

**OXID eSales
documentation
ERP interface (CSV)
protocol version 2.3.0**

Copyright

Copyright © 2008 OXID eSales AG, Germany

All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the prior written permission of OXID eSales AG.

Decompilation of source code, piracy as well as transfer to a third party is not allowed.

Use of this content for any other purpose is expressly prohibited and may result in severe civil and criminal penalties. Violators will be prosecuted to the maximum extent possible.

The information contained within this document was created accordingly to the state of the art technology.

OXID eSales AG assumes no liability or warranty for the accuracy, the completeness or reliability of any content.

In spite of all efforts, the occurrence of mistakes cannot be eliminated completely.

Thus we appreciate any tips.

Conventions

The following typographical conventions are used in this document:

cursive font

used for file names, file paths, email addresses and URLs.

`non-proportional font`

used for code examples and names of code elements.

Non-proportional cursive font

used for navigation steps.

Contact

OXID eSales AG

Bertoldstrasse 48

79098 Freiburg

Germany

Fon: +49 (761) 36889 0

Fax: +49 (761) 36889 29

Represented by the Supervisory Board: Roland Fesenmayr (CEO), Eric Jankowsky, Andrea Seeger

Registered Office Freiburg i. Br.

Commercial Register Freiburg

No. HRB 701648

Table of Contents

Copyright	2
Conventions	2
Contact	2
Table of Contents	3
About this Document	5
About the interface	5
1 Installation	5
2 File format conventions for import	5
2.1 Data field specifications	5
2.2 Replacement table	6
2.3 Import file requirements	6
3 Importing	6
3.1 By Cronjob	6
4 Import interface	6
4.1 Version line	6
4.2 Create/update products	6
4.3 Update extended properties of products	13
4.4 Delete products	13
4.5 Create/Update categories	14
4.6 Delete categories	16
4.7 Create/Update vendors	16
4.8 Delete Vendor	17
4.9 Create/update crossselling assignment	18
4.10 Delete Crossselling assignment	18
4.11 Create/Update Accessory assignment	18
4.12 Delete Accessory Assignment	18
4.13 Create/Update Category assignments	19
4.14 Set main category	19
4.15 Delete Category assignment	19
4.16 Create/update action assignment	20
4.17 Delete action assignment	20
4.18 Create/update scale prices	20
4.19 Delete scale prices	21
4.20 Create/update orders	21
4.21 Delete orders	25
4.22 Create/update product assignments to orders	25
4.23 Delete product assignments to orders	28
4.24 Change order status	28
4.25 Create/update product assignments to vendors	28
4.26 Delete product assignments to vendors	29
4.27 Updating stock	29

4.28	Create/update users	29
4.29	Delete users	31
4.30	Create/Update countries.....	31
4.31	Delete Countries	32
5	Export interface	32
5.1	Sorting export data	32
5.2	Login	32
5.3	OXERPGetCategory	33
5.4	OXERPGetCategories.....	33
5.5	OXERPGetArticle.....	33
5.6	OXERPGetArticles	34
5.7	OXERPGetArticleCategory	34
5.8	OXERPGetArticleAction	35
5.9	OXERPGetArticleCrossselling	35
5.10	OXERPGetArticleScalePrice	35
5.11	OXERPGetArticleAccessoire	36
5.12	OXERPGetArticleStock	36
5.13	OXERPGetOrder	36
5.14	OXERPGetOrderArticle.....	37
5.15	OXERPGetVendor	37
5.16	OXERPGetArticleVendor.....	37
5.17	OXERPGetVendorArticles	38
5.18	OXERPGetUser	38
5.19	OXERPGetUsers.....	38
5.20	OXERPGetOrders	39
5.21	OXERPGetVendors	39
5.22	OXERPCallPlugin.....	40
6	Using custom fields	41
6.1	Example	41
7	Extending ERP with plugins	42
7.1	Creating a plugin.....	42
7.2	Sample plugin.....	42
8	Expandability of CSV interface	44
8.1	Define new attribute object.....	44
8.2	Implementing function in oxerpgateway.php.....	44
8.3	Import definition in oxerpcsv.php	45
8.4	Using new OXERPSetAttribute function	45
8.5	Registering function in oxerpcsvexport.php.....	46

About this Document

This document describes the protocol version 2.3.0 of the ERP-CSV interface. This interface can be used since eShop version 4.0.1.0.

Older protocol versions can still be used. Valid protocol versions are:

- 1
- 1.1
- 2
- 2.1.0
- 2.2.0
- 2.3.0

About the interface

The CSV interface is used for importing and exporting eShop data (e.g. products, categories, users etc.).

- The import is made from files which were uploaded to the server.
- The export is done by calling a particular URL.

In `testdata_examples` directory sample import files for each protocol version are given.

1 Installation

- Copy all files from *copy_this* to the main directory of your eShop
- Open eShop admin
- Go to *Master Settings* -> *Core Settings* -> *System* -> *Modules*
- In *Installed Shop Modules*, add the line `oxerpinterface => erp/oxerpcsv`

2 File format conventions for import

- Text files with Windows line breaks
- character encoding: ANSI

2.1 Data field specifications

- All data fields are enclosed in double quotes and separated by a semicolon.
- For numbers, a point is used as decimal separator.
- Leading zeros are omitted.
- For fields filled with 0 or 1, 0 always means no, 1 means yes.

- Named data types are taken from MySQL database.

2.2 Replacement table

In Import files, the following chars have to be replaced:

char	Description	ASCII-Code	Replacement
CR	Carriage Return	13	
LF	Line Feed	10	

"	Double Quote	34	"
'	Single Quote	39	'

2.3 Import file requirements

The first line has to contain version information. Following data can be in any order.

3 Importing

3.1 By Cronjob

If import is to be run by a cronjob, adjust `modules/ERP/oxerpccron.php` according to your needs. Change file permissions to executable.

4 Import interface

At the beginning of each dataset, a char which indicates the data type is set. This char will be called „Indicator char“ in the following text.

4.1 Version line

The first line always has to contain information about the used protocol version. Example:

"V", "2.3.0"

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(1)	[V]	
1.	Version number	char(255) NOT NULL default ""	[0-9.]	

4.2 Create/update products

If a product with the given OXID already exists, the product is updated with the given values. Otherwise, a new product is created.

#	Content/Example	Data type	Range	Field name ¹
---	-----------------	-----------	-------	-------------------------

¹ Table: oxarticles

#	Content/Example	Data type	Range	Field name ¹
0.	Indicator char	char(1)	[A]	
1.	Product ID	char(32) NOT NULL	[0-9a-f.]	OXID
2.	Shop ID	char(32) NOT NULL default ""	1 - 64	OXSHOPID ²
3.	Product used in Shops	bigint(20) NOT NULL	binary encoded, 64 bit. <u>Example:</u> Product is in Shops #2 and #7. => 2nd 7th bit have to be set: The result is 66	OXSHOPINCL ²
4.	Product not used in Shops	bigint(20) NOT NULL	binary encoded, 64 bit. See above.	OXSHOPEXCL ³
5.	Parent-ID For variants: OXID of parent product	char(32) default NULL	[0-9a-f.]	OXPARENTID
6.	Active	tinyint(1) NOT NULL default '1'	[0,1]	OXACTIVE
7.	Active from	datetime default 0000-00-00 00:00:00	YYYY-MM-DD HH:MM:SS	OXACTIVEFROM
8.	Active to	datetime default 0000-00-00 00:00:00	YYYY-MM-DD HH:MM:SS	OXACTIVETO
9.	Product number	varchar(32) NOT NULL default ""	[SPACE-~] ⁴	OXARTNUM
10.	EAN	varchar(13) NOT NULL default ""	[0-9]	OXEAN
11.	EAN of manufacturer	varchar(13) NOT NULL default ""	[0-9]	OXDISTEAN
12.	Title	varchar(255) NOT NULL default ""	[SPACE-~]	OXTITLE
13.	Short description	varchar(255) NOT NULL default ""	[SPACE-~]	OXSHORTDESC

² Always 1 in OXID eShop Professional Edition

³ Always 0 in OXID eShop Professional Edition

⁴ ASCII chr(32) – chr(126)

#	Content/Example	Data type	Range	Field name ¹
14.	Price	double NOT NULL default '0'	- 1.7976931348623157 E+308 to - 2.2250738585072014 E-308, 0, and 2.2250738585072014 E-308 to 1.7976931348623157 E+308 ⁵	OXPRICE
15.	Product has fixed price	tinyint(1) NOT NULL default '0'	[0,1]	OXBLFIXEDPRICE
16.	Price A	double NOT NULL default '0'	- 1.7976931348623157 E+308 to - 2.2250738585072014 E-308, 0, and 2.2250738585072014 E-308 to 1.7976931348623157 E+308	OXPRICEA
17.	Price B	double NOT NULL default '0'	see above	OXPRICEB
18.	Price C	double NOT NULL default '0'	see above	OXPRICEC
19.	Purchase price	double NOT NULL default '0'	see above	OXBPRICE
20.	RRP	double NOT NULL default '0'	see above	OXTPRICE
21.	Unit	varchar(32) NOT NULL default ''	[SPACE-~]	OXUNITNAME
22.	Quantity	double NOT NULL default '0'	see field 16	OXUNITQUANTITY
23.	External URL	varchar(255) NOT NULL default ''	valid URL without <i>http://</i>	OXEXTURL
24.	Short description for external URL	varchar(255) NOT NULL default ''	[SPACE-~]	OXURLDESC
25.	URL of picture for external URL	varchar(128) NOT NULL default ''	see field 23	OXURLIMG

⁵ Ranges defined by IEEE. The range that can be used is determined by hardware and operating system.

#	Content/Example	Data type	Range	Field name ¹
26.	Special VAT for this product	double default NULL	see field 16 Default: 0	OXVAT
27.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~] ⁶	OXTHUMB
28.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	OXICON
29.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC1
30.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC2
31.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC3
32.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC4
33.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC5
34.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC6
35.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC7
36.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC8
37.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC9
38.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC10
39.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC11
40.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC12
41.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XZOOM1
42.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XZOOM2
43.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XZOOM3
44.	Filename (without path), Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XZOOM4
45.	Weight	double NOT NULL default '0'	see field 16	OXWEIGHT

⁶ Dependant on the file system

#	Content/Example	Data type	Range	Field name ¹
46.	Stock	int(11) NOT NULL default '-1' ⁷	0 – 2147483647	OXSTOCK
47.	Delivery Status	enum('1','2','3','4') NOT NULL default '1'	1. Standard 2. If out of Stock, offline 3. If out of Stock, not orderable 4. External Storehouse	OXSTOCKFLAG
48.	In Stock Message	varchar(255) NOT NULL default ''	[SPACE-~]	OXSTOCKTEXT
49.	Out of Stock Message	varchar(255) NOT NULL default ''	[SPACE-~]	OXNOSTOCKTEXT
50.	Available on	date NOT NULL default '0000-00-00'	YYYY-MM-DD	OXDELIVERY
51.	Date of creation (is ignored, actual date taken instead)	date NOT NULL default '0000-00-00'	YYYY-MM-DD	OXINSERT
52.	Date of last change (is ignored, actual date taken instead)	timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP	YYYY-MM-DD HH:MM:SS	OXTIMESTAMP
53.	Length	double NOT NULL default '0'	see field 16	OXLENGTH
54.	Width	double NOT NULL default '0'	see field 16	OXWIDTH
55.	Height	double NOT NULL default '0'	see field 16	OXHEIGHT
56.	File	varchar(128) NOT NULL default ''	[SPACE-~]	OXFILE
57.	Search Terms	varchar(255) NOT NULL default ''	[SPACE-~]	OXSEARCHKEYS
58.	Alt. Template	varchar(128) NOT NULL default ''	[SPACE-~]	OXTEMPLATE
59.	Alt. Contact	varchar(255) NOT NULL default ''	valid e-Mail address	OXQUESTIONEMAIL
60.	Searchable	tinyint(1) NOT NULL default '1'	[0,1]	OXISSEARCH

⁷ -1 was a special value in older versions. Negative values are not to be used in Enterprise Edition any more.

#	Content/Example	Data type	Range	Field name ¹
61.	For variants: Name of variant selection (e.g. Color)	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARNAME
62.	Stock of all variants	int(5) NOT NULL default '0'	0 – 2147483647	OXVARSTOCK
63.	Amount of variants	int(1) NOT NULL default '0'	0 – 2147483647	OXVARCOUNT
64.	Name of variant (e.g. green)	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARSELECT
65.	lowest price of all variants	double NOT NULL default '0'	see field 16	OXVARMINPRICE
66.	Name of variant selection in language 1	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARNAME_1
67.	Name of variant in language 1	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARSELECT_1
68.	Name of variant selection in language 2	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARNAME_2
69.	Name of variant in language 2	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARSELECT_2
70.	Name of variant selection in language 3	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARNAME_3
71.	Name of variant in language 2	varchar(32) NOT NULL default ''	[SPACE-~]	OXVARSELECT_3
72.	Title in language 1	varchar(255) NOT NULL default ''	[SPACE-~]	OXTITLE_1
73.	Short description in language 1	varchar(255) NOT NULL default ''	[SPACE-~]	OXSHORTDESC_1
74.	URL description in language 1	varchar(255) NOT NULL default ''	see field 23	OXURLDESC_1
75.	Search terms in language 1	varchar(255) NOT NULL default ''	[SPACE-~]	OXSEARCHKEYS_1
76.	Title in language 2	varchar(255) NOT NULL default ''	[SPACE-~]	OXTITLE_2
77.	Short description in language 2	varchar(255) NOT NULL default ''	[SPACE-~]	OXSHORTDESC_2
78.	URL description in language 2	varchar(255) NOT NULL default ''	see field 23	OXURLDESC_2
79.	Search terms in language 2	varchar(255) NOT NULL default ''	[SPACE-~]	OXSEARCHKEYS_2
80.	Title in language 3	varchar(255) NOT NULL default ''	[SPACE-~]	OXTITLE_3
81.	Short description in language 3	varchar(255) NOT NULL default ''	[SPACE-~]	OXSHORTDESC_3

#	Content/Example	Data type	Range	Field name ¹
82.	URL description in language 3	varchar(255) NOT NULL default ''	see field 23	OXURLDESC_3
83.	Search terms in language 3	varchar(255) NOT NULL default ''	[SPACE-~]	OXSEARCHKEYS_3
84.	(always empty)	varchar(32) NOT NULL default ''	always empty	OXFOLDER
85.	class	varchar(32) NOT NULL default ''	always <i>oxarticle</i>	OXSUBCLASS
86.	In Stock Message in language 1	varchar(255) NOT NULL default ''	[SPACE-~]	OXSTOCKTEXT_1
87.	In Stock Message in language 2	varchar(255) NOT NULL default ''	[SPACE-~]	OXSTOCKTEXT_2
88.	In Stock Message in language 3	varchar(255) NOT NULL default ''	[SPACE-~]	OXSTOCKTEXT_3
89.	Out of Stock Message in language 1	varchar(255) NOT NULL default ''	[SPACE-~]	OXNOSTOCKTEXT_1
90.	Out of Stock Message in language 2	varchar(255) NOT NULL default ''	[SPACE-~]	OXNOSTOCKTEXT_2
91.	Out of Stock Message in language 3	varchar(255) NOT NULL default ''	[SPACE-~]	OXNOSTOCKTEXT_3
92.	Sorting	int(5) NOT NULL default '0'	0 – 2147483647	OXSORT
93.	Amount of sold units	double NOT NULL default '0'	see field 16	OXSOLDAMOUNT
94.	Intangible Product	int(1) NOT NULL default '0'	[0,1]	OXNONMATERIAL
95.	Free Shipping	int(1) NOT NULL default '0'	[0,1]	OXFREESHIPPING
96.	Send e-mail if Stock falls below Value	int(1) NOT NULL default '0'	[0,1]	OXREMINDACTIV
97.	Value for low Stock e-mail	double NOT NULL default '0'	see field 16	OXREMINDAMOUNT
98.	(always empty)	varchar(32) NOT NULL default ''	always empty	OXAMITEMID
99.	(always empty)	varchar(16) NOT NULL default '0'	always empty	OXAMTASKID
100.	Vendor ID	varchar(32) NOT NULL default ''	[0-9a-f.]	OXVENDORID
101.	Skip all negative Discounts	tinyint(1) NOT NULL default '0'	[0,1]	OXSKIPDISCOUNTS
102.	Additional info in order confirmation	varchar(255) NOT NULL	[SPACE-~]	OXORDERINFO ²¹
103.	Export to pixi*	tinyint(4) NOT NULL default '0'	[0,1]	OXPIXIEXPORT ³

#	Content/Example	Data type	Range	Field name ¹
104.	Export to pixi* timestamp	timestamp NOT NULL default '0000-00-00 00:00:00'	YYYY-MM-DD HH:MM:SS	OXPIXIEXPORTED ⁸
105.	Packaging unit	int(11) NOT NULL default '1'	0 – 2147483647	OXVPE ²
106.	Long description	text	[SPACE-~]	OXLONGDESC
107.	Average rating	Double NOT NULL default '0'	see field 16	OXRATING
108.	Amount of ratings	Int (11) NOT NULL default '0'	0 – 2147483647	OXRATINGCNT
109.	Long description in language 1	text NOT NULL	[SPACE-~]	OXLONGDESC_1
110.	Long description in language 2	text NOT NULL	[SPACE-~]	OXLONGDESC_2
111.	Long description in language 3	text NOT NULL	[SPACE-~]	OXLONGDESC_3

4.3 Update extended properties of products

With this function long descriptions and tags can be imported for existing products.

#	Content/Example	Data type	Range	Field name ⁹
0.	Indicator char	char(1)	[Y]	
1.	Product ID	char(32) NOT NULL	[0-9a-f.]	OXID
2.	long description	text	[SPACE-~]	OXLONGDESC
3.	Long description in language 1	text NOT NULL	[SPACE-~]	OXLONGDESC_1
4.	Long description in language 2	text NOT NULL	[SPACE-~]	OXLONGDESC_2
5.	Long description in language 3	text NOT NULL	[SPACE-~]	OXLONGDESC_3
6.	Tags	varchar(255) NOT NULL	[SPACE-~]	OXTAGS
7.	Tags in language 1	varchar(255) NOT NULL	[SPACE-~]	OXTAGS_1
8.	Tags in language 2	varchar(255) NOT NULL	[SPACE-~]	OXTAGS_2
9.	Tags in language 3	varchar(255) NOT NULL	[SPACE-~]	OXTAGS_3

4.4 Delete products

When deleting a product, all assignments for this product (crossselling, categories etc.) are deleted from database aswell.

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XA]	

⁸ In OXID eShop Professional Edition always 0000-00-00 00:00:00

⁹ Table: oxartextends

1.	Product number	char(32) NOT NULL default ''	[0-9a-f.]	OXID
----	----------------	------------------------------	-----------	------

4.5 Create/Update categories

If a category with the given OXID already exists, the category is updated with the given values.
Otherwise, a new category is created.

#	Content/Example	Data type	Range	Field name ¹⁰
0.	Indicator char	char(1)	[K]	
1.	Category ID	varchar(32) NOT NULL default ''	[0-9a-f.]	OXID
2.	Parent category ID	varchar(32) NOT NULL default ''	[0-9a-f.]	OXPARENTID
3.	Nested category tree left	int(11) NOT NULL default '0'	0 to 2147483647	OXLEFT
4.	Nested category tree right	int(11) NOT NULL default '0'	0 to 2147483647	OXRIGHT
5.	Nested Category tree root	varchar(32) NOT NULL default ''	[0-9a-f.]	OXROOTID
6.	Sorting	int(11) NOT NULL default '9999'	0 to 2147483647	OXSORT
7.	Active	tinyint(1) NOT NULL default '1'	[0,1]	OXACTIVE
8.	Hidden	tinyint(1) NOT NULL default '0'	[0,1]	OXHIDDEN
9.	Shop-ID of Shop in which Category was created	varchar(255) NOT NULL default ''	1 to 64	OXSHOPID ²
10.	Category used in Shops	bigint(20) unsigned NOT NULL default '0'	binary encoded, 64 bit. <u>Example:</u> Category is in Shops #2 and #7. => 2nd 7th bit have to be set: The result is 66	OXSHOPINCL ²
11.	Category not used in Shops	bigint(20) unsigned NOT NULL default '0'	see field 10	OXSHOPEXCL ³
12.	Title	varchar(254) NOT NULL default ''	[SPACE-~]	OXTITLE
13.	Short description	varchar(255) NOT NULL default ''	[SPACE-~]	OXDESC

¹⁰ Table: oxcategories

#	Content/Example	Data type	Range	Field name ¹⁰
14.	Long description	text NOT NULL	[SPACE-~]	OXLONGDESC
15.	Filename of Thumbnail (without path)	varchar(128) NOT NULL default "	[SPACE-~]	OXTHUMB
16.	External URL	varchar(255) NOT NULL default "	valid URL	OXEXTLINK
17.	Alt. Template	varchar(128) NOT NULL default "	[SPACE-~]	OXTEMPLATE
18.	Fast sorting(field name from oxarticles)	varchar(64) NOT NULL default "	[a-z]	OXDEFSORT
19.	sorting mode (asc/desc)	tinyint(1) NOT NULL default '0'	[0,1]	OXDEFSORTMODE
20.	Price from	double	- 1.7976931348623157 E+308 to - 2.2250738585072014 E-308, 0, and 2.2250738585072014 E-308 to 1.7976931348623157 E+308	OXPRICEFROM
21.	Price to	double NOT NULL	see above	OXPRICETO
22.	Active in language 1	char(1) NOT NULL default "	[0,1]	OXACTIV_1
23.	Title in language 1	varchar(255) NOT NULL default "	[SPACE-~]	OXTITLE_1
24.	Short description in language 1	varchar(255) NOT NULL default "	[SPACE-~]	OXDESC_1
25.	Long description in language 1	text NOT NULL	[SPACE-~]	OXLONGDESC_1
26.	Active in language 2	char(1) NOT NULL default "	[0,1]	OXACTIV_2
27.	Title in language 2	varchar(255) NOT NULL default "	[SPACE-~]	OXTITLE_2

#	Content/Example	Data type	Range	Field name ¹⁰
28.	Short description in language 2	varchar(255) NOT NULL default ''	[SPACE-~]	OXDESC_2
29.	Long description in language 2	text NOT NULL	[SPACE-~]	OXLONGDESC_2
30.	Active in language 3	char(1) NOT NULL default ''	[0,1]	OXACTIV_3
31.	Title in language 3	varchar(255) NOT NULL default ''	[SPACE-~]	OXTITLE_3
32.	Short description in language 3	varchar(255) NOT NULL default ''	[SPACE-~]	OXDESC_3
33.	Long description in language 3	text NOT NULL	[SPACE-~]	OXLONGDESC_3
34.	Filename of category image (without path)	varchar(128) NOT NULL	[SPACE-~]	OXICON
35.	Special VAT	float default NULL		OXVAT
36.	Skip all negative Discounts	tinyint(1) NOT NULL default '0'	[0,1]	OXSKIPDISCOUNTS
37.	Show SEO Suffix in Category	tinyint(1) NOT NULL default '1'	[0,1]	OXSHOWSUFFIX

4.6 Delete categories

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XK]	
1.	Category ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.7 Create/Update vendors

If a vendor with the given OXID already exists, the vendor is updated with the given values. Otherwise, a new vendor is created.

#	Content/Example	Data type	Range	Field name ¹¹
0.	Indicator char	char(1)	[H]	
1.	Vendor ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

¹¹ Table: oxvendor

#	Content/Example	Data type	Range	Field name ¹¹
2.	Shop ID of Shop in which vendor was created	char(32) NOT NULL default ''	1 to 64	OXSHOPID ²
3.	Vendor used in Shops	bigint(20) unsigned NOT NULL default '0'	binary encoded, 64 bit. <u>Example:</u> Category is in Shops #2 and #7. => 2nd 7th bit have to be set: The result is 66	OXSHOPINCL ²
4.	Vendor not used in Shops	bigint(20) unsigned NOT NULL default '0'	Binary encoded, see above	OXSHOPEXCL ³
5.	Active	tinyint(1) NOT NULL default '1'	[0,1]	OXACTIVE
6.	Filename of Icon	char(128) NOT NULL default ''	[SPACE-~]	OXICON
7.	Title	char(255) NOT NULL default ''	[SPACE-~]	OXTITLE
8.	Short description	char(255) NOT NULL default ''	[SPACE-~]	OXSHORTDESC
9.	Title in language 1	char(255) NOT NULL default ''	[SPACE-~]	OXTITLE_1
10.	Short description in language 1	char(255) NOT NULL default ''	[SPACE-~]	OXSHORTDESC_1
11.	Title in language 2	char(255) NOT NULL default ''	[SPACE-~]	OXTITLE_2
12.	Short description in language 2	char(255) NOT NULL default ''	[SPACE-~]	OXSHORTDESC_2
13.	Title in language 3	char(255) NOT NULL default ''	[SPACE-~]	OXTITLE_3
14.	Short description in language 3	char(255) NOT NULL default ''	[SPACE-~]	OXSHORTDESC_3
15.	Show SEO suffix	tinyint(1) NOT NULL default '1'	[0,1]	OXSHOWSUFFIX

4.8 Delete Vendor

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XH]	
1.	Vendor ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.9 Create/update crossselling assignment

If a crossselling assignment with the given OXID already exists, the crossselling assignment is updated with the given values. Otherwise, a new crossselling assignment is created.

#	Content/Example	Data type	Range	Field name ¹²
0.	Indicator char	char(1)	[C]	
1.	ID of product which is assigned as crossselling	char(32) NOT NULL default ''	[SPACE-~]	OXOBJECTID
2.	ID of the product to which the other product is assigned to.	char(32) NOT NULL default ''	[SPACE-~]	OXARTICLEID
3.	Sorting	int(5) NOT NULL default '0'	0 to 2147483647	OXSORT
4.	Crossselling assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.10 Delete Crossselling assignment

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XC]	
1.	Crossselling assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.11 Create/Update Accessory assignment

If an accessory assignment with the given OXID already exists, the accessory assignment is updated with the given values. Otherwise, a new accessory assignment is created.

#	Content/Example	Data type	Range	Field name ¹³
0.	Indicator char	char(1)	[Z]	
1.	ID of product which is assigned as accessory	char(32) NOT NULL default ''	[SPACE-~]	OXOBJECTID
2.	ID of the product to which the other product is assigned to.	char(32) NOT NULL default ''	[SPACE-~]	OXARTICLEID
3.	Sorting	int(5) NOT NULL default '0'	0 to 2147483647	OXSORT
4.	Accessory assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.12 Delete Accessory Assignment

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XZ]	
1.	Accessory assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

¹² Table: oxobject2article

¹³ Table: oxaccessoire2article

4.13 Create/Update Category assignments

If a category assignment with the given OXID already exists, the category assignment is updated with the given values. Otherwise, a new category assignment is created.

#	Content/Example	Data type	Range	Field name ¹⁴
0.	Indicator char	char(1)	[T]	
1.	category assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID
2.	Shop-ID	integer(11) NOT NULL	1 to 64	OXSHOPID ²
3.	Assignment used in Shops	bigint(20) NOT NULL	binary encoded, 64 bit. <u>Example:</u> Category is in Shops #2 and #7. => 2nd 7th bit have to be set: The result is 66	OXSHOPINCL ²
4.	Assignment not used in Shops	bigint(20) NOT NULL	see above	OXSHOPEXCL ³
5.	Product ID	char(32) NOT NULL default ''	[0-9a-f.]	OXOBJECTID
6.	Category ID	char(32) NOT NULL default ''	[0-9a-f.]	OXCATNID
7.	Sorting	int(11) NOT NULL default '0'	0 to 2147483647	OXPOS
8.	Timestamp (The entry with the oldest timestamp is automatically set as main category)	int(11) NOT NULL default '0'	0 to 2147483647	OXTIME

4.14 Set main category

If a main category is already set, the setting is overwritten.

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(1)	[M]	
1.	Product ID	char(32) NOT NULL default ''	[0-9a-f.]	OXOBJECTID
2.	Category ID	char(32) NOT NULL default ''	[0-9a-f.]	OXCATNID

4.15 Delete Category assignment

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XT]	
1.	Category Assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

¹⁴ Table: oxobject2category

4.16 Create/update action assignment

If an action assignment with the given OXID already exists, the action assignment is updated with the given values. Otherwise, a new action assignment is created.

#	Content/Example	Data type	Range	Field name ¹⁵
0.	Indicator char	char(1)	[I]	
1.	Shop-ID	char(32) NOT NULL default ''	1 – 64	OXSHOPID ²
2.	Action ID	char(32) NOT NULL default ''	<u>oxstart</u> front page <u>oxtopstart</u> Top offer front page <u>oxfirststart</u> Big offer front page <u>oxbargain</u> Bargain <u>oxtop5</u> Top of the Shop <u>oxcatoffer</u> Category offer <u>oxnewest</u> Just arrived	OXACTIONID
3.	Product ID	char(32) NOT NULL default ''	[0-9a-f.]	OXARTID
4.	Sorting	int(11) NOT NULL default '0'	0 to 2147483647	OXSORT
5.	Action assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.17 Delete action assignment

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XI]	
1.	Action Assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.18 Create/update scale prices

If a scale price with the given OXID already exists, the scale price is updated with the given values. Otherwise, a new scale price is created.

#	Content/Example	Data type	Range	Field name ¹⁶
0.	Indicator char	char(1)	[P]	

¹⁵ Table: oxactions2article

¹⁶ Table: oxprice2article

1.	Shop ID	char(32) NOT NULL default ""	1 to 64	OXSHOPID ²
2.	Product ID	char(32) NOT NULL default ""	[0-9a-f.]	OXARTID
3.	Absolute surcharge	double NOT NULL default '0'	- 1.7976931348623157 E+308 to - 2.2250738585072014 E-308, 0, und 2.2250738585072014 E-308 to 1.7976931348623157 E+308	OXADDABS
4.	Relative surcharge in %	double NOT NULL default '0'	siehe Feld 3	OXADDPERC
5.	Amount from	double NOT NULL default '0'	siehe Feld 3	OXAMOUNT
6.	Amount to	double NOT NULL default '0'	siehe Feld 3	OXAMOUNTTO
7.	Scale price ID	char(32) NOT NULL default ""	[0-9a-f.]	OXID

4.19 Delete scale prices

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XP]	
1.	Scale price ID	char(32) NOT NULL default ""	[0-9a-f.]	OXID

4.20 Create/update orders

If an order with the given OXID already exists, the order is updated with the given values. Otherwise, a new order is created.

If `OXORDERNR` is set to 0, the eShop automatically creates a new order number: The highest existing order number is taken and incremented by 1.

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(1)	[0]	
1.	Order ID	char(32) NOT NULL default ""	[0-9a-f.]	OXID

#	Content/Example	Data type	Range	Field name
2.	Shop ID	char(32) NOT NULL default ''	1 to 64	OXSHOPID ²
3.	User ID	char(32) NOT NULL default ''	[0-9a-f.]	OXUSERID
4.	Order date	datetime NOT NULL default '0000-00-00 00:00:00'	YYYY-MM-DD HH:MM:SS	OXORDERDATE
5.	Order number	int(11) NOT NULL default '0'	0 to 2147483647	OXORDERNR
6.	Billing address: company	varchar(255) NOT NULL default ''	[SPACE-~]	OXBILLCOMPANY
7.	Billing address: e-mail	varchar(255) NOT NULL default ''	Gültige E-Mail- Adresse	OXBILLEMAIL
8.	Billing address: first name	varchar(255) NOT NULL default ''	[SPACE-~]	OXBILLFNAME
9.	Billing address: last name	varchar(255) NOT NULL default ''	[SPACE-~]	OXBILLNAME
10.	Billing address: Street	varchar(255) NOT NULL default ''	[SPACE-~]	OXBILLSTREET
11.	Billing address: street number	varchar(16) NOT NULL default ''	[SPACE-~]	OXBILLSTREETNR
12.	Billing address: Additional information	varchar(255) NOT NULL default ''	[SPACE-~]	OXBILLADDINFO
13.	Billing address: VAT ID	varchar(255) NOT NULL default ''	[SPACE-~]	OXBILLUSTID
14.	VAT ID status	tinyint(1) NOT NULL default '0'	[0;1]	OXBILLUSTIDSTATUS ³
15.	Billing address: city	varchar(255) NOT NULL default ''	[SPACE-~]	OXBILLCITY
16.	Billing address: country	varchar(32) NOT NULL default ''	[SPACE-~]	OXBILLCOUNTRYID
17.	Billing address: ZIP	varchar(16) NOT NULL default ''	[SPACE-~]	OXBILLZIP
18.	Billing address: phone	varchar(128) NOT NULL default ''	[SPACE-~]	OXBILLFON
19.	Billing address: fax	varchar(128) NOT NULL default ''	[SPACE-~]	OXBILLFAX
20.	Billing address: title	varchar(128) NOT NULL default ''	[SPACE-~]	OXBILLTITEL
21.	Shipping address: company	varchar(255) NOT NULL default ''	[SPACE-~]	OXDELCOMPANY

#	Content/Example	Data type	Range	Field name
22.	Shipping address: first name	varchar(255) NOT NULL default ''	[SPACE-~]	OXDELFNAME
23.	Shipping address: last name	varchar(255) NOT NULL default ''	[SPACE-~]	OXDELLNAME
24.	Shipping address: street	varchar(255) NOT NULL default ''	[SPACE-~]	OXDELSTREET
25.	Shipping address: street number	varchar(16) NOT NULL default ''	[SPACE-~]	OXDELSTREETNR
26.	Shipping address: additional information	varchar(255) NOT NULL default ''	[SPACE-~]	OXDELADDINFO
27.	Shipping address: city	varchar(255) NOT NULL default ''	[SPACE-~]	OXDELCITY
28.	Shipping address: country	varchar(32) NOT NULL default ''	[SPACE-~]	OXDELCOUNTRYID
29.	Shipping address: ZIP	varchar(16) NOT NULL default ''	[SPACE-~]	OXDELZIP
30.	Shipping address: phone	varchar(128) NOT NULL default ''	[SPACE-~]	OXDELFON
31.	Shipping address: fax	varchar(128) NOT NULL default ''	[SPACE-~]	OXDELFAX
32.	Shipping address: title	varchar(128) NOT NULL default ''	[SPACE-~]	OXDELSAL
33.	Payment ID(from table oxuserpayments)	varchar(32) NOT NULL default ''	[SPACE-~]	OXPAYMENTID
34.	Payment type (from table oxpayments)	varchar(32) NOT NULL default ''	[SPACE-~]	OXPAYMENTTYPE
35.	Net sum	double NOT NULL default '0'	- 1.79769313486231 57 E+308 to - 2.22507385850720 14 E-308, 0, und 2.22507385850720 14 E-308 to 1.79769313486231 57 E+308	OXTOTALNETSUM
36.	Gross sum	double NOT NULL default '0'	see above	OXTOTALBRUTSUM
37.	Total sum (incl. Shipping costs etc.)	double NOT NULL default '0'	see above	OXTOTALORDERSUM

#	Content/Example	Data type	Range	Field name
38.	Shipping costs	double NOT NULL default '0'	see above	OXDELCOST
39.	VAT of shipping costs	double NOT NULL default '0'	see above	OXDELVAT
40.	Payment costs	double NOT NULL default '0'	see above	OXPAYCOST
41.	VAT of payment costs	double NOT NULL default '0'	see above	OXPAYVAT
42.	Gift wrapping costs	double NOT NULL default '0'	see above	OXWRAPCOST
43.	VAT of gift wrapping	double NOT NULL default '0'	see above	OXWRAPVAT
44.	Greeting card ID	varchar(32) NOT NULL default ''	[0-9a-f.]	OXCARDID
45.	Greeting card text	text NOT NULL	[SPACE-~]	OXCARDTEXT
46.	Discount	double NOT NULL default '0'	see field 35	OXDISCOUNT
47.	Export status for Lexware	tinyint(4)	0,1	OXEXPORT
48.	Invoice number	varchar(128) NOT NULL default ''	[0-9]	OXBILLNR
49.	Tracking code	varchar(128) NOT NULL	[SPACE-~]	OXTRACKCODE
50.	Shipping date	Datetime	YYYY-MM-DD HH:MM:SS	OXSENDDATE
51.	Remark	text NOT NULL	[SPACE-~]	OXREMARK
52.	Coupon discount	double NOT NULL default '0'	see field 35	OXVOUCHERDISCOUNT
53.	Currency	char(32) NOT NULL default ''	[SPACE-~]	OXCURRENCY
54.	Conversion factor	double NOT NULL default '0'	see field 35	OXCURRATE
55.	Order folder in admin	varchar(32)	[SPACE-~]	OXFOLDER
56.	Payment gateway transaction ident code	varchar(128)	[SPACE-~]	OXPIDENT
57.	Payment gateway Transaction ID	varchar(64) NOT NULL default ''	[SPACE-~]	OXTRANSID
58.	Payment gateway payment ID	varchar(64) NOT NULL default ''	[SPACE-~]	OXPAYID
59.	Payment gateway X-ID	varchar(64) NOT NULL default ''	[SPACE-~]	OXXID
60.	Paid on	varchar(64) NOT NULL default ''	YYYY-MM-DD HH:MM:SS	OXPAID

#	Content/Example	Data type	Range	Field name
61.	Order cancelled	tinyint(1)	[0,1]	OXSTORNO
62.	IP number	varchar(64) NOT NULL default ''	valid IP	OXIP
63.	Payment gateway transaction status	varchar(30) NOT NULL default ''	[SPACE-~]	OXTRANSSTATUS
64.	language	int(2) NOT NULL default '0'	0 to 99	OXLANG
65.	Invoice number	integer(11) NOT NULL	0 to 2147483647	OXINVOICENR
66.	Shipping method	char(32) NOT NULL default ''	[SPACE-~]	OXDELTYPE
67.	Export status for pixi*	tinyint(4) NOT NULL	[0,1]	OXPIXIEXPORT ³

4.21 Delete orders

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XO]	
1.	Order ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.22 Create/update product assignments to orders

If a product assignment with the given OXID already exists, the product assignment is updated with the given values. Otherwise, a new product assignment is created.

#	Content/Example	Data type	Range	Field name ¹⁷
0.	Indicator char	char(1)	[R]	
1.	Product assignment ID	char(32) NOT NULL	[0-9a-f.]	OXID
2.	Order ID	char(32) NOT NULL	[0-9a-f.]	OXORDERID
3.	Amount	double NOT NULL default '0'	see field 9	OXAMOUNT
4.	Product ID	char(32) NOT NULL	[0-9a-f.]	OXARTID
5.	Product number	varchar(255) NOT NULL	[SPACE-~]	OXARTNUM
6.	Title	varchar(255) NOT NULL	[SPACE-~]	OXTITLE
7.	Short description	varchar(255) NOT NULL	[SPACE-~]	OXSHORTDESC
8.	Selected variant	varchar(255) NOT NULL	[SPACE-~]	OXSELVARIANT

¹⁷ Table: oxorderarticles

#	Content/Example	Data type	Range	Field name ¹⁷
9.	Total sum net	double NOT NULL default '0'	- 1.7976931348623157 E+308 to - 2.2250738585072014 E-308, 0, und 2.2250738585072014 E-308 to 1.7976931348623157 E+308 ¹⁸	OXNETPRICE
10.	Total sum brut	double NOT NULL default '0'	see field 9	OXBRUTPRICE
11.	VAT Total	double NOT NULL default '0'	see field 9	OXVATPRICE
12.	Special VAT	double NOT NULL	see field 9	OXVAT
13.	Persistent parameters (not used in standard eShop)	Text NOT NULL	[SPACE-~]	OXPERSPARAM
14.	Product price	double NOT NULL default '0'	see field 9	OXPRICE
15.	Brut product price	double NOT NULL default '0'	see field 9	AXBPRICE
16.	Net product price	double NOT NULL default '0'	see field 9	OXNPRICE
17.	Gift wrapping ID	varchar(32)	[SPACE-~]	OXWRAPID
18.	External URL	varchar(255) NOT NULL default ''	valid URL without <i>http://</i>	OXEXTURL
19.	Description for external URL	varchar(255) NOT NULL default ''	[SPACE-~]	OXURLDESC
20.	URL of picture for external URL	varchar(128) NOT NULL default ''	see field 23	OXURLIMG
21.	Filename (without path) , Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~] ¹⁹	OXTHUMB
22.	Filename (without path) , Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC1
23.	Filename (without path) , Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	XPIC2

¹⁸ Ranges defined by IEEE. The range that can be used is determined by hardware and operating system

¹⁹ Dependent on operating system

#	Content/Example	Data type	Range	Field name ¹⁷
24.	Filename (without path) , Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	OXPIC3
25.	Filename (without path) , Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	OXPIC4
26.	Filename (without path) , Default: <i>nopic.jpg</i>	varchar(128) NOT NULL default ''	[SPACE-~]	OXPIC5
27.	Weight	double NOT NULL default '0'	see field 9	OXWEIGHT
28.	Stock	int(11) NOT NULL default '-1' ²⁰	-2147483648 to 2147483647	OXSTOCK
29.	Available on	date NOT NULL default '0000-00-00'	YYYY-MM-DD	OXDELIVERY
30.	Date of creation (is ignored, actual date is used instead)	date NOT NULL default '0000-00-00'	YYYY-MM-DD	OXINSERT
31.	Date of last change (is ignored, actual date is used instead)	timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP	YYYY-MM-DD HH:MM:SS	OXTIMESTAMP
32.	Length	double NOT NULL default '0'	see field 9	OXLENGTH
33.	Width	double NOT NULL default '0'	see field 9	OXWIDTH
34.	Height	double NOT NULL default '0'	see field 9	OXHEIGHT
35.	File	varchar(128) NOT NULL default ''	[SPACE-~]	OXFILE
36.	Search terms	varchar(255) NOT NULL default ''	[SPACE-~]	OXSEARCHKEYS
37.	Alt. template	varchar(128) NOT NULL default ''	[SPACE-~]	OXTEMPLATE
38.	Alt. contact	varchar(255) NOT NULL default ''	valid e-mail address	OXQUESTIONEMAIL
39.	Searchable	tinyint(1) NOT NULL default '1'	[0,1]	OXISSEARCH
40.	(always empty)	varchar(32) NOT NULL default ''	always empty	OXFOLDER

²⁰

-1 was a special value in older versions. Negative values are not to be used in Enterprise Edition any more.

#	Content/Example	Data type	Range	Field name ¹⁷
41.	Class	varchar(32) NOT NULL default ''	always <i>oxarticle</i>	OXSUBCLASS
42.	Cancelled	Tinyint(1) NOT NULL default '1'	[0,1]	OXSTORNO
43.	Shop-ID	varchar(32) NOT NULL	[SPACE-~]	OXORDERSHOPID ²
44.	VAT Total	Double NOT NULL	see field 9	OXTOTALVAT
45.	Order status	text	See 4.24	OXERPSTATUS ²¹

4.23 Delete product assignments to orders

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XR]	
1.	Product assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.24 Change order status

If an order status with the given OXID already exists, the order status is updated with the given values. Otherwise, a new order status is created.

#	Content/Example	Data type	Range	Field name ²²
0.	Indicator char	char(1)	[S]	
1.	ID from <i>oxorderarticles</i>	char(32) NOT NULL default ''	[0-9a-f.]	OXID
2.	Status text	text NOT NULL	[SPACE-~]	OXERPSTATUS_STATUS ²¹
3.	Date and time	Datetime	YYYY-MM-DD HH:MM:SS	OXERPSTATUS_TIME ²¹
4.	Tracking ID	char(32) NOT NULL default ''	[SPACE-~]	OXERPSTATUS_TRACKID ²¹

4.25 Create/update product assignments to vendors

If a product is already assigned to a vendor, this assignment is overwritten.

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(1)	[D]	
1.	Product ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID
2.	Vendor ID	char(32) NOT NULL default ''	[0-9a-f.]	OXVENDORID

²¹ Always empty in OXID eShop Professional Edition

²² Table: *oxorderarticles*

4.26 Delete product assignments to vendors

#	Content/Example	Data type	Range	Field name
2.	Indicator char	char(2)	[XD]	
3.	Product assignment ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID

4.27 Updating stock

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(1)	[E]	
1.	Product ID	char(32) NOT NULL default ''	[0-9a-f.]	OXID
2.	Stock	Integer(11) NOT NULL default -1	-2147483648 to 2147483647	OXSTOCK

4.28 Create/update users

If a user with the given OXID already exists, the user is updated with the given values. Otherwise, a new user is created.

#	Content/Example	Data type	Range	Field name ²³
0.	Indicator char	char(1)	[U]	
1.	User ID	char(32) NOT NULL	[0-9a-f.]	OXID
2.	Active	tinyint(1) NOT NULL default 0	[0,1]	OXACTIVE
3.	Rights	char(32)	'user', 'malladmin' or subshop ID	OXRIGHTS
4.	Shop ID	char(32)	[SPACE-~]	OXSHOPID ²
5.	Login/e-mail address	varchar(255) NOT NULL	[SPACE-~]	OXUSERNAME
6.	Password	varchar(128) NOT NULL	[SPACE-~]	OXPASSWORD
7.	Salt (see also chapter 5.18)	char(128)	[SPACE-~]	OXPASSSALT
8.	Customer number	Integer(11)	0 to 2147483647	OXCUSTNR
9.	VAT ID	varchar(255) NOT NULL	[SPACE-~]	OXUSTID
10.	VAT ID status	tinyint(1) NOT NULL default '0'	[0,1]	OXUSTIDSTATUS ³
11.	Company	varchar(255) NOT NULL	[SPACE-~]	OXCOMPANY

²³ Table: oxuser

#	Content/Example	Data type	Range	Field name ²³
12	First name	varchar(255) NOT NULL	[SPACE-~]	OXFNAME
13	Last name	varchar(255) NOT NULL	[SPACE-~]	OXLNAME
14	Street	varchar(255) NOT NULL	[SPACE-~]	OXSTREET
15	Street number	varchar(16) NOT NULL	[SPACE-~]	OXSTREETNR
16	Additional information	varchar(255) NOT NULL	[SPACE-~]	OXADDINFO
17	City	varchar(255) NOT NULL	[SPACE-~]	OXCITY
18	Country ID	varchar(32) NOT NULL	[SPACE-~]	OXCOUNTRYID
19	ZIP	varchar(16) NOT NULL	[SPACE-~]	OXZIP
20	Phone number	varchar(128) NOT NULL	[SPACE-~]	OXFON
21	Fax number	varchar(128) NOT NULL	[SPACE-~]	OXFAX
22	Salutation	varchar(128) NOT NULL	[SPACE-~]	OXSAL
23	Credit rating	integer(11)	0 to 2147483647	AXBONI
24	Date of creation	datetime	YYYY-MM-DD HH:MM:SS	OXCREATE
25	Date of registration	datetime	YYYY-MM-DD HH:MM:SS	OXREGISTER
26	Private phone number	varchar(64) NOT NULL	[SPACE-~]	OXPRIVFON
27	Cellular phone number	varchar(64) NOT NULL	[SPACE-~]	OXMOBFON
28	Date of birth	date	YYYY-MM-DD	OXBIRTHDATE
29	URL	varchar(255) NOT NULL	[SPACE-~]	OXURL
30	Automatic user group assignment	tinyint(1) NOT NULL	[0,1]	OXDISABLEAUTOGRP
31	LDAP-Key	varchar(128) NOT NULL	[SPACE-~]	OXLDAPKEY ³
32	Amount of invalid login attempts	integer(11)	0 to 2147483647	OXWRONGLOGINS ³
33	Key for „forgot password“ function	char(32)	[SPACE-~]	OXUPDATEKEY

#	Content/Example	Data type	Range	Field name ²³
34	Timestamp when key from field 33 expires	integer(11)	0 to 2147483647	OXUPDATEEXP
35	OpenID Indicates if account was created by OpenID	tinyint(1)	[0,1]	OXISOPENID

4.29 Delete users

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XU]	
1.	User ID	char(32) NOT NULL	[0-9a-f.]	OXID

4.30 Create/Update countries

If a country with the given OXID already exists, the country is updated with the given values. Otherwise, a new country is created.

#	Content/Example	Data type	Range	Field name ²⁴
0.	Indicator char	char(1)	[N]	
1.	Country ID	char(32) NOT NULL	[0-9a-f.]	OXID
2.	Active	tinyint(1) NOT NULL default 0	[1,0]	OXACTIVE
3.	Name	char(128) NOT NULL	[SPACE-~]	OXTITLE
4.	ISO 2	char(2) NOT NULL	[A-Z]	OXISOALPHA2
5.	ISO 3	char(3) NOT NULL	[A-Z]	OXISOALPHA3
6.	ISO UN number	char(3) NOT NULL	[0-999]	OXUNNUM3
7.	Sorting	int(11) NOT NULL default 9999	0 to 2147483647	OXORDER
8.	Short description	char(255) NOT NULL	[SPACE-~]	OXSHORTDESC
9.	Description	char(255) NOT NULL	[SPACE-~]	OXLONGDESC
10	Name in language 1	char(128) NOT NULL	[SPACE-~]	OXTITLE_1
11	Name in language 2	char(128) NOT NULL	[SPACE-~]	OXTITLE_2
12	Name in language 3	char(128) NOT NULL	[SPACE-~]	OXTITLE_3

²⁴ Table: oxcountry

#	Content/Example	Data type	Range	Field name ²⁴
13	Short description in language 1	char(128) NOT NULL	[SPACE-~]	OXSHORTDESC_1
14	Short description in language 2	char(128) NOT NULL	[SPACE-~]	OXSHORTDESC_2
15	Short description in language 3	char(128) NOT NULL	[SPACE-~]	OXSHORTDESC_3
16	Description in language 1	char(255) NOT NULL	[SPACE-~]	OXLONGDESC_1
17	Description in language 2	char(255) NOT NULL	[SPACE-~]	OXLONGDESC_2
18	Description in language 3	char(255)	[SPACE-~]	OXLONGDESC_3
19	VAT Handling	tinyint(1) NOT NULL default 0	[0,1]	OXVATSTATUS

4.31 Delete Countries

#	Content/Example	Data type	Range	Field name
0.	Indicator char	char(2)	[XN]	
1.	Country ID	char(32) NOT NULL	[0-9a-f.]	OXID

5 Export interface

All objects (products, users, categories and so on) can be exported. The structure of the exported file is the same as described above. For accessing import, a particular URL is called via browser. We strongly recommend to password-protect and SSL-protect (https://) the export so it's not openly accessible.

5.1 Sorting export data

In some exports you can set by which field and in which direction (ASC or DESC) the data is to be sorted. If no sorting parameter is given, the fields are sorted by OXID.

5.2 Login

Before CSV Data can be exported, you have to login first. This is done with OXERPLLogin function:
<https://www.youreshop.com/modules/ERP/oxerpcsvexport.php?fnc=OXERPLLogin&user=admin&pw=password&version=2.3.0&shopid=1&langid=0>

The following parameters are given:

Parameter	Description
fnc=	Name of function (here: OXERPLLogin)
user=	Admin name (here: admin)
pw=	Password (here: password)
shopid=	Shop ID of Shop in which login is done (here: 1)
langid=	Language ID (here: 0)

The function returns a Session ID which has to be used in every other export function.

5.3 OXERPGetCategory

This function exports a category.

5.3.1 Parameters

Parameter	Description
fnc=OXERPGetCategory	Function name
sid=	Session ID returned by OXERPLLogin.
id=	ID of the category.

5.3.2 Example

<https://www.youreshop.com/modules/ERP/oxerpcsvexport.php?fnc=OXERPGetCategory&sid=xxxxxxxxx&version=2.3.0&id=yyyyyyyyyy>

5.4 OXERPGetCategories

This function exports several categories at once.

5.4.1 Parameters

Parameter	Description
fnc=OXERPGetCategories	Function name
sid=	Session ID returned by OXERPLLogin.
sSortFieldName= (optional)	The field the returned categories are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.4.2 Example

<https://www.youreshop.com/modules/ERP/oxerpcsvexport.php?fnc=OXERPGetCategories&sid=xxxxx&version=2.3.0&sSortFieldName=OXID&sSortType=DESC&iStart=0&iCount=5>

5.5 OXERPGetArticle

This function exports a product.

5.5.1 Parameters

Parameter	Description
fnc=OXERPGetArticle	Function name
sid=	Session ID returned by OXERPLogin.
id=	ID of the product.

5.6 OXERPGetArticles

This function exports several products at once.

5.6.1 Parameters

Parameter	Description
fnc=OXERPGetArticles	Function name
sid=	Session ID returned by OXERPLogin.
sSortFieldName= (optional)	The field the returned products are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.
id= (optional)	Category ID. Only products assigned to this category are exported.

5.7 OXERPGetArticleCategory

This function exports the categories a product is assigned to.

5.7.1 Parameters

Parameter	Description
fnc=OXERPGetArticleCategory	Function name
sid=	Session ID returned by OXERPLogin.
id=	Product ID
sSortFieldName= (optional)	The field the returned categories are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.8 OXERPGetArticleAction

This function exports the actions a product is assigned to.

5.8.1 Parameters

Parameter	Description
fnc=OXERPGetArticleAction	Function name
sid=	Session ID returned by OXERPLLogin.
id=	The ID of the product.
sSortFieldName= (optional)	The field the returned actions are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.9 OXERPGetArticleCrossselling

This function exports the products which are assigned as crosssellings to another product.

5.9.1 Parameters

Parameter	Description
fnc=OXERPGetArticleCrossselling	Function name
sid=	Session ID returned by OXERPLLogin.
id=	The ID of the product.
sSortFieldName= (optional)	The field the returned products are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.10 OXERPGetArticleScalePrice

This function exports scale prices assigned to a product.

5.10.1 Parameters

Parameter	Description
fnc=OXERPGetArticleScalePrice	Function name
sid=	Session ID returned by OXERPLLogin.

id=	The ID of the product.
sSortFieldName= (optional)	The field the returned scale prices are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.11 OXERPGetArticleAccessoire

This function returns products which are assigned as accessories to another product.

5.11.1 Parameters

Parameter	Description
fnc=OXERPGetArticleAccessoire	Function name
sid=	Session ID returned by OXERPLogin.
id=	The ID of the product.
sSortFieldName= (optional)	The field the returned products are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.12 OXERPGetArticleStock

This function exports the stock of a product.

5.12.1 Parameters

Parameter	Description
fnc=OXERPGetArticleStock	Function name
sid=	Session ID returned by OXERPLogin.
id=	The ID of the product.

5.13 OXERPGetOrder

This function exports an order.

5.13.1 Parameters

Parameter	Description
fnc=OXERPGetOrder	Function name
sid=	Session ID returned by OXERPLLogin.
id=	ID of the order.

5.14 OXERPGetOrderArticle

This function exports products of assigned to an order.

5.14.1 Parameters

Parameter	Description
fnc=OXERPGetOrderArticle	Function name
sid=	Session ID returned by OXERPLLogin.
id=	The OXID of the order
sSortFieldName= (optional)	The field the returned products are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.15 OXERPGetVendor

This function exports a vendor.

5.15.1 Parameters

Parameter	Description
fnc=OXERPGetVendor	Function name
sid=	Session ID returned by OXERPLLogin.
id=	ID of the vendor.

5.16 OXERPGetArticleVendor

This function exports the vendor of a product.

5.16.1 Parameters

Parameter	Description
fnc=OXERPGetArticleVendor	Function name
sid=	Session ID returned by OXERPLLogin.
id=	ID of the product.

5.17 OXERPGetVendorArticles

This function returns all products assigned to a vendor.

5.17.1 Parameters

Parameter	Description
fnc=OXERPGetVendorArticles	Function name
sid=	Session ID returned by OXERPLLogin.
id=	The ID of the vendor.
sSortFieldName= (optional)	The field the returned products are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.18 OXERPGetUser

This function exports a user.

Since OXID eShop version 4.0.1.0, passwords are MD5 encrypted with a salt:

```
OXUSER.OXPASSWORD = MD5 (PASSWORD+SALT)
```

These passwords cannot be decrypted. They cannot be imported in older eShop versions. The salt is always a HEX value.

5.18.1 Parameters

Parameter	Description
fnc=OXERPGetUser	Function name
sid=	Session ID returned by OXERPLLogin.
id=	The ID of the user.

5.19 OXERPGetUsers

This function exports several users at once.

5.19.1 Parameters

Parameter	Description
fnc=OXERPGetUsers	Function name
sid=	Session ID returned by OXERPLLogin.

Parameter	Description
sSortFieldName= (optional)	The field the returned users are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.
timestamp=yyyy-mm-dd hh:mm:ss (optional)	Only users created after this point of time are exported.

5.20 OXERPGetOrders

This function exports several users at once.

5.20.1 Parameters

Parameter	Description
fnc=OXERPGetOrders	Function name
sid=	Session ID returned by OXERPLLogin.
sSortFieldName= (optional)	The field the returned orders are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.
timestamp=yyyy-mm-dd hh:mm:ss (optional)	Alle Bestellungen werden exportiert, die an und nach diesem Zeitpunkt angelegt wurden.

5.21 OXERPGetVendors

This function exports several vendors

5.21.1 Parameters

Parameter	Description
fnc=OXERPGetVendors	Function name
sid=	Session ID returned by OXERPLLogin.
sSortFieldName= (optional)	The field the returned vendors are sorted by.
sSortType= (optional)	ASC (ascending) or DESC (descending)

Parameter	Description
iStart= (optional)	Amount of objects which are skipped and not returned.
iCount= (optional)	Limits amount of returned objects.

5.22 OXERPCallPlugin

OXERPCallPlugin invokes an external plugin. Sample call:

`http://shopurl/modules/erp/oxerpcsvexport.php?sid=MYSID&fnc=OXERPCallPlugin&sPluginName=examplePlugin&aRequestData[var1]=val1&aRequestData[var2]=val2`

5.22.1 Parameters

Parameter	Description
fnc=OXERPCallPlugin	Function name
sid=	Session ID returned by OXERPLogin.
sPluginName=	The name of a plugin to be called.
aRequestData=	Data passed to the plugin

6 Using custom fields

If additional fields were created in tables, the ERP module files have to be adjusted. The files are stored in `/modules/erp/objects`. The filename indicates for which table file is used. At the beginning, the files contain a field list. This list has to be extended with the names of the additional fields.

6.1 Example

In this example, the table `oxcategories` was extended with the field `articles_amount`. This field contains the amount of products assigned to the category. For using this field, `oxerptype_oxcategory.php` is adjusted:

The file contains a definition of database fields:

```
'OXID'           => 'OXID',  
'OXPARENTID'    => 'OXPARENTID',  
'OXORDER'       => 'OXORDER',  
'OXACTIV'       => 'OXACTIV',
```

This definition is extended with `articles_amount`:

```
.  
.   
.   
'OXSEOID_3'      => 'OXSEOID_3',  
'ARTICLES_AMOUNT' => 'ARTICLES_AMOUNT'  
);
```

7 Extending ERP with plugins

With plugins, you can extend the functionality of the CSV interface. Plugins are called using the function `OXERPCallPlugin` (See 5.22).

7.1 Creating a plugin

A Plugin class has to extend the class `oxErpPluginBase` and implement `invoke()` function (which is defined abstract by `oxErpPluginBase`). The class name has to contain *Plugin* at the end (e.g. *examplePlugin* as below).

The Plugins are stored in in the `<erp base>/plugins` directory. The filename has to be the same as the class name of the plugin **in lowercase** (e.g. *exampleplugin.php* for the example below).

7.2 Sample plugin

```
<?php
/**
 * plugin which duplicates input
 */
class examplePlugin extends oxErpPluginBase
{
    /**
     * plugin entry point
     *
     * @param array $aParams 'name=>value' based passed parameters map
     *
     * @return oxERPReturn[]
     */
    public function invoke($aParams)
    {
        $oResult = new oxERPReturn();
        $oResult->aResult = $aParams;
        $oResult->blResult = true;
        $oResult->sMessage = 'example msg';
        return array($oResult, clone $oResult);
    }
}
```

7.2.1 Data flow of the sample plugin

Assuming data of a `csvexport` request is:

```
http://...&fnc=OXERPCallPlugin&sPluginName=examplePlugin&aRequestData[myvarname1]=value1&
aRequestData[myvarname2]=value2
```

The plugin's `invoke` method gets:

```
array (  
    'myvarname1' => 'value1',  
    'myvarname2' => 'value2',  
)
```

The example plugin duplicates data, and returns it to the call handler. The result data is:

```
array (  
    0 =>  
        OXERPReturn::__set_state(array(  
            'blResult' => true,  
            'sMessage' => 'example msg',  
            'aResult' =>  
                array (  
                    'myvarname1' => 'value1',  
                    'myvarname2' => 'value2',  
                ),  
            ),  
        )),  
    1 =>  
        OXERPReturn::__set_state(array(  
            ... [same as 0th entry] ...  
        )),  
)
```

Such result is formatted (by `oxERPGateway`) according to the request type (SOAP or CSV) and returned to the client.

CSV client would get duplicated input in csv format:

```
myvarname1,myvarname2  
value1,value2  
value1,value2
```

The first line of the output file is a header line, containing variable names (array key values of the first data record, returned by the plugin, if resulting lines column count is equal). If column count in data is not equal, header line is skipped.

8 Expandability of CSV interface

You can also extend the CSV interface with own functions, but we advise you to use plugins (See chapter 7) for keeping the ERP easily updatable.

How to extend CSV interface is explained with an example: The import of attributes is implemented. As a guideline, oxcountry entries can be taken.

8.1 Define new attribute object

For accessing attributes, an attribute object has to be defined. Existing object files are stored in *modules/objects*. Here, a new file has to be created: *oxerptype_attribute.php*:

```
<?php

require_once( 'oxerptype.php' );

class oxERPTType_Attribute extends oxERPTType
{
    public function __construct()
    {
        global $myConfig;

        parent::__construct();

        $this->_sTableName = 'oxattribute';
        $this->_sShopObjectName = 'oxattribute';

        $this->_aFieldList = array(
            'OXID'          => 'OXID',
            'OXSHOPID'     => 'OXSHOPID',
            'OXSHOPINCL'   => 'OXSHOPINCL',
            'OXSHOPEXCL'   => 'OXSHOPEXCL',
            'OXTITLE'      => 'OXTITLE',
            'OXTITLE_1'    => 'OXTITLE_1',
            'OXTITLE_2'    => 'OXTITLE_2',
            'OXTITLE_3'    => 'OXTITLE_3',
            'OXPOS'        => 'OXPOS'
        );
    }
}
```

In `$this->_aFieldList`, all database fields from *oxattribute* table are defined.

8.2 Implementing function in oxerpgateway.php

```
public function OXERPSetAttribute($oParameters)
```

```

{
    try{
        $this->_loadSession($oParameters->sSessionID);
        $aAttribute = $this->_formatInput($oParameters->aAttribute);
        $oResult = $this->_import('attribute',$aAttribute);
    }
    catch (Exception $e) {
        $oResult = $this->_exception2OxErpReturn($e);
    }
    return $this->_formatResult($oResult,'OXERPSetAttributeResult',true);
}

```

8.3 Import definition in oxerpcsv.php

In *oxerpcsv.php* the function `_ImportAttribute` has to be implemented:

```

protected function _ImportAttribute(oxERPTType & $oType, $aRow)
{
    $sResult = $this->_Save( $oType, $aRow);
    $this->_aResults[$this->_iIdx] = $sResult;
    return (boolean) $sResult;
}

```

In *oxerpcsv.php* the indicator chars are defined in `$_aObjects` array. For attribute import, we choose 'B' as indicator char and add it to `$_aObjects` array:

```
'B' => 'attribute',
```

Take care that each indicator char is used only once.

8.4 Using new OXERPSetAttribute function

If an attribute with the given OXID already exists, the attribute is updated with the given values. Otherwise, a new attribute is created.

#	Content/Example	Data type	Range	Field name ²⁵
0.	Indicator char	char(1)	[B]	
1.	Attribute ID	char(32) NOT NULL	[0-9a-f.]	OXID
2.	Shop-ID	char(32) NOT NULL	1 - 64	OXSHOPID ²

²⁵ Table: oxattribute

#	Content/Example	Data type	Range	Field name ²⁵
3.	Attribute used in Shops	bigint(20) NOT NULL UNSIGNED	binary encoded, 64 bit. <u>Example:</u> Product is in Shops #2 and #7. => 2nd 7th bit have to be set: The result is 66	OXSHOPINCL ²
4.	Attribute not used in Shops	bigint(20) NOT NULL	binary encoded, 64 bit. See above.	OXSHOPEXCL ³
5.	Name	char(128) NOT NULL	[SPACE-~]	OXTITLE
6.	Name in language 1	char(128) NOT NULL	[SPACE-~]	OXTITLE_1
7.	Name in language 2	char(128) NOT NULL	[SPACE-~]	OXTITLE_2
8.	Name in language 3	char(128) NOT NULL	[SPACE-~]	OXTITLE_3
9.	Sorting	integer(11)	0 to 2147483647	OXPOS

8.5 Registering function in oxerpcsvexport.php

In oxerpcsvexport.php, all get functions are defined. `OXERPSetAttribute` is not an export function, nothing has to be changed.

If a get function `OXERPGetAttribute` was implemented, in the field list at the beginning of the file the following line had to be added:

```
"OXERPGetAttribute" => "sAttributeID",
```