

shoptimax GmbH | Ulmenstraße 52H | 90443 Nürnberg

Anleitung smxDynNAV2OXID Modul

Stand: 05.03.2015

Version 0.9

Autor des Dokuments	Matthias Zistler	Erstellt am	05.03.2015
		© <i>shoptimax GmbH</i>	Vertraulich!

1 Inhaltsangabe

2 Historie des Abschlussberichts.....	3
3 Beschreibung.....	4
3.1 Funktionsumfang.....	4
3.1.1 Artikel.....	4
3.1.2 Preise.....	5
3.1.3 Kunden.....	5
3.1.4 Bestellungen.....	5
3.1.5 Kategorien.....	5
3.1.6 Bestände.....	5
3.1.7 Datenaustausch Möglichkeiten.....	5
3.1.8 Masterdata.....	5
3.2 Erweiterbarkeit.....	6
3.3 Schaubild.....	6
4 Installation.....	7
4.1 Rahmenbedingen.....	7
4.2 Vorgehensweise.....	7
5 Konfiguration.....	8
5.1 Generierung der SOAP Klassen.....	8
5.2 Konfigurationsparameter.....	8
6 Betrieb.....	14
6.1 Starten der Schnittstellen.....	14
6.2 Log-Dateien und Error-Handling.....	14

2 Historie des Abschlussberichts

Version	Datum	Autor	Änderungsgrund / Bemerkungen
0.9	05.03.2015	Matthias Zistler	Erstellung

3 Beschreibung

Das Modul **smxDynNAV2OXID** ermöglicht den Datenaustausch zwischen Microsoft Dynamics Navision und OXID-eShop.

Durch das Modul ist es möglich, komplette Datensätze aus Dynamics Navision nach OXID-eShop zu importieren und in die Gegenrichtung zu exportieren. Dazu werden die Daten aus Dynamics Navision in einen **Mastershop** von OXID-eShop importiert, von dem alle **Subshops** die Daten erben. Die **Subshops (Mandanten)** entsprechen dabei den **Verkaufskanälen** aus Dynamics Navision. Genauso wird in die andere Richtung exportiert und die Informationen aus allen **Subshops** auf alle **Vertriebskanäle** abgebildet.

3.1 Funktionsumfang

3.1.1 Artikel

- Stammdaten
- Artikelbilder
- Artikelbilder mandantenspezifisch
- Attribute
- Verpackungseinheiten
 - Möglichkeit Verpackungseinheiten zu verkaufe
- Grundpreisangaben
- Verschiedene Sprachen möglich
- Artikel Titel
- Artikel Titel mandantenspezifisch
- Varianten
 - Navision Varianten
 - Navision Varianten mit Template
- Pfandartikel inklusive Pfandhandling in Oxid
- Kundenspezifische Sichtbarkeit von Artikeln in OXID importieren aus Navision

3.1.2 Preise

- Verkaufskanalspezifisch
- Preise auf Kundenebene
- Preise für Kundengruppen

3.1.3 Kunden

- Kundenanlage aus Navision in OXID
- Kundenanlage OXID an Navision

3.1.4 Bestellungen

- Export Bestellungen OXID an Navision
- Import OXID Status Bestellung aus Navision inklusive:
 - Paketnummer(n) je Auftrag
 - Rechnung(en)

3.1.5 Kategorien

- Kategorieimport aus Navision
- Mandantenspezifische Kategorien

3.1.6 Bestände

- Artikelbestände Import aus Navision

3.1.7 Datenaustausch Möglichkeiten

- Zeitgesteuerter Vollimport der Daten aus Navision in OXID
- Import in OXID wird in Echtzeit gestartet bei Datenänderung in Navision

3.1.8 Masterdata

- Import und Handling der Navision Masterdata

3.2 Erweiterbarkeit

Durch eine intelligente und generische Schnittstellenkonstruktion ist es mittels verschiedener definierter Hooks möglich, kundenspezifische Navision Anforderungen und Anpassungen jederzeit in der Schnittstelle zu implementieren.

3.3 Schaubild



4 Installation

4.1 Rahmenbedingen

Die Daten werden per **SOAP Protokoll** ausgetauscht. Auf Navisionseite wird ein **TSO Connector**¹ benötigt.

Die Schnittstelle aktualisiert automatisch ihre SOAP Klassen aus Navision und ist somit immer aktuell.

Diese müssen allerdings **einmalig manuell** generiert werden.

Bitte stellen Sie sicher, dass die **SOAP** und **CURL** Erweiterungen installiert und in der **php.ini** des Servers, auf dem Ihr Shop läuft, eingetragen sind.

4.2 Vorgehensweise

1. Beginnen Sie die Installation mit dem Kopieren der Daten des Moduls in das Verzeichnis **modules/** des OXID-eShops.
2. Navigieren Sie im Admin-Bereich ihres OXID-eShops zu dem Menüpunkt **Erweiterungen** → **Module**.
3. Dort klicken sie bitte auf das Modul **smxDynNAV2OXID** und klicken unten auf den Button **Aktivieren**.
4. Sie haben das Modul erfolgreich installiert!

1 http://www.tso.de/fileadmin/dateien/0-downloads/Kompetenzen/Flyer_Web-Connector_2014.pdf

5 Konfiguration

5.1 Generierung der SOAP Klassen

Sie müssen die zur Kommunikation notwendigen Klassen noch **manuell** erzeugen. Die dafür vorgesehenen Scripts finden sie im Modulverzeichnis unter dem Ordner **/tools**.

Bitte erzeugen Sie die Klassen auf ihrem **Server**, indem sie das Skript **generate_soap_classes_server.php** auf ihrem Server ausführen.

```
php -f generate_soap_classes_server.php
```

5.2 Konfigurationsparameter

Um das Modul anzupassen öffnen sie bitte die **config.inc.php** Ihres OXID-eShops und fügen folgende Parameter hinzu (siehe auch Beispiele darunter):

sDynNAV2OXIDBaseUrl

String

URL-Pfad von Dynamics NAVISION

Beispiel:

```
$this->sDynNAV2OXIDBaseUrl = 'http://exampleqmbh.com:444/example/';
```

sDynNAV2OXIDUseAuth

Boolean

Benutze Authentifizierung: Ja / Nein. True: Parameter Domain und User erforderlich. False: Keine Authentifizierung

Beispiel:

```
$this->sDynNAV2OXIDUseAuth = true;
```

sDynNAV2OXIDDomain

String

Angabe der Domain

smxDynNAV2OXID Modul

Beispiel:

```
$this->sDynNAV2OXIDDomain = 'DOMAIN';
```

sDynNAV2OXIDUser

String

Benutzername (Dynamics NAVISION)

Beispiel:

```
$this->sDynNAV2OXIDUser = 'smx-oxid';
```

sDynNAV2OXIDPassword

String

Passwort (Dynamics NAVISION)

Beispiel:

```
$this->sDynNAV2OXIDPassword = 'secret01';
```

aDynNAV2OXIDSalutationMapping

Array

Bildet die Anrede aus Dynamics NAVISION auf OXID-eShop ab

Beispiel:

```
$this->aDynNAV2OXIDSalutationMapping = array(  
    'MR' => 'HERR',  
    'MRS' => 'FRAU',  
);
```

aDynNAV2OXIDMasterShopId

Integer

Die ID des Mastershops (OXID)

Beispiel:

```
$this->iDynNAV2OXIDMasterShopId = 1;
```

aDynNAV2OXIDVatMapping

Array

Bildet die Mehrwertsteuercodes aus Dynamics NAVISION in OXID-eShop ab

smxDynNAV2OXID Modul

Beispiel:

```
$this->aDynNAV2OXIDVatMapping = array(  
    'VAT19' => 19,  
    '19' => 19,  
    '7' => 7,  
);
```

aDynNAV2OXIDShopMapping

Array

Bildet die Verkaufskanäle aus Dynamics NAVISION in OXID-eShop ab

Beispiel:

```
$this->aDynNAV2OXIDShopMapping = array(  
    'Mastershop' => 1,  
    'Subshop' => 2,  
);
```

blDynNAV2OXIDRemoveOldArticles

Boolean

Bestimmt, ob alte Artikel bei Import gelöscht werden

Beispiel:

```
$this->blDynNAV2OXIDRemoveOldArticles = true;
```

sDynNAV2OXIDPicturePath

String

Angabe des Verzeichnisses für Bilder, welche importiert werden sollen

Beispiel:

```
$this->sDynNAV2OXIDPicturePath = $this->sShopDir . '/import';
```

sDynNAV2OXIDImportLockFile

String

Dateipfad der Sperr-Datei für Imports²

Beispiel:

```
$this->sDynNAV2OXIDImportLockFile =  
$this->sShopDir . '/import/import_lock';
```

2

Die Sperr-Datei signalisiert das gerade ein Prozess stattfindet und daher keine anderen Import/Exports gleichzeitig gestartet werden

smxDynNAV2OXID Modul

sDynNAV2OXIDExportLockFile

String

Dateipfad der Sperr-Datei für Exports³

Beispiel:

```
$this->sDynNAV2OXIDExportLockFile =  
$this->sShopDir . '/import/export_lock';
```

aDynNAV2OXIDLanguageMapping

Array

Abbildung der Sprachen aus Dynamics NAVISION auf OXID-eShop

Beispiel:

```
$this->aDynNAV2OXIDLanguageMapping = array(  
    "DEU" => 0,  
    "ENU" => 1,  
    "ENG" => 1,  
);
```

aDynNAV2OXIDPayMapping

Array

Abbildung der Zahlungsarten aus Dynamics NAVISION auf OXID-eShop

Beispiel:

```
$this->aDynNAV2OXIDPayMapping = array(  
    "oxidcreditcard" => "KREDITKART",  
    "oxiddebitnote" => "LASTSCHR",  
);
```

sDynNAV2OXIDResponsibility

String

Bestimmt den Verantwortungsbereich aus Dynamics NAVISION

Beispiel:

```
$this->sDynNAV2OXIDResponsibility = 'SHOPTIMAX';
```

sDynNAV2OXIDResourceNoVoucher

String

3

Die Sperr-Datei signalisiert das gerade ein Prozess stattfindet und daher keine anderen Import/Exports gleichzeitig gestartet werden

smxDynNAV2OXID Modul

Ressourcennummer (Gutscheine) aus Dynamics NAVISION

Beispiel:

```
$this->sDynNAV2OXIDResponsibility = '1234';
```

sDynNAV2OXIDResourceNoDeliveryCost

String

Ressourcennummer (Lieferkosten) aus Dynamics NAVISION

Beispiel:

```
$this->sDynNAV2OXIDResponsibility = '5678';
```

sDynNAV2OXIDPaymentCost

Array

Abbildung der Zahlungskodes aus Dynamics NAVISION auf OXID-eShop

Beispiel:

```
$this->aDynNAV2OXIDPaymentCost = array(  
    7 => '863200',  
    19 => '864200',  
);
```

aDynNAV2OXIDVPECodeMapping

Array

Abbildung der Verpackungseinheitencodes von Dynamics NAVISION auf OXID-eShop, die Nummern bestimmen die Spracheinträge

Beispiel:

```
$this->aDynNAV2OXIDVPECodeMapping = array(  
    'STÜCK' => array(  
        0 => 'Stück',  
        1 => 'Piece',  
    ),  
);
```

sDynNAV2OXIDNotRegisteredContactNo

String

Kontaktnummer für nicht registrierte Kontakte

Beispiel:

```
$this->sDynNAV2OXIDNotRegisteredContactNo = '1011111';
```

aDynNAV2OXIDShippingAgentCodeMapping*Array*

Abbildung der Lieferanten (bzw ID) aus Dynamics NAVISION auf OXID-eShop
Abhängig vom Mandanten (im Beispiel: 2)

Beispiel:

```
$this->aDynNAV2OXIDShippingAgentCodeMapping = array(  
    2 => array('4b6b3f' => 'DHL',  
              '36a4b' => 'DPD'  
            ),  
    4 => array('6b4bb' => 'UPS',  
              '46412' => 'DPD',  
            ),  
);
```

sDynNAV2OXIDShippingAgentServiceCode*String*

Dienststart des Lieferanten

Beispiel:

```
$this->sDynNAV2OXIDShippingAgentServiceCode = 'EXPRESS';
```

aDynNAV2OXIDManufacturerMapping*Array*

Abbildung der Hersteller aus Dynamics NAVISION auf OXID-eShop

Beispiel:

```
$this->aDynNAV2OXIDManufacturerMapping = array(  
    'SIEMENS' => 'SIEMENS_1',  
    'GOOGLE' => 'YOUTUBE',  
);
```

6 Betrieb

6.1 Starten der Schnittstellen

Für den Start des Imports bzw. Exports der Schnittstelle, sind die Shell-Skripte im Ordner **/scripts** des Moduls zuständig.

export_orders.sh	startet den Export der Bestellungen aus OXID nach Navision
import_full_contacts.sh	startet den Import aller Kontaktdaten aus Navision nach OXID
import_full.sh	startet den Import der Artikel, Kategorien, Attribute und Bestände aus Navision nach OXID
import_changed.sh	startet den Import aller Artikel, Kategorien, Attribute, Bestände und Kontakte die sich geändert haben

Sie können diese (testweise) entweder **manuell** ausführen oder als **Cronjob** zum ständigen Betrieb einrichten. Die Ausführung als **Cronjob** wird dringend empfohlen.

6.2 Log-Dateien und Error-Handling

Die verarbeiteten Dateien werden in einer Logdatei im Ordner **/logs** unter dem entsprechenden Namen **dynnavoxid_[xxxx].log** mitgeschrieben.

Sollte eine Datei nicht verarbeitet werden können, wird ebenfalls ein Eintrag in der Logdatei generiert.