgenommen
- SH-372 Management Interface: Link "Zum Shop" funktioniert nun zuverlässig, öffnet in einem neuen Tab und wird nur angezeigt, wenn in der Shop-Config eine URL hinterlegt ist. Fehlende URL-Schemas werden automatisch ergänzt. Tippfehler in der Support-Mailadresse korrigiert.
- SH-373: Wenn Bestellungen nicht in die DB eingefügt werden kann, wird diese in die Fehler-Liste geschrieben.
- SH-374 Die Synchronisations-Kacheln im Managementinterface werden während eines laufenden Syncs nicht mehr ausgegraut und bleiben klickbar. Die Live-Log-Anzeige bleibt weiterhin erhalten, und ein zweiter Sync-Versuch wird wie bisher mit einer Fehlermeldung verhindert.
- SH-377 Auf der Sync-Fehler-Seite werden Entity-Typen im Filter-Dropdown nicht mehr mehrfach angezeigt und erscheinen nun alphabetisch sortiert.
- SH-379 Der Kategorie-Sync bricht nicht mehr mit Foreign-Key-Fehlern auf `parent_id` ab. Kategorien werden vor dem Upsert topologisch sortiert, so dass Eltern-Kategorien immer vor ihren Kindern angelegt werden. Dies gilt sowohl für die initiale Kategorie-Anlage als auch für den Artikel-Upsert, bei dem bisher nur Blatt-Kategorien mitgeschickt wurden und deren Eltern bei ad-hoc angelegten Kategorien fehlen konnten.
- SH-371 Die Aufgabenplaner-Seite stürzt nach dem Laden des Standard-Zeitplan-Presets nicht mehr mit einem Fehler ab. Cron-Aufgaben ohne gesetztes Startdatum werden in der Übersicht mit „-" als Startzeit angezeigt.

v3.0.3

Release Date: 2026-04-13

Fehlerbehebungen

- SH-364:
    1. Default Configs (config, minimal, mnanagement) überarbeitet.
    2. Artikel + Preis sync

- SH-367 Fehler beim Filtern von WinLine-Belegen behoben, wenn ein Belegfeld keinen Wert enthält.
- SH-368:
  - Fehler behoben, wenn das ID Feld im Beleg auf None gesetzt wurde
  - Fehlerhafte Kunden werden übersprungen und in den Fehler-Log des Management Interfaces geschrieben.
- SH-369: Alter Timestampt aufruf auf timestamp_manager umgestellt

v3.0.2

Release Date: 2026-04-08

Sicherheits-Updates

- SH-361: Update aiohttp to 3.13.5 to resolve 10 new found vulnerabilities in 3.13.3

Fehlerbehebungen

- SH-358 Behebt einen Fehler im Job-Dashboard (apdashboard), durch den die Seite mit einem TypeError abstürzte, wenn Events ohne zugehörigen Job in der Datenbank vorhanden waren. Verwaiste Events werden jetzt übersprungen und im Log vermerkt.
- SH-359: Keine Position in die DB importieren wenn die Versandkosten Leer sind.
- SH-360: Init muss for dem benutzen des consumer_Keys ausgeführt werden.

Sonstiges

- SH-351 Veraltete Screenshot-Hinweisboxen aus der Dokumentation entfernt.

v3.0.1

Release Date: 2026-03-27

Fehlerbehebungen

- SH-353 Management-Interface kann nun über den Shop-Launcher gestartet werden, ohne dass der Django Auto-Reloader einen Fehler verursacht.
- SH-354 Sync-Start aus PyCharm schlug mit ModuleNotFoundError fehl, da die Django-Settings-Konfiguration nicht korrekt an den Sync-Subprocess weitergegeben wurde.

Sonstiges

- Der Standardport für das Management Interface ist jetzt 8001

v3.0.0

Release Date: 2026-03-26

Neue Funktionen

- SH-345 SQLite-Datenbankperformance durch PRAGMA-Optimierungen verbessert (WAL-Modus, Memory-Mapped I/O, Busy-Timeout).
- SH-347 Der Init-Befehl erstellt nun automatisch eine management.ini mit Standardwerten fuer den ManagementService.

Update-/Installationshinweise

- SH-332 Shop-URLs werden nun automatisch aus der jeweiligen config.ini ausgelesen. Der Launcher-Pfad wird automatisch bestimmt.
- SH-332 Neue Fehlerübersicht: Sync-Fehler werden nun in der Datenbank protokolliert und können über eine eigene Seite eingesehen, gefiltert und als gelöst markiert werden.

Fehlerbehebungen

- SH-332 Management Interface: Fehlgeschlagene Synchronisierungen werden nun korrekt als fehlerhaft erkannt, auch wenn kein Traceback in der Ausgabe enthalten ist.
- SH-332 Management Interface: SyncProgress-Datensätze werden nun pro Shop wiederverwendet statt bei jedem Sync-Vorgang neue Einträge zu erzeugen. Die Datenbank wächst dadurch nicht mehr endlos an.
- SH-332 PermissionError beim Starten der Exe behoben, wenn der konfigurierte tmp-Pfad relativ angegeben ist.
- SH-337 Fehler behoben, bei dem ein unbekannter Shop-Name im Management Interface zu einem Serverfehler führte.
- SH-338 Management Interface: Ungültige URL-Pfade wie /.well-known oder robots.txt führen nicht mehr zu einem Serverfehler, sondern werden korrekt mit einer 404-Seite beantwortet.
- SH-339 In Testumgebungen wird nun der korrekte Shop-Launcher (energy_connect_shop_launcher.py) über Python gestartet, statt die nicht vorhandene ess.exe zu suchen.
- SH-340 Sync-Prozess startet nun mit korrektem Arbeitsverzeichnis, sodass relative Pfade (z.B. Mapping-Ordner) korrekt aufgelöst werden.
- SH-340 SW5-only Sync-Funktionen (z.B. syncAttributes, syncEigenschaften, syncCross, syncTranslations) werden bei SW6-Shops nun automatisch übersprungen statt einen Fehler zu verursachen.
- SH-342 Management Interface: Einzel-Sync Tab wird nun korrekt angezeigt und ist wieder nutzbar.
- SH-343: Fehler wenn orderId nicht im Shop gefunden werden kann.

- SH-346 Sync-Timestamps werden jetzt in der SQLite-Datenbank gespeichert statt in .timestamp-Dateien. Dadurch werden Schreibfehler bei schreibgeschützten Config-Verzeichnissen (z.B. Program Files) behoben. Bestehende Timestamp-Dateien werden beim ersten Start automatisch migriert.
- SH-346 Die Logs-Seite zeigt jetzt alle Logdateien des aktiven Shops an. Zuvor wurden nur wenige Dateien angezeigt, da der Log-Pfad nicht korrekt anhand des Shop-Mandanten aufgelöst wurde.
- SH-352: CustomFieldSet CustomField Config wird nicht mehr ersetzt und kann dadurch zum teil im Shop gepflegt werden. Vor allem können Select Felder angelegt und deren Optionen dann im Shop gepflegt werden.
- SH-316 Dokumentation an aktuellen Code-Stand (v2.12.4) angepasst. Falsche Default-Werte, fehlende Konfigurationsoptionen und veraltete Statusbezeichnungen in der mkdocs-Dokumentation korrigiert.

Sonstiges

- Die Endnutzer-Dokumentation wurde vollständig für Shopware 6 aktualisiert. Veraltete Shopware 5 Inhalte und Warnhinweise wurden entfernt, fehlende Seiten zur Navigation hinzugefügt und die Konfigurationsdokumentation überarbeitet.

Major Release

- SH-332 Breaking Change: Das Management Interface wurde von Flask auf Django 6 migriert. Die management.ini muss angepasst werden – die Sektionen [HOST], [MANAGEMENT], [SOCKET], [PROFILER], [SHOPURL] und [SHOPPATH] müssen entfernt werden. Die [SHOPS]-Sektion enthält nun direkt den Pfad zur config.ini jedes Shops. Der Startbefehl ändert sich zu `ess.exe manage runserver` (Port 8000 statt 5000). Siehe Migrationsanleitung in der Dokumentation unter "Update".

v2.12.4

Release Date: 2026-03-13

Fehlerbehebungen

- SH-341: Configurations file und timestamp wurden immer neu geschrieben

v2.12.3

Release Date: 2026-03-11

Fehlerbehebungen

- SH-330: Es werden nur noch beim Kunden Import per Kontakt alle Kunden im Shop abgerufen und hier wird auch ein differenzieller Ansatz verwendet.

v2.12.2

Release Date: 2026-03-09

Fehlerbehebungen

- Fallbackwerte im Validator müssen Strings sein
- SH-313:
  - Werden mehr als 1 Kontakt gefunden kann wird nun richtig gefiltert auch wennd as Updated Datum fehlt
  - Wenn das Updated Datum fehlt kannd as Datumsfeld im Shop nicht aktualisiert werden, als Fallback dient nun das Created at feld der WinLine.
- SH-323: Wenn sich die Varianten (bzw. die Konfiguration derer) in der WinLine ändert, werden nun die nicht mehr benutzten gelöscht.

Sonstiges

- SH-262:
  - Dokumentation zu Artikelgrafiken überarbeitet.
  - Dokumentation zu Ausprägungen/Varianten überarbeitet.
- SH-315: Plugin um neue Text-Positionen auf einen Beleg zu importieren wenn eine bestimmte Zahlungsart gewählt wurde.
- SH-327 FAQ um Anleitung zur Behebung defekter Bilder in Shopware ergänzt.

v2.12.1

Release Date: 2026-01-14

Fehlerbehebungen

- SH-310: Bestellungen die per Datenbank abgerufen werden, werden nun korrekt im Shop in einen neuen Status gesetzt.

v2.12.0

Release Date: 2026-01-14

Neue Funktionen

- SH-303 Automatischer Import in das dem aktuellen Jahr zugehörige Wirtschaftsjahr solange das Feature WEBSERVICE_COMPANYYEAR nicht manuell deaktiviert wurde.
- SH-270: Shopaccounts werden als Kontakte in die WinLine importiert. Hierzu wird die Email-Adresse herangezogen, um bestehende Kontakte zu finden. Zudem wird die Kundennummer ersetzt, das Datum des Imports und die Nummer des Kontaks (Abweichend zur Kontaktnummer) in den Shop zurückgeschrieben (Abweichend von der). So kann beim Import der Bestellungen der Kontakt mit angegeben werden.
- SH-309: Personenkonten und Kontakte können über die Datenbank importiert werden.

Fehlerbehebungen

- Beim Update der Kunden im SHop wurden immer alle Aktualisiert und es handelt sich nicht immer bei dem shop-konto um ein Ressource object.

Sonstiges

- Error handling

v2.11.6

Release Date: 2026-01-08

Fehlerbehebungen

- Turned logger in shopware6.py _map_position from error to warning. position will get skipped when not in mapping.

Sonstiges

- Plugin to get KundenNr from Email. Little addition in Shop to make new Plugin work. fixed typo

v2.11.5

Release Date: 2025-11-13

Fehlerbehebungen

- SH-299: Mapping muss nach den meisten generate*_data_for_article() funtionen laufen

v2.11.4

Release Date: 2025-11-12

Fehlerbehebungen

- SH-295: Plugability für generate_article_data verbessert und refactoring
- SH-297 - Archiveinträge ohne das Schlagwort "Artikelbezeichnung" wurden komplett rausgefiltert. Da es aber kein Pflichtfeld ist, ist der Join jetzt ein Outerjoin und der Filter ist in der Join-On-Clause.

v2.11.3

Release Date: 2025-11-04

Fehlerbehebungen

- SH-298: Wenn ein PDF als Artikelbild verwendet wird crashed die Schnittstelle. Shopware unterstützt so auch keine PDFs aber das Medium wird trotzdem übertragen.

v2.11.2

Release Date: 2025-10-15

Fehlerbehebungen

- SH-290 Rabattzeilen mit nur einem zugeordneten Kunden wurden als kundenindividueller Preis angelegt. Das sorgte außerdem dafür, dass Preise teilweise unvollständig übertragen wurden.
- SH-291 Die Configdatei wird nicht in UTF-8 geschrieben, was beim Laden zu problemen führt.
- SH-289: Wenn die Preisliste nicht gesetzt ist kommt es bei einem Plugin zu Fehlern.
- SH-293: Der Kundensync hat in bestimmten Fällen die Zeitzone verloren und dadurch Kunden nicht abrufen können.

v2.11.1

Release Date: 2025-09-26

Fehlerbehebungen

- Wenn keine Kategorien übertragen werden sollen, wurden dennoch eine Leere Liste gesetzt (Manuell im Shop gepflegte Kategorien werden damit entfernt)

v2.11.0

Release Date: 2025-09-25

Neue Funktionen

- SH-266 tqdm für search-Endpunkte.
- SH-278 Rabattmatrix zusammenfassen: Wird die Rabattspalte für alle Preiseinträge des Artikels auf [fixe Rabattspalte vom Artikel]{.title-ref} gesetzt, kann die Schnittstelle Preise der Rabattmatrix zusammenfassen, um Preiseinträge im Shop zu reduzieren und in diesem die Performance zu verbessern. Weicht die Rabattspalte einer Preiszeile in der WinLine ab, werden die Preise im Shop pro Kunden erstellt. Dieses Feature wird über die Featureflag [SUMMARIZE_ACCOUNT_MATRIX = True]{.title-ref} aktiviert.
- Es kann nun ein Pfad in der Konfiguration angegeben werden welcher bei der Bildübertragung aus Zusatzfeldern genutzt wird. zusatzgrafikpfad = C:PfadzuOrdnerBilder

Fehlerbehebungen

- SH-267 Korrektur der Preisübertragung für einzelne Artikel (-n). Diese löschte all Preise.
- SH-269 Die Funktion syncPrices konnte Preise nicht bereinigen, wenn noch kein Preis in den Shop übertragen wurde.
- SH-274 Die Übertragung der Rabattmatrix konnte den Rabatt des Standardpreises überschreiben.
- SH-276 Korrektur der Bismengenberechnung bei Verwendung der Rabattmatrix.
- SH-280 plugin-Logger wird nichtmehr automatisch auf DEBUG gesetzt.
- SH-280 Encoding wird in DEBUG-Ausgaben berücksichtigt.
- SH-282 Bereitstellung der ursprünglichen Order-Daten in optionalShopData für Order_Beleg_Rechnung und Order_Beleg_Lieferung.
- SH-288 Filtern von überschneidenden Preisen berücksichtigt jetzt die Rabattflag.
- SH-286: Wenn weder Artikeluntergruppen noch Kategorien aus Eigenschaften Konfiguriert sind, werden keine Kategorien an den Shop übermittelt. Die Artikeluntergruppen werden über die Konfig-Wert im Segment 'WL': 'Kategorien_aus_Artikeluntergruppen = True' aktiviert

Sonstiges

- SH-267 ODBC Driver 18 for SQL Server als Standardtreiber gesetzt. SQL Server wird nicht mehr unterstützt.

- SH-275 Erweitertes (DEBUG-) Logging für die Übermittlung der Daten als Chunk.
- SH-278 gen_rule_uuid ignoriert ab sofort den Parameter "value", da dieser sonst dafür sorgt, dass Rabattmatrixregeln nicht aktualisiert werden können. Für alle Fälle, ausser der Rabattmatrix, enthält rule in Verbindung mit dem Namen immer den selben Wert.
- SH-273: Tracing für Plugins und Logging
- Schöneres Logging für Bildübertragung. Mini SQL Fix bei Anfragen mit Sonderzeichen.
- SH-284: Fallback für nicht konfigurierte Eigenschaften ist nun nicht Filterbar und nicht auf Produktseite anzeigen.
- Brutto/Netto-Flag kann jetzt aus Preisliste eines Preislisteneintrags ausgelesen werden

v2.10.1

Release Date: 2025-08-29

Fehlerbehebungen

- SH-264 Löschen von Preisen in Chunks, da Shopware bei Löschen per Criteria bei großen Datenmengen Probleme macht.

v2.10.0

Release Date: 2025-08-29

Neue Funktionen

- SH-264 Neuer Parameter onlyDeleteUnused für syncPrices um alle ungültigen preise zu entfernen ohne Neue anzulegen.
- SH-265 Unterstützung für WSL2.
- SH-265 Unterstützung für Self-Signed-Certificates und MSODBC18.

Fehlerbehebungen

- SH-264 Behebt einen Speicherleck am Ende der Preisübertragung.
- SH-265 Korrektur falsch kodierter Dateien.
- SH-265 Optimierung der Preisbereinigung im Shop.
- SH-257: Anstatt die Preise aus dem shop abzurufen, wird eine delete condition erstellt die alle Preise löscht, die nicht im Update enthalten waren.

Sonstiges

- Local SSL with truststore
- Feld in der Config hinzugefügt dass es ermöglicht die is_exe() Methode zu umgehen und somit SSL local möglich zu machen.

- viztracer und hunter in Pipfile hinzugefügt, können jetzt in Plugins genutzt werden.

v2.9.6

Release Date: 2025-08-13

Prelude

Fehlerbehebungen

- SH-256: Es werden nun keine Rabattmatrix-Preise für Kunden angelegt die keine Rabattleiste 1 oder Rabattleiste 2 hinterlegt haben oder bei denen der Rabatt 1 und 2 0% hat.

v2.9.5

Release Date: 2025-08-08

Fehlerbehebungen

- SH-254: Fehlende Rabattmatrix-Einträge für Preisart Verkauf werden nun generiert.

v2.9.4

Release Date: 2025-08-06

Fehlerbehebungen

- SH-233: Es werden nun korrekt die Plugins aus dem Plugin Ordner geladen, wenn die die exe ausgeführt wird. Vorher mussten die Plugins sowohl im Plugins ordner als auch im _internal/plugins ordner liegen.

Sonstiges

- get_article_groups_filter_by_eigenschaften kann jetzt auch ab einer bestimmten Untergruppennummer filtern.
- get_article_groups_filter_by_eigenschaften kann jetzt auch nur Kategorien ab einer bestimmten Artikeluntergruppe übertragen.

v2.9.3

Release Date: 2025-08-04

Fehlerbehebungen

- SH-253 _data nicht initialisiert

v2.9.2

Release Date: 2025-08-04

Fehlerbehebungen

- SH-252: search aufrufe funktionieren ohne BulkData daher musste hier gesondert geprüft werden ob ein _Filter übergeben wurde um diesen dann mir .toDict() zu konvertieren.

v2.9.1

Release Date: 2025-07-31

Fehlerbehebungen

- SH-246 Umstellen der send.json für den send-Befehl auf UTF-8-Encoding.
- SH-247 Anpassung der Beispieldatei für Logging ohne Dateigrößenbegrenzung.
- SH-249: Da _Filter (und die erbenden Klassen) von dict erben, wurde die interne representation nicht aktualisiert, wenn eine der Properties der Instanz aktualisiert wurden. Das hat mit der internen implementation von dict zu tun. Da das Problem eigentlich nur bei der Konvertierung eines Bulkdata-Object zu einem dict wichtig war, musste nur die toDict() Methode von Bulkdata richtig implementiert werden und die vererbung zu dict war nicht mehr nötig.
- Funktion "get_article_groups" wird nun mit SQLAlchemy gemacht. Funktion "get_article_groups" ist nun plugable.

v2.9.0

Release Date: 2025-07-24

Neue Funktionen

- SH-247 Verbessertes Logging-System mit konfigurierbaren Log-Datei-Pfaden. Das System unterstützt jetzt dynamische Pfade für Log-Dateien über Konfigurationsvariablen.
- SH-247 Neue detaillierte Request-Protokollierung für Shop-API-Aufrufe. HTTP-Requests werden nun separat geloggt und können über den Logger 'shop.requests' gesteuert werden.

- SH-247 Der Standardlogger kann in der logging.cfg optional unter dem Namen baseHandler überschrieben werden.
- SH-246: Developer-Parameter um direkt requests an den SHop zu schicken. Dies kann einfach ein get an einen endpunkt sein, ein post an einen endpunkt mit dem json als Übergabeparameter oder ein post mit json aus einer Datei.

v2.8.1

Release Date: 2025-07-23

Fehlerbehebungen

- SH-241 Verbesserung der Dateipfad-Verarbeitung bei RTF-Konvertierung.
- SH-245 Plugable-Registrierung für den SyncManager ergänzt.
- Shopware nimmt keine spitzen Klammern für Kategoriennamen an. Hier am besten auf Sonderzeichen verzichten. Es wird jetzt explizit auf "\<" und ">" geprüft und diese entfernt.
- Bessere Checks und Errorhandling wenn order ein String ist und Cast to String entfernt.

Sonstiges

- Die Versionsnummer wird nun auch in der "__init__.py" hochgezählt wodurch die ".log" Dateien richtig abgelegt werden.

v2.8.0

Release Date: 2025-06-30

Neue Funktionen

- SH-224 Featureflag remove_price_duplicates, welche Preisduplikate mit überschneidenden Gültigkeitsdaten entfernt.
- SH-236 Installer per MSI für vereinfachte Installation und Upgrades.
- Es ist nun möglich im Mapping [DYNACROSS] für das dynamischen Crossselling die Felder sort_by und sort_direction zu setzen wodurch die Sortierung angegeben werden kann.

Fehlerbehebungen

- Reihenfolge der unterschiedlichen Bulkdata entities war nicht deterministisch. Dadurch wurden manchmal Artikel geupdatet obwohl noch Datem gefehlt haben. Dazu wurde im "example_article.yml" noch ein fehler behoben, da hier der Lieferzeit_Unit Defaultwert "Tag" war aber "day" sein muss.
- cross_gruppe_shop wird jetzt nur noch erstellt wenn alle Werte hierfür gesetzt sind.

Sonstiges

- Bessere Logs fürs Debugging mit der Möglichkeit sie per Argument zu aktivieren.
- hauptartikel_als_varianten in plugins/examples + erklärung in installation.md

v2.7.0

Release Date: 2025-06-18

Neue Funktionen

- SH-232 Neue Vorlage zum Ausblenden von Eigenschaften.

Fehlerbehebungen

- SH-230 Unicode-Excaping von Stück im Standardmapping.
- Bulkdata mit Filterkriterien können jetzt korrekt zusammengefasst werden.

v2.6.0

Release Date: 2025-06-17

Prelude

Upgrade Guide 2.x -> 3.0

Es ist nun möglich Zusatzfelder in der Winline zu definieren die Produkten die Möglichkeit gibt, Teil einer dynamische Produktgruppe zu sein und welche Produktgruppe bei dem jeweiligen Artikel angezeigt wird.

Neue Funktionen

- SH-165 Logkonfiguration per logging.cfg
- SH-173 Zuweisung von mehreren Sales-Channels über Artikeleigenschaften.
- SH-104: Großes Refactoring. Aufteilung der Auszuführenden Dateien und Programmstart von Logic getrennt. Umbenennung von einigen Klassen und Dateien um die Lesbarkeit zu verbessern.
- In der config.ini kann das Feature DYNAMIC_CROSS_SELLING aktiviert und mit den Werten in dem Block [DYNACROSS] konfiguriert werden. Die Standartwerte sind hier:

  cross_name = Unsere Vorschläge cross_gruppe_feld = C229_Zusatzfeld29 cross_anzeige_feld = C230_Zusatzfeld30

Des weiteren wurde dem Block [CROSS] der Wert "crossselling_aktiv = True" hinzugefügt wodurch diese Funktion ausgeschaltet werden kann

- Kundenexport kann nun auch mit Mapping durchgeführt werden. Die erzeugte Datei ist XML-EXIM kompatibel.

Nicht mehr unterstützte Funktionen

- Shopware 5 und WooCommerce werden nicht mehr Unterstützt.

Fehlerbehebungen

- SH-172: CMSPageId muss nicht mehr in der Konfig hinterlegt werden und die CMSPage bestehende Kategorien werden nicht überschrieben.
- SH-177 Ermittlung des Shoppreislistentyps anhand der Preisflag anstatt der Kontonummer, um auch im Fall von fehlerhaften Preisdaten in der WinLine diese übertragen zu können.
- SH-179 Die erste Preisliste aus dem Konfigurationsfeld WL.preislistennr wird als Fallback-/Allgemeiner-VK-Preis verwendet.
- SH-196 Upgrade auf PyInstaller für python 3.11 Kompatibilität.
- SH-199 Anlage von Varianten ohne gesetzte Bezeichnung in der WinLine.

  kleiner bugfix: options id wird aus langtext generiert und muss daher nur den namen ersetzen 'Nicht gesetzt' ohne eine neue uuid zu generiert.
- SH-205 Exception für ENV-Fehler inklusive Traceback ausgeben.
- SH-206 Artikel ohne WinLine-Bezeichnung wurden nicht übertragen.
- SH-207 Logging Beispielconfig funktionierte nicht.
- SH-207 Logging für http.client wenn debug-Modus angeschaltet wurde.
- SH-144 Umstellung im Mapping auf Dynamische Names erkennung anhand von Order_Line_Item Type. So können Bundleartikel und andere custom Artikel ohne anpassung gemapped werden. Zudem kann für jeden Typ eine zusätzliche Textzeile gemapped werden.
- Ausprägungen übertragen funktioniert jetzt mit shopware6
- SH-190 - Statt einem search der alle daten enthält wird nun standardmäßig ein search mit limit verwendet. Hintergrund ist eine Änderung in Shopware die Ergebnisse limitiert aber nicht richtig sortiert bzw. den User informiert. Beispiel: Der default für ein search nach allen Kunden gibt nicht wie vorher alle Kunden zurück sondern im Standard nur 500 Kunden (das ist ein limit welches nur über eine Datei auf dem Webserver von Shopware

verändert werden kann) und diese sind nicht sortiert wodurch nicht alle Kunden abgerufen werden können, selbst mit einer Zeitstempelfilterung.

- Es werden nun nicht zugeordnete Kategorien entfernt und zudem wird nur die Unterste Ebene des Kategoriebaums dem Artikel zugeordnet.
- SH-217: Da der Status über den abgerufen werden soll fix auf einen gesetzt war konnten nicht zwei stati gleichzeitig gefiltert werden. Zudem ist die Dynamisch Status Änderung nun flexibler und aufgeräumter
- SH-222 Es wurde im Trackingnummern parameter die Bestellnummer als Dict übergeben statt der Trackingnummer.
- SH-223 Wenn kein Gast WinLine Konto hinterlegt ist wird die Kundennummer einfach übernommen.
- auf config variablen (cross_gruppe_feld, cross_anzeige_feld) wird nun nur noch zugegriffen wenn diese auch existieren
- Artikeluntergruppen werden jetzt mit deutsch in allen Sprachen gefüllt.

Sonstiges

- SH-209 Bei abweichender Ebene in Kategorieübersetzungen, wird die betroffene Kategorie ausgegeben.
- Changed Pythonversion to 3.11
- Added new feature to generate price from netto or brutto. with the addition of the new features a method to check if mutually exclusive features are active at the same time was added. added missing .value for the exclusive features list. set base_price to art_object.C007_allgVerkaufspreis when uvp feature is selected bandaid fix für determine_brutto_netto_faktor bei price_data bis pricelist_entry implementiert ist
- mit PRICE_PRIORITY_HIGHEST = True kann das Feature genutzt werden mit dem der Preise nach der Preispriorität mit '(Int32) 2147483647 - Preis' ausgewählt wird Bugfix weil aufs enum und nicht die value aus dem configfile geguckt wurde validatorenmatrix eingefügt
- Mapping update für Zusatzfelderlabel

v2.5.0

Release Date: 2024-11-01

Neue Funktionen

- SH-170 Die SSL-Validierung wird in Entwicklungsumgebungen automatisch deaktiviert, um mit lokalen Shopware-Instanzen per API kommunizieren zu können.

Fehlerbehebungen

- SH-167 Bessere Fehlermeldung bei falscher Verwendung von default im Mapping.

v2.4.2

Release Date: 2024-10-23

Fehlerbehebungen

- SH-157: Menge2 Berechnung ist nun korrekt und berücksichtigt beim berechnen des nächsten Steps den Umrechnungsfaktor. Varianten Plugin für Wittstock übergibt nun korrekt die Optionen.
- SH-160 Optimierung des Artikelabgleichs während der Archivübertragung (Bilder und Downloads).
- SH-164 asList: True wurde für Folgeelemente im Mapping nicht zurückgesetzt

Sonstiges

- SH-146: Existierende Kunden werden nun zunächst gemapped und an das XML aus dem Shop angehängt. So kann zum einen im XML Refactoring darauf zugegriffen werden und zum anderen können im XML import die Default werte überschrieben werden.

v2.4.1

Release Date: 2024-10-14

Fehlerbehebungen

- SH-151 Multikategorien aus mehreren Eigenschaftsfeldern wurden nicht korrekt aus der Konfiguration ausgelesen.
- SG-153 Die Preissynchronisierung prüft ab sofort auf in Shop vorhandene Artikel anhand der UUID.
- SH-154: Deferred Felder werden nicht mehr durch _add_parent_columns hinzugefügt, da dies sonst crashen kann.

v2.4.0

Release Date: 2024-09-26

Neue Funktionen

- Plugin für gen_article_uuid erstellt. Im Beispiel wird die UUID aus einem Zusatzfeld im Artikel ausgelesen.
- SH-121: EXIM lädt jetzt die Daten korrekt

Fehlerbehebungen

- SH-145 Preisübertragung ohne Menge2 korrigiert.
- Fixxed defaultconfig import_sequences invalid json
- Sentry-Sdk auf version >2.8 geupdatet (Safety Issue) CVE-2024-6345 auf Ignore gesetzt, weil die neuste Pipenv-Version nicht kompatible ist und diese eine Abhängigkeit zu Setuptools hat SH-123 Testpipeline Läuft wieder durch, da einige Tests auf skip gesetzt wurden die derzeit nicht erfolgreich laufen können.
- SH-130 Bis menge wurde zweimal um 1 reduziert
- SH-133 Fehlende Beispiel Dateien in die SPec eingefügt

Sonstiges

- Mapping Testordner initialisiert
- git ignore angepasst
- SH-128: Debug für Request kann über die Config umgestellt werden [['DEFAULT'] debug=True]{.title-ref}
- Pipfile source als erstes bei pypi
- SH-111 get_last_changed_article_date plugable
- SH-140 Beispielmapping für Artikel mit den meisten Feldern die im Shop zur verfügung stehen.

v2.3.0

Release Date: 2024-08-08

Neue Funktionen

- SH-120 - Menge2 kann nun für die Preisberechnung berücksichtigt werden und wird anhand von dem Umrechnungsfaktor aus der Menge2 und den entsprechenden Collis übermittelt werden. Gesteuert wird dieses Feature über die Feature-Flag in der Konfig "MENGE2 = True" (Fallback ist False)
- SH-99 Die Eigenschaften die an den Shop übermittelt werden sollen, können nun in der WinLine konfiguriert werden und mithilfe des wittstock plugins als Varianten typen verwendet werden.

Fehlerbehebungen

- SH-112: Mehr Plugable Funktionen
- SH-114: Wenn keine Übersetzung für die Eigenschaften im Mapping definiert ist, wird es übersprungen.
- SH-75 Behebt Probleme beim Abrufen der Bestellungen.

Sonstiges

- Der Beleg-Lookup kann deaktiviert werden.

v2.2.1

Release Date: 2023-07-26

Sicherheits-Updates

- Führt Sicherheitsrisiko 53326 ein, welches zunächst ignoriert wird.

Fehlerbehebungen

- SH-66 Eigenschaften werden übersetzt und durch den Mapper erweitert.
- SH-72 Wenn der Cache ordner nicht existiert wird dieser angelegt.
- SH-62 Fixes inkompatibilität mit neueren Werkzeug Versionen

v2.2.0

Release Date: 2023-06-22

Prelude

Sammelupdate für Shopware 6

Neue Funktionen

- # 429 - Generelle Grundfunktionen für Shopware 6 Funktionieren nun.

  Stammdaten, Bilder, Preise.

- Filter auf existierende Artikel im Webshop bei der Preisübertragung

- # 444 Daten für das Management-Interface werden Userverzeichnis

C:UsersXXXAppDataLocalnewenergy gmbhShopware Management Interface zwischengespeichert.

- # 542 Mapping-Dateien können nun über die erweiterte Ansicht

bearbeiten werden

- # 546 - Ist die option 'skip_inaktiv_variants' in der Konfiguration

gesetzt, werden Inaktive Ausprägungen nicht übermittelt. Zudem werden Varianten in Shopware entfernt die es nicht mehr in der WinLine gibt (oder wenn diese Inaktiv sind).

- # 551 Über den Aufruf

'self.umsetzungstabelle('\<umsetzungsname>', wl_data)' kann ein Wert Transformiert werden, anhand der .json datei.

- # 575 Es kann in der Umsetzungstabelle nun ein Wildcard Symbol

verwendet werden (Default: '*'). Beachte es wird nur 1 Vorkommen berücksichtigt.

- ◦ SyncOrders: Batch-XML
- ◦ XML-Refaktorierung: Kopieren von anderen Elementen mit Bedingung

- ## 606 XML-Refaktorierung nun an weiteren Stellen möglich

- ## 607 - Es kann nun über das Flag -on bzw. --onlyNew gesteuert

werden, ob Artikel im Shop im syncProducts nur angelegt (Flag gesetzt) oder überschrieben werden sollen (Default).

- ## 609 Es kann im Mapping nun auf jede Ausprägung des Hauptartikels

zugegriffen werden. Im Beispiel wird dies benutzt um für den Hauptartikel die Lagerwerte aus einer spezifischen Ausprägung zu übermitteln.

- ## 610 Wird eine Artikelnummer beim übertragen angegeben und es

handelt sich um einen Artikel mit Ausprägungen, werden alle Ausprägungen mit aktualisiert.

- ## 611 - Es können nun Gutschriften importiert werden wenn ein Status

in Shopware für diese Hinterlegt wird.

- ## 623 - ID-Lookup-Liste für Artikel und Varianten, damit diese nicht

einzeln aus dem Shop abgerufen werden müssen

- ## 625 Artikelnummern mit Sonderzeichen werden nun unterstützt

- ## 636 Neuer Sync: Nachträglich in der Winline geänderte Versandart

  können nun an den Shop synchronisiert werden

- ## PA-34 Die Generierung von Code für Plugins ist nun dynamisch und

  muss nur noch minimal und wartbarer angepasst werden.

- ## SH-17 - Das übertragen der Tracking-Nummern ist eine separate

  Funktion, damit diese vorm Übertragen der Shipment und des Orderstatus passieren kann.

- SH-23
  - Statusupdate-Tracking: Wenn in der Konfiguration der Wert "orderstatus_tracking" auf "-1" gesetzt wird, wird beim Aktualisieren der Versand-Trackingnummer der Bestellstatus nicht verändert - Storno: - Teillieferscheine: - Datum-Filter:

Bekannte Probleme

- Nur für Artikelbasisdaten Übertragung, da für Archivimages und Crossselling noch zusätzliche Daten benötigt werden.

444 Die Datei jobs.sqlite aus dem Installationsverzeichnis muss vor

dem ersten Start in das Verzeichnis C:UsersXXXAppDataLocalnewenergy gmbhShopware Management verschoben werden.

- Kundensync Upgrade für den Kunden Ranberg: Nur für neue Kunden, die eine Bestellung aufgegeben haben, werden .xml-Files generiert.
- Ranberg-Plugin für den syncFilteredCustomers: Verzicht auf Timestmap. Nun wird geschaut, ob für die eingetragene Winline-Kontonummer bereits ein Konto in der Winline existiert.
- -syncFilteredCustomers: Erzeugte XML-Dateien können nun individuell gemappt und refaktoriert werden.
  - Copy-Value von einem anderen Tag
  - Einfache Value Zuweisung über die Refactoring.json
  - syncFilteredCustomers: Refactoring für die Winline-Belegart
- - Flexible/Rekursive add_tags.json, um aus den Values der bestehenden Tags aus der XML einen '|'-getrennten String zu erzeugen. Dieser kann im refactor_xml.json weiterverarbeitet werden, um "IF VALUE = X DANN VALUE = Y" zu beachten.
  - Artikelgruppen können als Schattenkategorien verwendet werden. Einstellung definierbar über die Config
  - Kombinieren von N-beliebigen Values aus einem XML in ein neuen Tag (syncFilteredCustomers)
  - Ranberg-Plugin: Ignore Schnellbesteller beim syncFilteredCustomers
  - XML-Refactoring: Den Tag-Values Präfix hinzufügen
  - XML-Refactoring: Leerzeichen hinzufügen

- ## 613 Kopieren von Werten nach der gleichen Bedingung nun möglich

- ## 634 Bestell- und Zahlungsstatus werden aus der Winline heraus

gesetzt

Fehlerbehebungen

- ## 463 Leerzeichen aus default Installationspfad entfernt. Der AppData

name wurde von "New Energy Gmbh" geändert zu "newenergy gmbh" um den tatsächlichen namen widerzuspiegeln. Die 'management.ini' wird nicht mehr überschrieben, wenn ein Update durchgeführt wird.

- Durch das update auf WLAPI 11.6.0 können SQL_Abfragen mit mehr als 2100 Parametern realisiert werden.

- ## 549 Ausgabe des Traceback wenn im Artikel eine Exception aufkommt.

- ## 550 Im Mapping kann nun auf das ShopAPI Objekt zugegriffen werden

- 'concat_string_shop' kann nun jeden datentyp der zu String konvertiert werden kann behandeln. Orderabruf überspringt nun keine Orders bei denen es Artikel gibt die nicht in der WinLine vorhanden sind.

- ## 553 Optionen bei denen alle Varianten inaktiv sind werden nicht

mehr im Shop gesetzt.

- ## 556 Problem behoben wenn ein Hauptartikel mit Ausprägungen

(Artikeltyp: 1) nur genau eine Ausprägung hat.

- If-Abfrage zur Ausführung des Hooks beim Plugin filteredCustomersSync oder des eingebetteten Codes war fehlerhaft.

- ## 567 - Update auf neue WLAPI version benötigte umstellungen im

import und den Datenbankverbindungen

- ## 570 Durch das Entfernen der Kategorie-Eigenschaft und der

root-category-id werden nun die im Shop gesetzten Kategorien nicht überschrieben.

- ## 571 Problem behoben wenn die Kategorieeigenschaft-Optiomn eine

Liste enthält

- ## 606 Leerzeichen-Bug behoben

- ## 612 - Fehler behoben bei dem am ende der Preisübermittlung nur die

Varianten-Preise geprüft wurden um den erfolgreichen Programmdurchlauf zu ermitteln.

- ## 616 - Wenn in der Config keine Preisliste steht wird eine Leere

liste geladen daher müssen die alten SQL Funktionen anders prüfen ob der wert vorhanden ist.

- ## 618 - Update auf Patchversion 11.9.1 von WLAPI um Bug mit der HSL

zu beheben.

- ## 619 XML-Refactor: Leere Tags werden auf None gesetzt

- ## 620 - Update auf WLAPI v11.9.2 für Performanz-Optimierung

- ## 621 - Die Preise der Varianten werden nun nicht mehr übertragen

wenn diese im Shop schon vorhanden sind. Hier gab es einen Fehler der das Aktualisieren der Varianten verhindert hat.

- **627 Fix für den Kundenexport falls ein None-Wert gelöscht werden**

soll

- **628 Timestamp für den syncFilteredCustomers wird nun von dem**

aktuellsten Kunden übernommen

- **629 - Fehlerfall in dem Beschreibung gar nicht gesetzt ist**

- **PA-35 - Fixes Issue with parameters**

- **PA-4 - syncCustomersFiltered erweitert damit die Attribute**

(Freitextfelder) mit in die XML geschrieben werden können.

- **SH-10 Installer-Template update auf neuste version und alte config**

aufgeräumt

- **SH-11 Bestellungen werden erst ab konfiguriertem Datum**

berücksichtigt. Zeitstempel können einheitlich über die Config verwaltet werden.

- ## SH-14 Wenn die Textfelder, über die die Bestellungen für den

Statusabgleich gemached werden, werden diese nun korrekt raus gefiltert.

- ## SH-15 Flask version muss erst mal fixiert sein.

- ## SH-25 - Behebt ein Problem Belege in der WinLine zu finden, wenn

die Shopbelegnummer vom Mapping verändert wird. Die Lösung ist ein Config eintrag in dem die gleiche Transformation hinterlegt ist. Dazu kann beim WinLine-Beleg Abruf eine weitere Einschränkung aktiviert werden. Durch das setzen der Einstellung "gutschrift" in einer Import Sequenz werden nur WinLine-Belege mit negativem Endbetrag (Gutschriften) für den Vergleich herangezogen.

- ## SH-28 - Der Parameter -a/--all wird nicht korrekt an die Aufgabe

im Interface angehängt.

- ## SH-29 - Filterparameter wird wieder korrekt mit StatusId gefüllt.

Das erzeugen der Suchparameter für die Shopware REST-API wurde überarbeitet und erlaubt es nun für jeden Filter den Vergleichsoperator und den Vergleichstyp zu setzen.

- # SH-31 Durch die neue Einstellung "ignore_missing_wl_konto" kann

nun übersteuert werden wenn kein WinLine Personenkonto gefunden wurde. Diese Einstellung sollte nur beim XML-Bestellimport verwendet werden, da sonst der Webservice die Bestellungen als Fehlerhaft ablehnt.

- # SH-32 Nachdem auf der XML Datei gearbeitet wurde, werden alle None

Texte zu leeren texten umgewandelt.

- # SH-36:

  - Filter auf Datum werden nach Shopware 5 Dateformat formatiert
  - bei fehler beim encoding der XML wird nun jede Exception gefangen anstatt zu einem

Programmabsturz zu führen.

- # SH-37 - Wenn die XML Datei zu einem Kunden nicht

geschrieben/erzeugt werden kann wird eine Warnung im Logger geschrieben und eine extra datei mit den fehlgeschlagenen Kundennummern oder IDs geschrieben.

- # SH-38 Behebt Fehler bei dem die locale Variable die ober geordnete

Variable überschreibt. Es wird der Orderstatus sowohl aus dem Auftrag als auch dem Lieferschein gelöscht.

- ## SH-46 - Behebt Duplikate im erlaubten Protokoll

- ## SH-54: Mappinghelper nutzt nun einen YAML loader dadurch kann

  leichter auf yml umgestellt werden aber json wird weiterhin unterstützt.

- SH-63 - Wenn keine Steuer bzw. Steuersperre hinterlegt ist, wird 0 Prozent genommen.

- ## SH-8 Behebt crash wenn firstname oder lastname None sind.

- fixes pyinstaller hidden import für sqlalchemy.sql.default_comparator

Sonstiges

- ## 441 - Aktualisierung der Templates

- Sonderprogrammierung für direkte Tabellenübertragung von WinLine zu Shopware

- ## 489 - Dokumentation Kundenimport

- ## 501 - Update von SQLAlchemy auf Version 1.4.18 Update von WLAPI auf

  Version 11.6.0

- ## 545 Die Änderungshistorie wird für die Dokumentation automatisch

  erzeugt.

555 Log-Level für das Fehlschlagen von AsyncIO requests war

vertauscht. So wurden bei einem Retry Fehler als Exception geloggt obwohl der darauf folgende retry erfolgreich war. Gleichzeitig wurde ein fehlschlagen nach dem alle Retries verbraucht waren nur in Debug-Level geschrieben.

563 - Dokumentation für Ausprägungen

565 - Dokumentation Sendungsverfolgung und Statusupdate

- ◦ Downgrade auf APScheduler 3.8.0, weil neuere Version verbuggt ist.
- ◦ Separator für das Refactoring der erzeugten XML-Dateien von "-" zu "|" geändert
- ◦ Pfad für XML-Preferences über Config

579 - Dokumentation für die Umsetzungstabellen des Mappings

580 - Dokumentation für den Kundenimport (Plugin, xml-refactoring,

Sammelkonten)

- ◦ Timestamp intial für syncFilteredCustomers
  ◦ Initialisierung des Strings für das Refactoring der Kunden XML-Dateien in die for-Schleife gschoben

583 - Generelle Aktualisierung der Dokumentation

- ◦ Änderung der Belegstatus bei Batchexport
  ◦ Config für Batchexposrt auf True/False gesetzt
- ◦ Doku für Batch-Order

- # 592 - Dokumentation Aktionserver-Restart script

- # 594 - Über die Config kann nun gesteuert werden ob

Schnellbestellungen über das Sammelkonto gezogen werden oder als eigene Konten behandelt werden.

- ◦ Auto-Scroll für den Live-Log

- # 603 Sequenz-Diagramm für die Statusupdates

- # 608 Update WLAPI zu Version 11.9.0. Es kann nun über die Property

v021_Artikelview_Auspraegungen auf ein Query zugegriffen werden um all Ausprägungen zu einem Artikel zu erhalten. Dies funktioniert auch aus einer Ausprägung heraus.

- Update auf WLAPI Version 11.9.3

- # 635 - Gitlab-CI hat ein altes Image gezogen. Neues image sorgt

dafür, dass die pipenv in Flake8 nicht neu installiert werden muss

- ## PA-35 - Neue Funktion um RTF zu Plaintext zu Konvertieren.

- ## PA-40 - Extra Route für Kunden- und Bestellungsabruf in einer

Kachel für Management Interface

- ## SH-18 Update auf WLAPI version 11.10 für dynamische

U-Felderweiterung

- ## SH-19 - Bestellungen können nun in SW5 in mehrere Stati nach

einander gesetzt werden. So kann z.B. erst der Zahlstatus und anschließend der Bestellstatus gesetzt werden.

- ## SH-20 - Einfacher Verlauf der Statusupdates welche 1 Woche zurück

gehen.

- ## SH-42 - Neue Route die erst die Kunden und anschließend die

Bestellungen abruft

- ## SH-43 - Der Text(Sprache) der aus den Eigenschaften übertragen

  wird, kann nun frei Konfiguriert werden.

- ## SH-48 Plugin um Rabattleisten als Kundengruppen zu übertragen.

- SH-61 Artikeluntergruppen als Kategorien funktioniert wieder mit Shopware 6

- ## SH-9 Anlage Dokumentation Leitfaden

v2.1.0

Release Date: 2021-09-21

Neue Funktionen

- ## 321 - Für den Abruf der Artikel aus der WinLine wird nun

  abwärtskompatibles SQLAlchemy verwendet. Die alte Funktion bleibt erstmal bestehen, sollte aber in einer späteren Version entfernt werden.

- ## 456 - Plugin-System implementiert. Im Installationsverzeichnis

  können im ordner 'plugins' die Preisfindungslogik und die Kundengruppenlogik per Plugin verändert werden.

- ## 457 - Über das Mapping kann nun ein Feld aus der WinLine als Liste

verarbeitet werden. Hierzu muss das Feld '_options' die Option 'asList: True' enthalten.

- ## 458 - Es können nun Ausprägungsgruppen explizit ausgeschlossen

werden für die im Shop keine Varianten angelegt werden.

- Kunden aus dem KontaktenStamm in der Winline-Datenbank können nun in die Shopware importiert werden. Neben der Neuanlage werden auch geänderte Daten in die Shopware übertragen.

- ## 475 - In der Config können nun mehrere Zusatzfelder hinterlegt

werden in denen Pfade zu extra Bildern hinterlegt werden können. Diese Pfade sind die Pfade in das WinLine-Archiv1, wie die Grafik-Datei in den Texten.

- Der Timestamp-Config ist nun über das Interface editierbar.
- Neue Methode zur Abfrage von Web-Personenkonten über das Zusatzfeld, welches in der Config deklariert wird.

- ## 488 Neuer Config-Parameter xmlexport_customers. In das hier

angegebene Verzeichnis werden die Dateien für den WinLine-Kundenimport abgestellt.

- ## 491 - Trackingnummern werden nun zurückgeschrieben sobald diese in

den zugehörigen Lieferscheinen hinterlegt wurden.

- ## 497 - Es können nun Archivbilder explizit den Varianten in Shopware

  zugeordnet werden.

- ## 498 - Übermittelt Trackingnummern aus Shopware in Lieferscheine in

  der WinLine.

- Die Daten der Varianten können nun auch über das Mapping gefüllt werden. Dafür muss im Mapping das Segment "Variant" angelegt werden.
- Validator mit dem geprüft werden kann ob mit der aktuellen Konfiguration alle Webartikelnummern verlustfrei Konvertiert werden können.
- Mapping wurde der Übersichtlichkeit halber uin mehrere Dateien aufgeteilt. Welche Dateien geladen werden, wird über die Index.json gesteuert.

Bekannte Probleme

- ## 494 - Images can not directly transmitted to WooCommerce.

- ## 508 - Nicht alle Bilder werden gesetzt.

- Fälschlicherweise erkannte umlaute werden zu Umlauten gewandelt. Ist in einer Artikelnummer die Zeichen folge 'ae', 'oe' oder 'ue' werden diese immer zu 'ä',ö und ü gewandelt auch wenn diese eigentlich keine Umlaute repräsentieren.
- Fehler behoben durch den der Mapper Crashed wenn ein Segment fehlt.

- ## 542 - Durch den Umbau funktioniert die Bearbeitung des Mappings aus

  dem Management Interface nicht mehr.

Update-/Installationshinweise

- Zusatzfeld für die ShopwareID nun über die Config deklarierbar.
- Bei Änderungen der Kontaktdaten wird der Name des Kunden und der Name auf der Rechnungsadresse aufeinander abgestimmt.
- Erkennung der ShopwareID von Personenkonten in der Winline nun über ein Zusatzfeld. Deklaration in der Config notwendig.
- Kontaktesync: Preislisten werden berücksichtigt sofern die dazugehörigen Kundengruppen im Shopware angelegt sind.

Fehlerbehebungen

- Nach dem Starten eines Syncs über das Management-Interface tut sich nun was. Der Progress-Bar funktioniert wieder. Das Öffnen der Log-Dateien ist auch gefixt.
- Fix der XML Kundenexports. Der Datensatz wird nun einwandfrei aufgebaut.

487 - Fixxes Progressbar bug welches den text anhand von

sonderzeichen splitted welche nicht enthalten sein dürfen.

- compatibility issues with WooCommerce fixxed.

496 - Bugfix für den die URL parameter immer wieder an den Aufruf

angehängt wurden. So kam es zu Doppelungen.

499 - Keine doppelten Optionen an den Shop übermitteln, da dieser

sonst in einen Fehler läuft.

- ## 503 - Kommas in Artikelnummern werden nun ersetzt

- ## 503 - Nummer wurde nach jedem ersetzen wieder zurückgesetzt.

- ## 504 - Beim dem Abrufen der Artikel aus dem Shop werden die

Artikelnummern wieder zurück gewandelt. Das ersätzen von Zeichen die nicht in Shopware erlaubt sind kann nun in der Konfigurationsdatei über ein Tupel von Tupeln gesteuert werden.

- Präventiver Fix für den MemoryError.

- ## 506 - Da die Option aus dem Mapping entfernt wurde, wurde bei dem

nächsten Artikel die Liste falsch verarbeitet. So wurde immer nur der erste Artikel korrekt übertragen. Durch das Kopieren der Felder wurde dieser Fehler behoben.

- ## 507 - Abhängigkeit der Webpreisliste in den Einstellung behoben.

Leer lassen hat zu einem fehler bei der Artikelübertragung geführt, war jedoch notwendig um Preise aus mehreren Preislisten zu übertragen.

- ## 508 - Durch das umbenennen einer Variable kam es zu Konflikten wenn

die Kundennummer aus dem Shop nicht abgrufen werden konnte.

- ## 513 - Es werden nun nur noch Ausprägungen in Shopware angelegt die

existieren anstatt alle Ausprägungsoptionen zu einer Gruppe.

- ## 514 - Anpassung an WLAPI release für Relation die umbenannt und

keine Liste mehr ist.

- ## 516 - Unnötige Logger ausgaben bei der Artikelnummer konvertierung

wurden entfernt.

- ## 519 - Varianten-Liste konnte None enthalten, was beim übertragen in

eine Exception gelaufen ist.

- ## 520 - Die ID Liste enthält bereits die in die Ursprungs form

gewandelten Artikelnummern und daher müssen die Nummern nicht gewandelt werden.

- ## 521 - Pfad muss erst gejoint werden.

- ## 522 - Artikelnummer wurde an einigen stellen von Sonderzeichen

befreit an denen es nicht gemacht werden durfte.

- ## 526 - Wenn die Variant-Daten die an den Shop übermittelt werden

  keine Preise enthalten, werden die Varianten aus dem Artikel entfernt. Um dies zu umgehen werden die bestehenden Preise aus dem Artikel abgerufen und ggf. dann überschrieben.

- ## 529 - Hier wurde die Methode raise_for_status nicht aufgerufen und

  somit konnten keine HTTP errors gefangen werden.

- ## 530 - Artikel_Data mapping nachdem die Eigenschaften abgefragt

  wurden um auf diese zugreifen zu können.

- ## 543 - Die neue Mapping Struktur wird nun in den tests verwendet.

- ## 544 - Fehlender Config-Eintrag für Tests nachgetragen.

Sonstiges

- Kunden Configs und Mapping im Repo um versions Tracking zu haben.

- # 473 - Logging für Pluginaufrufe ergänzt.

- # 476 Update Template-Projekt auf Version 0.4.4.

- Inital Configs für das Interface und die ess.exe.
- Extra Ordner für Kundenspezifische Plugins.
- Beispielmapping für features von Rannenberg.
- Default Preis für Varianten wird nun unterdrückt wenn das PreisSync-Modul verwendet wird.

- # 527 - print für folder

- Beispiel-Mapping um Werte direkt aus den Eigenschaften zu mappen. Dies ist für den Eigenschaftstyp Bool (checkbox) nötig.
- Major releases werden nur noch manuell über das segment 'major' erstellt.

v2.0.0

Release Date: 2021-06-09

Neue Funktionen

- Aufgabenplaner untersützt nun die Auswahl von Wochentagen zum Starten von Syncs.

- Aufgabenplaner untersützt nun die Auswahl von Wochentagen zum Starten von Syncs.

- Stücklisten können über das Mapping in Attribut-Felder vom Shop übermittelt werden. Dazu wurde das Beispiel-Mapping angepasst, dieses sollte an die Kundenwünsche angepasst werden.

- Neues Dashboard zur Anzeige der Informationen bezüglich des AP-Schedulers. JobEvents werden abgefangen und in einer Liste dargestellt. Zusätzlich wird eine Statistik erstellt.

- Starten eines Syncs leitet wieder auf die Hauptseite mit Live-Log. So ist gewährleistet, dass wenn der User die Seite neulädt der Sync nicht wieder getriggert wird.

368 Preisübertragung wird nun mit AsyncIO durchgeführt um den

Leerlauf zwischen den Antworten der Requests zu minimieren.

368 LRU_Cache ohne Limit für die Funktion

winlineapi.get_rabatt_from_rabattmatrix(). Diese Daten werden wärend des Preissyncs nicht verändert und sind simple Lookups in Tabellen (Int, Int) -> Float.

368 Die Anzahl der maximalen parallelen TCP-Verbindungen kann über

die Configuration gesteuert werden.

Konsolenausgabe und Ausgabe in .prof-Dateien.

392 Optionale Fehleranalyse mit Sentry. Wird ein Logging-Server in

der Konfig eingestellt werden Fehlermeldungen automatisch zur späteren Analyse übertragen.

- Es ist nun möglich einen einzelnen Sync mit Custom-Start-Parameter zu starten. Zusäztlich werden Sync-Optionen automatisch erkannt und müssen nur noch mit in die management.ini aufgenommen werden, um diese zu aktvieren.
- Bilder mit Transparenz können nun auf eine, in der Config.ini definierten, Hintergrundfarbe gesetzt werden. Dies sorgt dafür, dass Bilder mit Transparenz im Fall von PNG nicht auf automatisch auf schwarzen Hintergrund gesetzt werden.
- Im Mapping kann nun im eval über self.wlapi.systemhelper.convertISOtoCountryCode(wl_data) der WinLine Ländercode ausgelesen wird.
- Beim Abrufen der Artikel aus der WinLine kann anhand eines Zusatzfeldes gesteuert werden ob es sich um einen Hauptartikel oder um eine Variante handelt. Anhand dieser Informationen können Bestände und Preise korrekt an den Shop übermittelt werden.

- Über das Config-Feld 'Gruppe_Shopkunden_nicht_ueberschreiben' kann gesteuert werden ob die Shopware-Kundengruppe 'Shopkunden' für das Erzeugen der Kundengruppen ausgeschlossen werden soll. Dies sorgt dafür, dass Shopkonten mit der Kundengruppe 'Shopkunden' nicht überschrieben werden und nur den Standard Preis für alle Artikel bekommen.
- Beim Abruf der Kundennummernliste aus dem Shop können optional nur die aktiven Kunden zurückgegeben werden.
- Im Mapping kann nun auf jedes Feld in der Bestellung aus Shopware zugegriffen werden.
- Shopware-Kudendaten-Export als XML-Files. Winline-Kundennummer export, wenn diese als Shopkunden gekennzeichnet sind.

452 Sentry-Config in App-Data und User-Data berücksichtigen.

- Personenkonten-Cache für den Preis sync um die SQL-Abfragen zu reduzieren.

Bekannte Probleme

- Die Bestände sind noch nicht korrekt, dafür muss jedoch die WLAPI-Bibliothek ein neues Update bekommen.
- Momentan werden von der WLAPI-Bibliothek noch die Standard-Varianten ausgeschlossen, daher muss auch hier auf ein Update gewartet werden. Dieser Bug ist durch eine Änderung der WinLine entstanden.

368 Logging überarbeitet um Datenmüll zu veringern wenn Debug-Level

DEBUG (-ddd) verwendet wird.

- fixes #448

Update-/Installationshinweise

- Maximale Prozessdauer ist über die management.ini einstellbar.
- Durch das Update auf Wlapi version 11.0.6 Kann nun die Enum für das FlagZeilenrabtt in der Preisliste verwendet werden.
- Update von WLAPI auf die Version 11.0.5
- Update auf WLAPI Version 11.0.6 für die Unterstützung der Bestandsberechnung für Handelsstücklisten mit und ohne Dispositionen.

- ## 437 Die Beispiel-Konfigurationsdatei wurde überarbeitet und

  Kommentare für die Optionen wurden vervollständigt. Dazu gibt es einen Validator der anhand von Migrationen prüft ob die Konfigurationsdatei auf dem neusten stand ist. Zusätzlich Hotswapped diese die Felder und Sektionen damit die Schnittstelle abwärtskompatibel bleibt.

- Update changelog template to version 0.2.2 https://gitlab.newenergy-bremen.de/basislager-developer/templates *(https://gitlab.newenergy-bremen.de/basislager-developer/templates)*

Nicht mehr unterstützte Funktionen

- ## 368 woo_wl.update_article_prices() sollte nicht mehr verwendet

  werden, stattdessen die neue Funktion run_article_price_update_async() verwenden. Durch den Parameter [-os]{.title-ref} kann der alte Preissync verwendet werden.

- ## 437 Einige Config-Felder sind aus der Konfigurationsdatei entfernt

  worden.

Kritische Fehler

- ## 413 Wird der Produkt Sync ohne Parameter gestartet so wird kein

  Artikel geupdated und kein timestamp zurück geschrieben.

Fehlerbehebungen

191 Libreoffice beenden wenn die Konvertierung einen Fehler läuft.

- Wenn ein Sync länger als 24h dauert, dass das Management Interface stoppt.
- Python 3.7 unterstützt noch nicht die annotation @cache. Diese ist erst ab Python Version 3.9 verfügbar. @lru_cache(maxsize=None) ist equivalent vom verhalten aber ein bisschen 'größer'
- Die WinLine behandelt in der Preisliste bei dem FlagZeilenrabatt 0 und NULL gleich daher mussten die Abfragen um diesen Fall erweitert werden
- Entfernt Punkte aus dem Dateinamen der an Shopware übermittelt wird, damit dort nicht eine falsche Dateiendung ermittelt wird. Zusätzlich wird der Fehlerfall nun richtig behandelt, damit die Übertragung nicht Komplett abstürzt.
- Bilder Ohne Alpha Channel sind in einen Fehler gelaufen udn werden nun abgefangen.
- Behebt einen Fehler der den Fallback zum Sammelkonto verhindert.

438 Behebt einige Bugs mit Varianten und Hauptartikel mit Varianten

für Bestand, Preis und Produkt Synchronisation

- pipfile aktuallisiert für Flask-Profiler

446 Problem mit der Default Config behoben im Bezug auf den

Validator sowie fehlerhafte Aufrufe auf Klassenvariablen

- Vergleich Kundengruppe und Kontonummer schlug immer fehl da es sich um verschiedenen Datentypen handelt. Dadurch wurde die Rabattmatrix für Kundengruppenpreise falsch erzeugt.

462 - Filtert nun die tatsächlichen Chargen aus der Liste der

Ausprägungen herraus

- Überträgt nun nicht mehr alle Datensätze einzeln für jeden Chunk der fehlgeschlagen ist, sondern nur noch die Datensätze aus dem Chunk.

Sonstiges

- **237 Fehlerbehandlung wenn die Requests fehlschlagen. Prinzipiell**

  kann die Schnittstelle erstmal weitermachen selbst wenn ein Request fehlschlägt.

- Das Bearbeiten von wöchentlichen Syncs ist noch nicht möglich
- Das Bearbeiten von wöchentlichen Syncs ist noch nicht möglich
- Dokumentation in Pipeline korrigiert, requirements.doc.txt enthielt die falschen Werte.

- **415 In der CI keine neuen Environments pro MR erstellen, da die**

  Deploymnets akuell nicht funktionieren.

- Dokumentation zu LibreOffice Einstellungen die für die Schnittstelle nötig sind.

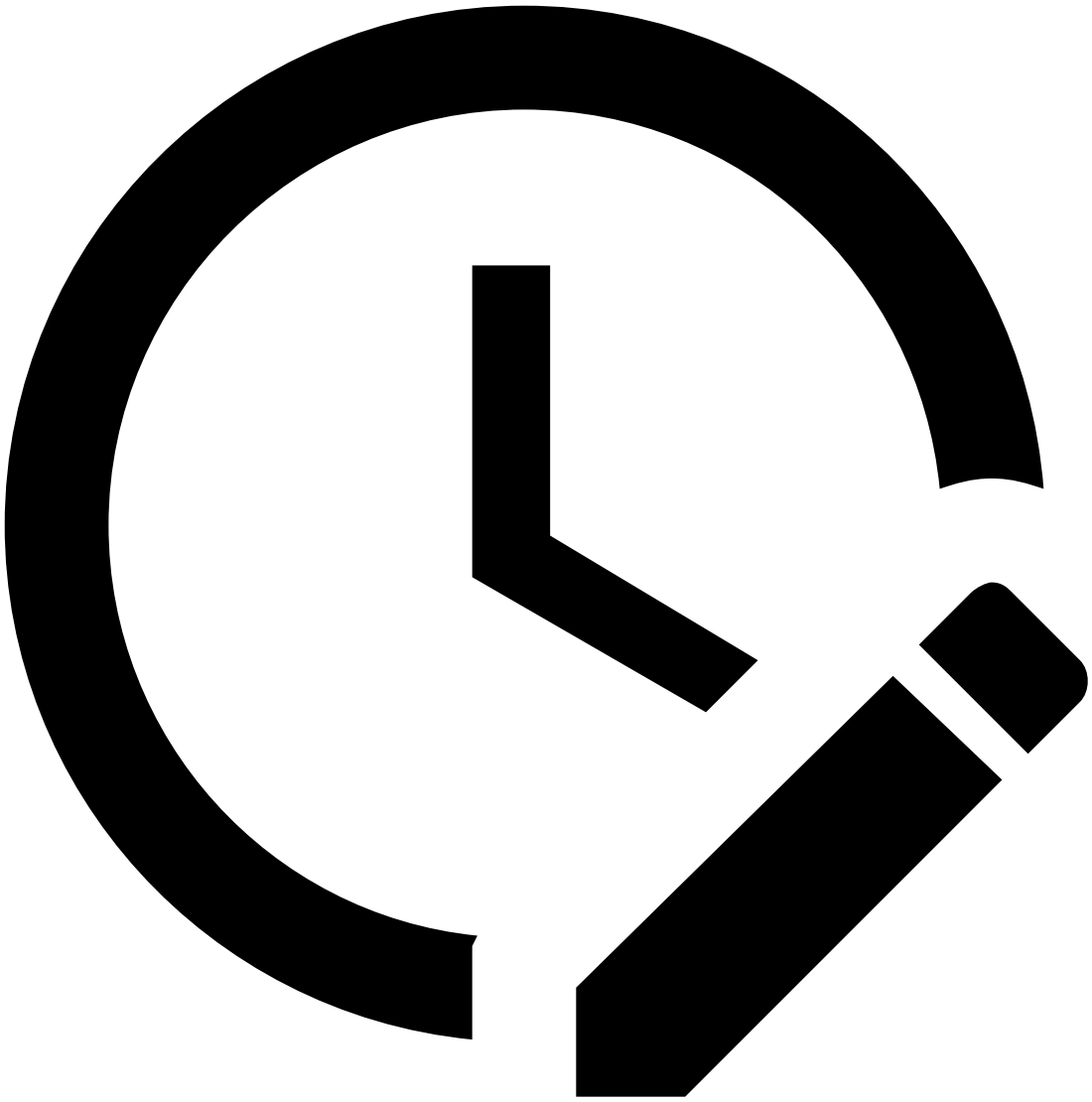
- **435 Dokumentation für Actionserver, Webservicevorlagen und minimal**

  Vorlagen für die Aufgabenplanung (sowie Beispiel-Aufgeben in XML form)

- **464 Integration von automatisiertem Release-Prozess inklusive**

  Meldung an Sentry.

- Changelogverwaltung überarbeitet. Neue Funktionen, Fehlerbehebungen oder Probleme werden automatisch erfasst und zusammengestellt.
- Dokumentation über den Action Server erweitert um die behandlung des Jahreswechsels in bezug auf den Beleg Import
- Produkt Synchronisierung überspringt Preise wenn syncPrices in der Konfig aktiv ist



2020-10-26