

shoptimax GmbH | Ulmenstraße 52H | 90443 Nürnberg

Manual smxDynNAV2OXID Modul

Stand: 05.03.2015

Version 0.9

Autor des Dokuments	Matthias Zistler	Erstellt am	05.03.2015
		© shoptimax GmbH	Vertraulich!

1 Index

2 History of report.....	3
3 Description.....	4
3.1 Functional scope.....	4
3.1.1 Articles.....	4
3.1.2 Price.....	5
3.1.3 Customers.....	5
3.1.4 Orders.....	5
3.1.5 Categories.....	5
3.1.6 Stocks.....	5
3.1.7 Data exchange possibilities.....	5
3.1.8 Master data.....	5
3.2 Extensibility.....	6
3.3 Graph.....	6
4 Installation.....	7
4.1 Requirements.....	7
4.2 Instructions.....	7
5 Configuration.....	8
5.1 Configuration parameters.....	8
5.2 Generating the SOAP-Classes.....	13
6 Ongoing operation.....	14
6.1 Start of the interface.....	14
6.2 Log-Files and Error-Handling.....	14

2 History of report

Version	Date	Author	Comments
0.9	05.03.2015	Matthias Zistler	Creation

3 Description

The module **smxDynNAV2OXID** allows the exchange of data between Microsoft Dynamics NAVISION and OXID-eShop.

It enables you to import complete data records out of Dynamics NAVISION into OXID-eShop and export them in the opposite direction. Therefore the data out of Dynamics NAVISION is imported into a **mastershop** of OXID-eShop, and all its **subshops** inherit the data. The **subshops** correspond to the **distribution channels** of Dynamics NAVISION. The export equally maps the **distribution channels** to the **subshops**.

Dynamics NAVISION is superior in terms of **articles**, **categories**, and **stock**, while OXID-eShop is superior in terms of **orders**. **Customer** data can be synced in both directions.

3.1 Functional scope

3.1.1 Articles

- Master data
- Article pictures
- Artikel pictures client specific
- Attributes
- Packing units
 - option to sell packing units
- Base price
- Various languages are possible
- Article title
- Article title client specific
- Variants
 - Navision Variants
 - Navision Variants with template
- Deposit articles incl. Deposit handling in OXID
- Import customer specific scope of articles into OXID

3.1.2 Price

- Distribution channel specific
- Prices for customer groups
- Prices on customer level

3.1.3 Customers

- Customer system from Navision to OXID
- Customer system from OXID to Navision

3.1.4 Orders

- Export orders from OXID to Navision
- Import OXID status orders from Navision including:
 - packet number per order
 - invoices

3.1.5 Categories

- Import categories from Navision
- Client specific categories

3.1.6 Stocks

- Article stock import from Navision

3.1.7 Data exchange possibilities

- Event driven fully import of the data from Navision to OXID
- Imports into OXID will be triggered in real-time if data has changed in Navision

3.1.8 Master data

- Import and handling of the Navision masterdata

3.2 Extensibility

The intelligent and generic construction of the interface, makes it possible to implement customer specific requirements and customization from Navision into the interface, through several defined hooks.

3.3 Graph



4 Installation

4.1 Requirements

The data is transferred via the **SOAP protocol**. On Navision side the **TSO connector** is mandatory.

The interface updates your **SOAP** classes from Navision, so the interface is always up-to-date. These classes have to be generated once **manually**.

Please make sure the **SOAP** and **CURL** extensions are installed and registered in the **php.ini** file on the same server on which the shop is running.

4.2 Instructions

1. Start the installation by copying all files and folders of the module to the directory **/modules** of the OXID-eShop
2. Log in to the admin backend of your OXID-eShop and navigate to the menu item **Extensions → Modules**
3. Please select the module **smxDynNAV2OXID** and press the button **Activate** below
4. You have successfully installed the module!

5 Configuration

5.1 Configuration parameters

Please open the **config.inc.php** of your OXID-eShop and add the following configuration parameters, to configure the interface (see also the examples below):

sDynNAV2OXIDBaseUrl

String

URL path of Dynamics NAVISION

Example:

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

sDynNAV2OXIDUseAuth

Boolean

Use Authentication: Yes/No. True: Parameters Domain and User are required. False: No Authentication.

Example:

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

sDynNAV2OXIDDomain

String

Dynamics NAVISION's domain.

Example:

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

sDynNAV2OXIDUser

String

Username (Dynamics NAVISION)

Example:

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


smxDynNAV2OXID

sDynNAV2OXIDPassword

String

Password (Dynamics NAVISION)

Example:

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

aDynNAV2OXIDSalutationMapping

Array

Maps the Salutation of Dynamics NAVISION to that of the OXID eShop

Example:

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

aDynNAV2OXIDMasterShopId

Integer

ID of the Mastershop (OXID)

Example:

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

aDynNAV2OXIDVatMapping

Array

Bildet die Mehrwertsteuercodes aus Dynamics NAVISION in OXID-eShop ab

Example:

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

smxDynNAV2OXID

aDynNAV2OXIDShopMapping

Array

Maps the distribution channel name from Dynamics NAVISION to the corresponding subshops of your OXID-eShop

Example:

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

bDynNAV2OXIDRemoveOldArticles

Boolean

If false, no old Articles will be removed if the Article is re-imported

Example:

```
$this->bDynNAV2OXIDRemoveOldArticles = true;
```

sDynNAV2OXIDPicturePath

String

Specifies the path to the directory of pictures, which shall be imported

Example:

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

sDynNAV2OXIDImportLockFile

String

Path of Lock-File for imports, which signalsizes that the process in running and no other imports can be started

Example:

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

sDynNAV2OXIDExportLockFile

String

Path of Lock-File for exports, which signalsizes that the process in running and no other exports can be started

smxDynNAV2OXID

Example:

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

aDynNAV2OXIDLanguageMapping

Array

Maps the multi-tenancy of Dynamics NAVISION to that of the OXID eShop

Example:

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

aDynNAV2OXIDPayMapping

Array

Maps the paying methods of Dynamics NAVISION to those in OXID-eShop

Example:

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

sDynNAV2OXIDResponsibility

String

Determines the responsibility in Dynamics NAVISION

Example:

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

sDynNAV2OXIDResourceNoVoucher

String

Resource number (voucher) out of Dynamics NAVISION

Example:

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

smxDynNAV2OXID

sDynNAV2OXIDResourceNoDeliveryCost

String

Resource number (delivery cost) out of Dynamics NAVISION

Example:

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

sDynNAV2OXIDPaymentCost

Array

Maps the payment codes of Dynamics NAVISION to those in OXID-eShop

Example:

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

aDynNAV2OXIDVPECodeMapping

Array

Maps the package codes of Dynamics NAVISION to those in OXID-eShop;
The numbers determine the language entries

Example:

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

sDynNAV2OXIDNotRegisteredContactNo

String

Contact number for not registred contacts

Example:

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

sDynNAV2OXIDShippingAgentServiceCode

String

Shipping agent's service (from Dynamics NAVISION)

smxDynNAV2OXID

Beispiel:

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

aDynNAV2OXIDShippingAgentCodeMapping

Array

Maps the shipping agent codes of Dynamics NAVISION to those in OXID-eShop
The Keys specify the subshops. In the example below 2, 4 are the subshops.

Beispiel:

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

aDynNAV2OXIDManufacturerMapping

Array

Maps the manufacturers of Dynamics NAVISION to those of Dynamics NAVISION

Beispiel:

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

5.2 Generating the SOAP-Classes

Before data can be exchanged you have to generate the classes **manually**.

The provided scripts for that purpose are located in the directory of the module inside the folder **/tool**.

Please generate the classes on your **server**. To do so, execute the script **generate_soap_classes_server.php** on your server.

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

6 Ongoing operation

6.1 Start of the interface

The scripts provided in the directory **/scripts** are responsible to start the imports/exports.

export_orders.sh	starts the export of orders out of the OXID-eShop to Navision
import_full_contacts.sh	starts the import of all customer data from Navision to OXID-eShop
import_full.sh	starts the import of articles, categories, attributes and stocks from Navision to OXID
import_changed.sh	starts the import of articles, categories, attributes, stocks and contacts which have been created or modified

You can execute them **manually** (for the purpose of testing, e.g) or define a **cronjob** for that. The automatic execution via **cronjob** is highly recommended.

6.2 Log-Files and Error-Handling

The processed data will be logged in the file **dynnav2oxid_[xxxx].log** which is located inside the folder **/log**.

Also if data can **not** be processed an error entry will be logged inside this file.