

# LexCom standard interface description for DMS vendors

Version: 2.0.1 - September 2017



## Part I: Prologue

1	Purpose of this document.....	3
2	Executive summary .....	3
3	Company introduction .....	4
3.1	LexCom customers .....	4
3.2	LexCom products .....	4
3.2.1	LexCom EPCs.....	4
3.2.2	partslink24.....	5
4	Conceptual features .....	6
4.1	Scenarios .....	6
4.2	Competitive advantages.....	7
4.3	Operational process .....	7
4.4	Business model.....	7
4.5	Further contact.....	7
5	Technical scenario .....	8

## Part II: Protocol Specification

1	Introduction .....	3
2	Basic communication .....	3
3	DMS commands.....	4
3.1	Get available commands .....	4
3.2	Get stock information.....	5
3.3	List open orders .....	7
3.4	Get order data .....	8
3.5	Create order .....	10
3.6	Append order data .....	11
4	Version history.....	14
4.1	What's new in version 2.0? .....	14
	Appendix A Brand codes .....	15
	Appendix B Contact .....	16

## 1 Purpose of this document

This document is designed as a guideline for DMS vendors to provide a DMS connection with LexCom services. It contains conceptual as well as technical information regarding the implementation of a standard interface to LexCom communication systems.

## 2 Executive summary

Over the last decade the competition between retailers in the automotive industry has significantly increased. Therefore customised features have become crucial for the long term success of a car dealership management system. With a connection to the advanced parts information and selling systems offered by LexCom, DMS vendors make their DMS significantly more valuable.

For 40 years LexCom has provided its expertise for car manufacturers in the genuine spare parts business. As a result of this long experience LexCom is one of the three global players in “global publishing and ordering network for the automotive industry”.

With increasing frequency LexCom has been asked by clients to provide DMS vendors with the opportunity to easily interface their DMS with LexCom's specialised products and services.

Consequently this document

- gives a brief overview about the LexCom company
- illustrates the LexCom client and product portfolio
- and closes with the standard LexCom interface description.

A LexCom DMS interface generates benefits for the DMS user as well as the DMS vendor. On the one hand DMS users take advantage from improved efficiency at the dealership due to targeted information exchange and accelerated sales processes. On the other hand a DMS vendor increases the value of its DMS, sustains its position in the automotive industry and gains new business opportunities.

## 3 Company introduction

LexCom is one of the worldwide leading companies providing specialised IT system solutions for the automotive industry. Founded in 1967, the family owned company concentrated its services exclusively into the automotive after sales segment in 1977.

In addition to its headquarters in Munich, LexCom has branches in Poole (UK) Yokohama (JP), Shanghai (CN) and Sao Paolo (BR). Currently there are 300 employees working for LexCom worldwide.

### 3.1 LexCom customers

LexCom offers its services exclusively for manufacturers in the genuine spare parts business. Over recent decades the number of LexCom clients in the automotive industry has been steady rising.

Today to the LexCom portfolio comprises, amongst others:

Alfa Romeo, Audi, Bentley, BMW, BMW Motorrad, Bugatti, Chrysler, Citroën, Citroën DS, Dacia, Dodge, Fiat, Ford, Hyundai, Jeep, Kia, Lancia, MAN, Mercedes, Mercedes Trucks, Mercedes Vans, MINI, Mitsubishi, Nissan, Opel, Peugeot, Porsche, Renault, Seat, Škoda, Smart, Vauxhall, Volkswagen and Volvo.

Meanwhile more than 80,000 customers are using LexCom systems in around 150 countries.

### 3.2 LexCom products

In general all LexCom products are developed for the after sales segment and cover the complex demand in information management between OEMs, National Sales Companies, importers, authorised dealers, independent motor traders (IMTs) and end-consumers.

#### 3.2.1 LexCom EPCs

For some of the above mentioned OEMs LexCom produces the original spare part catalogue (dealer EPC) such as "ASA" for Mitsubishi, "ETKA" for the Volkswagen group, "MANTIS" for MAN or "PET" for Porsche.

### 3.2.2 partslink24

In parallel to the LexCom EPCs mentioned in 2.2.1, LexCom also provides a service named "partslink24".

partslink24 is a multi-brand online portal which gives IMTs the possibility to identify and order original parts directly from authorised dealers of meanwhile 36 brands. It was created by LexCom in 2004 on behalf of various car manufacturers.

partslink24 is a "web market place" where two kinds of user profiles can meet:

#### **Seller profile:**

- Authorised dealers, which are listed in partslink24, have been individually chosen for each brand and market by the local brand importer. Those dealers can use the partslink24 user profile free of charge\*\*.  
With partslink24 dealers offer easily customised order conditions on the internet for IMTs, gaining flexibility in their working hours 24/7 and liberating themselves from undesired phone calls or fax orders.  
partslink24 orders arrive already populated with a high quality information and can be processed more quickly. These facts lead not only to increased customer satisfaction, but also to a higher parts turnover as an IMT orders in average 5-20% more parts in partslink24 due to shopping recommendations\*, fittings parts\* or even supersession\*.

(\*depending on each catalogue)

#### **Buyer profile:**

- IMTs use partslink24 to identify original spare parts and to send online orders to authorised dealers.  
An annual partslink24 subscription cost 20€ per month, providing full access to original parts information of 36 brands.

Nowadays partslink24 is recognised as the unique "one stop shop" for genuine spare parts – without any installation, manual update or training requirements on the IMT side.

(\*\*) without pl24connect/promotion management which remains optional

## 4 Conceptual features

A connection between a DMS and a LexCom communication system allows to:

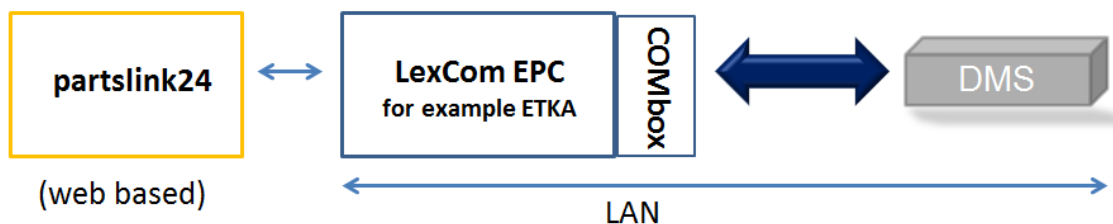
- transfer a pick list (order) from a LexCom EPC / partslink24 to the authorised dealer DMS.
- publish DMS stock information (parts inventory) and bin location in the LexCom EPC or partslink24 in real-time.
- publish DMS price in the LexCom EPC / pl24connect or customer purchase price in partslink24.

### 4.1 Scenarios

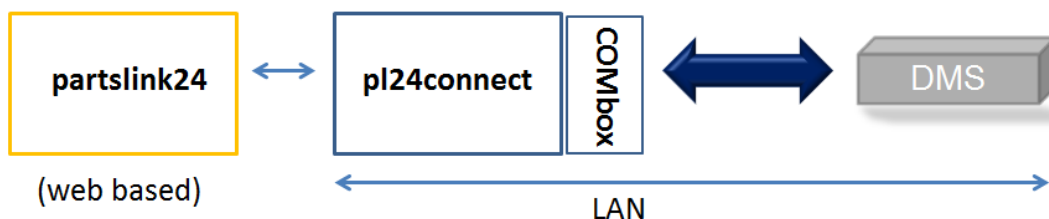
Several LexCom services already support the LexCom DMS protocol\*. There are two scenarios to connect a DMS to a LexCom product.

**Option A)** DMS installed locally at a single dealership, usually accessible only via LAN

In case of a LexCom EPC



In case of partslink24\*\*



**Option B)** DMS installed centrally at a hosting center of the OEM / importer or even DMS provider, usually accessible via internet (WebService)

For further information about “Option B” a DMS vendor should contact LexCom directly.

\* So far pl24connect, ETKA and ASA 2.1 support the LexCom DMS protocol. Other LexCom services will follow midterm.

\*\* The dealer needs to install the LexCom application pl24connect.

## 4.2 Competitive advantages

Generally a DMS interfaced with a LexCom product enables new features for the DMS and offers various advantages for the DMS user.

### Advantages for the DMS vendor

A DMS vendor gains with this interface direct connection to all LexCom communication systems (e.g. ETKA for the complete Volkswagen group or partslink24 for 38 brands) which will significantly widen its product range.

In addition the interface optimises the daily processes for the current DMS customers and makes the DMS more attractive for existing and potential customers.

### Advantages for the DMS users

The DMS user prevents amongst others double part number entry and reduces the risk of typing errors or incorrect orders. In the case of partslink24 it also stimulates parts sales for the dealer and avoids time consuming inquiries due to the stock publication.

Consequently the DMS users save time, lower the risk of miscalculation and increase customer loyalty. In summary the new DMS features raise the productivity at the dealership and give the possibility to close more business.

## 4.3 Operational process

A technical protocol is attached in the second chapter of this document. It includes all details for the DMS vendor to start with the LexCom standard interface setup.

During the implementation phase, the DMS vendor will receive additionally a test tool as well as free access to a LexCom EPC or partslink24.

Afterwards the new DMS interface will be installed at selected pilot dealerships, and can finally be offered for further clients worldwide.

## 4.4 Business model

LexCom offers its professional and technical support for DMS vendors to guarantee a quick and easy interface implementation.

This means a DMS vendor can increase business opportunities worldwide by requesting e.g. an individual fee for a successfully implemented interface to LexCom's communication systems.

Usually the DMS vendor offers, sells, administers and charges a fee for this interface directly. On the other side LexCom may charge a fee for its application.

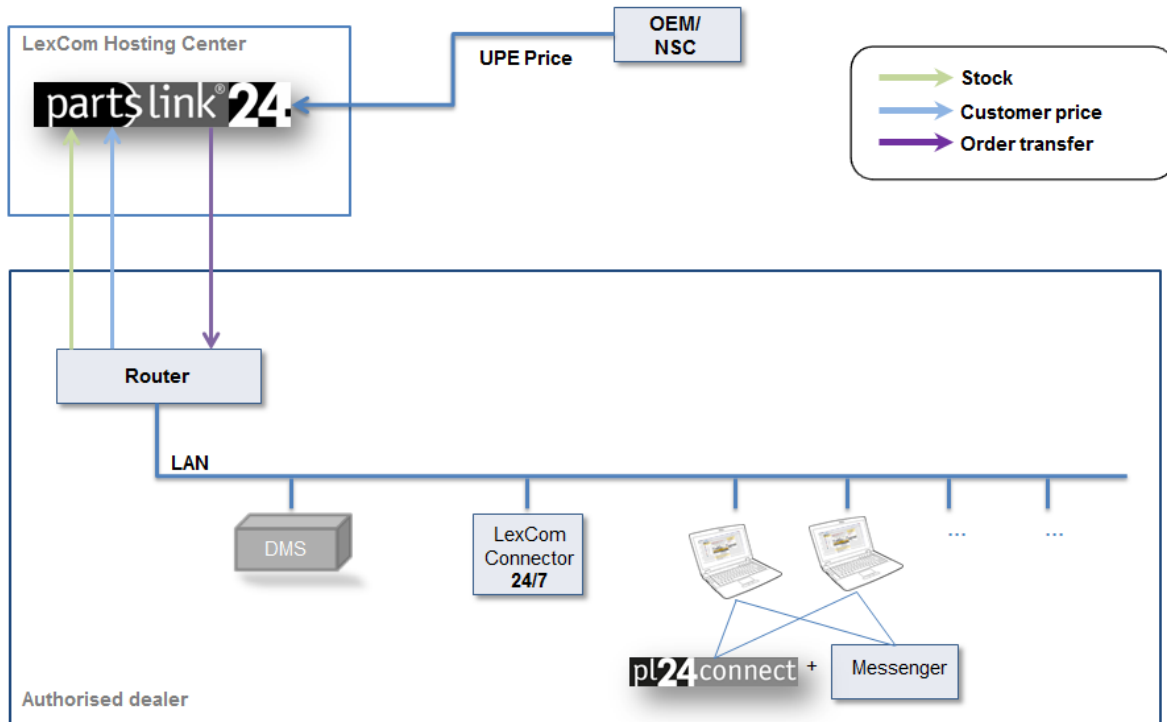
Still every business model can be discussed and adopted on the individual market demand.

The LexCom products and services are restricted to be run and administered by LexCom.

## 4.5 Further contact

For further information a DMS vendor can contact the LexCom sales representative or sends a request to the mail address "integration@lexcom.de".

## 5 Technical scenario



support tools (e.g for testing purposes) will also be available there.





LexCom  
Informationssysteme GmbH

# **DMS Protocol Specification**

**Version 2.0, 22.09.2015**

## Table of Contents

1	Introduction.....	3
2	Basic communication .....	3
3	DMS commands.....	4
3.1	Get available commands .....	4
3.2	Get stock information.....	5
3.3	List open orders.....	7
3.4	Get order data .....	8
3.5	Create order .....	10
3.6	Append order data .....	11
4	Version history.....	14
4.1	What's new in version 2.0? .....	14
Appendix A	Brand codes .....	15
Appendix B	Contact .....	16

## 1 Introduction

The main purpose of this document is to specify a standard communication protocol for selected DMS services. The protocol actually focusses on two DMS features:

- Get stock information and prices for articles
- Submit orders for articles

The protocol will be used by partslink24 and other LexCom products to automate shopping processes.

## 2 Basic communication

### HTTP-based

The communication is based on HTTP (or HTTPS if encryption is required). For all commands the same request URL is used, e.g.

```
http://dmshost:8001/dms/service
```

The concrete hostname, port, and path can be adapted as needed. The response HTTP code should always be 200. Only in case of a critical server error a HTTP code 5xx can be returned.

### XML representation

Request and response data is transmitted as XML document using UTF-8 encoding and HTTP method POST. Both, request and response should set the HTTP header:

```
Content-Type: text/xml; charset=utf-8
```

The detailed XML structure depends on the DMS command, see chapter 3. In case of a protocol error, e.g. syntactical incorrect XML request or unknown command, the server responds with an error message:

```
<?xml version="1.0" encoding="UTF-8"?>  
<error version="2.0">  
  <message>Unknown command 'stckinfo'</message>  
</error>
```

### Atomic processing

In general, a DMS command must behave atomic, i.e. if a request results in an error message, all modifications in the DMS database done so far by this command must be undone, e.g. by rolling back a database transaction.

### Communication security

If the DMS server can be accessed over internet (web service), the communication should be encrypted and password protected. For encryption the SSL/TLS protocol (HTTPS) is used and for

client authentication HTTP Basic Authentication is used. Username and password can be chosen arbitrarily.

### 3 DMS commands

The following sections specify possible DMS commands. To be conform to the DMS protocol specification a server doesn't have to implement all commands. There is only one command that is mandatory: "commands-available". All other commands are optional and can be implemented if appropriate.

#### 3.1 Get available commands

This command helps to discover the specification version and the commands that are actually implemented by the DMS.

##### Request

Name	Occ.	Data type	Description
commands-available	1		Root element
@version	0..1	String	Protocol version, can be any version.

The version attribute should be ignored by the server as it could be a newer version. The value is just an indication for the newest protocol version the client supports.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<commands-available />
```

##### Response

Name	Occ.	Data type	Description
commands-available-response	1		Root element
@version	1	String	Protocol version, value must be "2.0".
command	0..n	String	Command name

The version attribute in the response defines the specification version that is implemented by the server. The command list contains all commands that are implemented by the server. All command implementations have to comply with specification version "2.0". A mixture of specification versions is not allowed.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<commands-available-response version="2.0">
  <command>commands-available</command>
  <command>stock-information</command>
  <command>order-create</command>
  <command>order-append</command>
</commands-available-response>
```

## 3.2 Get stock information

This command queries stock and price information for a list of articles.

### Request

Name	Occ.	Data type	Description
stock-information	1		Root element
@version	1	String	Protocol version, value must be "2.0".
dealer-id	1	String	ID to identify the dealer
customer-number	1	String	Customer number in relationship with dealer
customer-contract	0..1	String	Extends the customer number for individual pricing.
brand	1	String	Brand the articles belong to. Valid values are defined in Appendix A.
item	0..n		Article to get stock information for
item-id	1	String	Client specific item identifier, must be unique per request
article-id	1	String	Article number to identify the article
stock-requested	1	Integer	Number of pieces requested, value must be positive

The name column specifies the name of the XML element. If a name has an "@" prefix, it declares an attribute of an XML element. The name indentation specifies the path depth related to the XPath representation. The column "occurrence" specifies how often an element can appear in an XML document.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<stock-information version="2.0">
  <dealer-id>Dealer1</dealer-id>
  <customer-number>12345678</customer-number>
  <customer-contract>0003</customer-contract>
  <brand>Audi</brand>
  <item>
    <item-id>1</item-id>
    <article-id>1H0512345</article-id>
    <stock-requested>3</stock-requested>
  </item>
  <item>
    <item-id>2</item-id>
    <article-id>1H0523456</article-id>
    <stock-requested>1</stock-requested>
  </item>
  <item>
    <item-id>3</item-id>
    <article-id>2G6637</article-id>
    <stock-requested>1</stock-requested>
  </item>
</stock-information>
```

### Response

Name	Occ.	Data type	Description
stock-information-response	1		Root element
@version	1	String	Protocol version, value must be "2.0".
item	0..n		Article with stock information

item-id	1	String	Client specific item identifier given in request
description	0..1	String	Article description
@lang	0..1	String	Language code according to ISO 639-1
price	0..1	Decimal	Net price per unit
@currency	1	String	Currency code according to ISO 4217
store	0..n		Stock information for a store
@id	1	String	Name of the store
@type	1	String	Location relating to dealer, possible values: local, remote
stock	1	Integer	Number of pieces available in stock, must be non-negative
storage	0..1	String	Name of the storage, e.g. rack number
shipping-on	0..1	Date	Estimated ready for shipment date, format YYYY-MM-DD or YYYY-MM-DD'T'HH:MM:SS'Z' in UTC time zone, according to ISO 8601
error	0..1	Boolean	Indicates an error, valid values are "true" and "false"

### Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<stock-information-response version="2.0">
  <item>
    <item-id>1</item-id>
    <description lang="en">Oil filter</description>
    <price currency="EUR">7.89</price>
    <store id="localStore" type="local">
      <stock>2</stock>
      <storage>Rack13</storage>
      <shipping-on>2013-03-15</shipping-on>
    </store>
    <store id="supplyCenter123" type="remote">
      <stock>120</stock>
      <shipping-on>2013-03-22</shipping-on>
    </store>
  </item>
  <item>
    <item-id>2</item-id>
    <description lang="en">Air filter</description>
    <price currency="EUR">13.00</price>
    <store id="localStore" type="local">
      <stock>1</stock>
      <storage>Rack5</storage>
      <shipping-on>2013-03-15T10:30:00Z</shipping-on>
    </store>
  </item>
  <item>
    <item-id>3</item-id>
    <error>true</error>
  </item>
</stock-information-response>
```

The response lists all requested items the DMS has information about. Unknown articles may be omitted. The article description may be in any language, the language can be annotated by attribute.

Each item can refer to several stores if appropriate. For instance if the preferred (local) store has insufficient stock, the DMS can list the remaining stock with early shipping date and additionally another store with later shipping date. Please be aware that several items can have the same "article-id". In this case the requested number of pieces should be added to determine the store(s)

necessary to serve the full request. It is recommended that all items referring to the same article get the same store information – in particular, no stock calculation should be done.

The pricing and the store information may depend on the number of units requested. If an item has no price information, the price tag must be omitted. It is not a good idea to use “0.00” as missing price indication as it will be interpreted as “for free” and might be shown e.g. in partslink24.

In case of an error the DMS can indicate by an “error” tag that it has some data about the article but could not determine it.

### 3.3 List open orders

This command lists orders in the DMS that are not closed.

#### Request

Name	Occ.	Data type	Description
order-list	1		Root element
@version	1	String	Protocol version, value must be “2.0”.
dealer-id	1	String	ID to identify the dealer
customer-number	0..1	String	Customer number in relationship with dealer. If set only orders for this customer are listed.
brand	1	String	Brand the orders have to refer to. Valid values are defined in Appendix A.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-list version="2.0">
  <dealer-id>Dealer1</dealer-id>
  <customer-number>12345678</customer-number>
  <brand>Audi</brand>
</order-list>
```

#### Response

Name	Occ.	Data type	Description
order-list-response	1		Root element
@version	1	String	Protocol version, value must be “2.0”.
order	0..n		Order information
order-id	1	String	Unique order ID
creation-date	1	Date	Creation date of the order entry, format YYYY-MM-DD
editable	1	Boolean	Can the order be modified? Valid values are “true” and “false”.
status	0..1	String	DMS specific order status code
brand	1	String	Brand the order refers to. Valid values are defined in Appendix A.
customer-number	1	String	Customer number in relationship with dealer
customer-contract	0..1	String	Extends the customer number for individual pricing.
customer-ref	1	String	Customer order description
customer-name	1	String	Customer name
billing-address	0..1		Customer billing address
address-line	0..n	String	Address line
delivery-address	0..1		Customer delivery address

address-line	0..n	String	Address line
vin	0..1	String	Vehicle Identification Number, may be set if the order refers to only one vehicle
license-plate	0..1	String	Vehicle license plate information, may be set if the order refers to only one vehicle
order-type	0..1	String	DMS specific order type code

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-list-response version="2.0">
  <order>
    <order-id>1234</order-id>
    <creation-date>2013-03-02</creation-date>
    <editable>true</editable>
    <status>open</status>
    <brand>Audi</brand>
    <customer-number>12345678</customer-number>
    <customer-contract>0003</customer-contract>
    <customer-ref>Order #003 for Audi</customer-ref>
    <customer-name>Autocenter Meier</customer-name>
    <billing-address>
      <address-line>Alleeweg 1</address-line>
      <address-line>80331</address-line>
      <address-line>Munich</address-line>
      <address-line>Bavaria</address-line>
      <address-line>Germany</address-line>
    </billing-address>
    <delivery-address>
      <address-line>Alleeweg 1</address-line>
      <address-line>80331</address-line>
      <address-line>Munich</address-line>
      <address-line>Bavaria</address-line>
      <address-line>Germany</address-line>
    </delivery-address>
    <vin>ABCDE26Y213112345</vin>
    <license-plate>M-LC 100</license-plate>
    <order-type>Warranty</order-type>
  </order>
</order-list-response>
```

### 3.4 Get order data

This command returns the complete order data for a single order.

#### Request

Name	Occ.	Data type	Description
order-get	1		Root element
@version	1	String	Protocol version, value must be "2.0".
order-id	1	String	Unique order ID

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-get version="2.0">
```



```
<order-id>1234</order-id>
</order-get>
```

## Response

Name	Occ.	Data type	Description
order-get-response	1		Root element
@version	1	String	Protocol version, value must be "2.0".
order-id	1	String	Unique order ID
creation-date	1	Date	Creation date of the order entry, format YYYY-MM-DD
editable	1	Boolean	Can the order be modified? Valid values are "true" and "false".
status	0..1	String	DMS specific order status code
brand	1	String	Brand the order refers to. Valid values are defined in Appendix A.
customer-number	1	String	Customer number in relationship with dealer
customer-contract	0..1	String	Extends the customer number for individual pricing.
customer-ref	1	String	Customer order description
customer-name	1	String	Customer name
billing-address	0..1		Customer billing address
address-line	0..n	String	Address line
delivery-address	0..1		Customer delivery address
address-line	0..n	String	Address line
vin	0..1	String	Vehicle Identification Number, may be set if the order refers to only one vehicle
license-plate	0..1	String	Vehicle license plate information, may be set if the order refers to only one vehicle
order-type	0..1	String	DMS specific order type code
comment	0..1	String	Additional order advice from customer
item	0..n		Order items
item-id	1	String	Client specific item identifier, must be unique per order
article-id	1	String	Article number to identify the article
description	0..1	String	Article description
@lang	0..1	String	Language code according to ISO 639-1
units	1	Integer	Number of pieces, must be positive
price	0..1	Decimal	Net price per unit
@currency	1	String	Currency code according to ISO 4217
tax-rate	0..1	Decimal	Tax rate, e.g. 0.19 for 19 %
category	0..1	String	Article category, DMS specific
discount-group	0..1	String	Article discount group, DMS specific
vin	0..1	String	Vehicle Identification Number, only relevant for vehicle intelligent parts like keys and embedded systems
comment	0..1	String	Additional order advice from customer for order item

## Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-get-response version="2.0">
  <order-id>1234</order-id>
  <creation-date>2013-03-02</creation-date>
  <editable>true</editable>
  <status>open</status>
  <brand>Audi</brand>
  <customer-number>12345678</customer-number>
  <customer-contract>0003</customer-contract>
  <customer-ref>Order #003 for Audi</customer-ref>
  <customer-name>Autocenter Meier</customer-name>
```

```

<billing-address>
  <address-line>Alleeweg 1</address-line>
  <address-line>80331</address-line>
  <address-line>Munich</address-line>
  <address-line>Bavaria</address-line>
  <address-line>Germany</address-line>
</billing-address>
<delivery-address>
  <address-line>Alleeweg 1</address-line>
  <address-line>80331</address-line>
  <address-line>Munich</address-line>
  <address-line>Bavaria</address-line>
  <address-line>Germany</address-line>
</delivery-address>
<vin>ABCDE26Y213112345</vin>
<license-plate>M-LC 100</license-plate>
<order-type>Warranty</order-type>
<comment>Shipping address has changed</comment>
<item>
  <item-id>1</item-id>
  <article-id>1H0512345</article-id>
  <description lang="en">Oil filter</description>
  <units>3</units>
  <price currency="EUR">7.89</price>
  <tax-rate>0.19</tax-rate>
  <category>CA</category>
  <discount-group>08</discount-group>
  <vin>ABCDE26Y213112345</vin>
  <comment>green color</comment>
</item>
<item>
  <item-id>2</item-id>
  <article-id>1H0523456</article-id>
  <description lang="en">Air filter</description>
  <units>1</units>
  <price currency="EUR">13.00</price>
</item>
</order-get-response>

```

### 3.5 Create order

This command creates a new order header in the DMS and returns the order ID. To append order items see section 3.6.

#### Request

Name	Occ.	Data type	Description
order-create	1		Root element
@version	1	String	Protocol version, value must be "2.0".
dealer-id	1	String	ID to identify the dealer
customer-number	1	String	Customer number in relationship with dealer
customer-contract	0..1	String	Extends the customer number for individual pricing
customer-name	1	String	Customer name
brand	1	String	Brand the order refers to. Valid values are defined in Appendix A.
vin	0..1	String	Vehicle Identification Number, may be set if the order refers to only one vehicle
license-plate	0..1	String	Vehicle license plate information, may be set if the

Name	Occ.	Data type	Description
order-type	0..1	String	order refers to only one vehicle DMS specific order type code

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-create version="2.0">
  <dealer-id>Dealer1</dealer-id>
  <customer-number>12345678</customer-number>
  <customer-contract>0003</customer-contract>
  <customer-name>Autocenter Meier</customer-name>
  <brand>Audi</brand>
  <vin>ABCDE26Y213112345</vin>
  <license-plate>M-LC 100</license-plate>
  <order-type>Warranty</order-type>
</order-create>
```

Response

Name	Occ.	Data type	Description
order-create-response	1		Root element
@version	1	String	Protocol version, value must be "2.0".
order-accepted	1	Boolean	"true" if the order was accepted by the DMS, "false" in case of an error
order-id	0..1	String	New order ID, has to be set in case of success
message	0..1	String	Feedback message
@level	1	String	Severity of the feedback, valid values are "Info", "Warning" and "Error"

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-create-response version="2.0">
  <order-accepted>true</order-accepted>
  <order-id>1234</order-id>
</order-create-response>
```

In case of an error the response contains an error message instead of the order ID.

```
<?xml version="1.0" encoding="UTF-8"?>
<order-create-response version="2.0">
  <order-accepted>false</order-accepted>
  <message level="Error">Unknown customer '123'</message>
</order-create-response>
```

### 3.6 Append order data

This command updates order details and appends order items to an existing open order.

Request

Name	Occ.	Data type	Description
order-append	1		Root element
@version	1	String	Protocol version, value must be "2.0".
order-id	1	String	Unique order ID
customer-ref	0..1	String	Customer order description

comment	0..1	String	Additional order advice from customer
billing-address	0..1		Customer billing address
gender	0..1	String	Gender, valid values are "male" and "female"
title	0..1	String	Title
first-name	0..1	String	First name
last-name	0..1	String	Last name
company	0..1	String	Company name
department	0..1	String	Department
position	0..1	String	Position
address1	0..1	String	Address line 1, e.g. street and street number
address2	0..1	String	Address line 2
po-box	0..1	String	Post-office box
postal-code	0..1	String	Postal code
city	0..1	String	City
province	0..1	String	Province / administrative district / county, different formats allowed, e.g. Bayern, Bavaria, BY, DE-BY
country	0..1	String	Country code according to ISO 3166-1 alpha-2
email	0..1	String	E-mail
phone	0..1	String	Telephone
fax	0..1	String	Fax
delivery-address	0..1		Customer delivery address
gender	0..1	String	Gender, valid values are "male" and "female"
title	0..1	String	Title
first-name	0..1	String	First name
last-name	0..1	String	Last name
company	0..1	String	Company name
department	0..1	String	Department
position	0..1	String	Position
address1	0..1	String	Address line 1, e.g. street and street number
address2	0..1	String	Address line 2
po-box	0..1	String	Post-office box
postal-code	0..1	String	Postal code
city	0..1	String	City
province	0..1	String	Province / administrative district / county, different formats allowed, e.g. Bayern, Bavaria, BY, DE-BY
country	0..1	String	Country code according to ISO 3166-1 alpha-2
email	0..1	String	E-mail
phone	0..1	String	Telephone
fax	0..1	String	Fax
item	0..n		Order items
item-id	1	String	Client specific item identifier, must be unique per request
article-id	1	String	Article number to identify the article
description	0..1	String	Article description
@lang	0..1	String	Language code according to ISO 639-1
units	1	Integer	Number of pieces, must be positive
price	0..1	Decimal	Net price per unit
@currency	1	String	Currency code according to ISO 4217
tax-rate	0..1	Decimal	Tax rate, e.g. 0.19 for 19 %
category	0..1	String	Article category, DMS specific
discount-group	0..1	String	Article discount group, DMS specific
vin	0..1	String	Vehicle Identification Number, only relevant for vehicle intelligent parts like keys and embedded systems
comment	0..1	String	Additional order advice from customer for order item

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-append version="2.0">
  <order-id>1234</order-id>
  <customer-ref>Order #003 for Audi</customer-ref>
  <comment>Shipping address has changed</comment>
  <billing-address>
    <address1>Alleeweg 1</address1>
    <postal-code>80331</postal-code>
    <city>Munich</city>
    <province>Bavaria</province>
    <country>DE</country>
  </billing-address>
  <delivery-address>
    <address1>Alleeweg 1</address1>
    <postal-code>80331</postal-code>
    <city>Munich</city>
    <province>Bavaria</province>
    <country>DE</country>
  </delivery-address>
  <item>
    <item-id>1</item-id>
    <article-id>1H0512345</article-id>
    <description lang="en">Oil filter</description>
    <units>3</units>
    <price currency="EUR">7.89</price>
    <tax-rate>0.19</tax-rate>
    <category>CA</category>
    <discount-group>08</discount-group>
    <vin>ABCDE26Y213112345</vin>
    <comment>green color</comment>
  </item>
  <item>
    <item-id>2</item-id>
    <article-id>1H0523456</article-id>
    <description lang="en">Air filter</description>
    <units>1</units>
    <price currency="EUR">13.00</price>
  </item>
</order-append>
```

If more than one command order-append is sent for the same order-id, header data like customer-ref or delivery-address should be overwritten, whereas the items should be accumulated. (In case of a double submission of the same order-append command, this would lead to the number of units of each item to be doubled.) Currently, there is no possibility to update or delete an item in an order.

## Response

Name	Occ.	Data type	Description
order-append-response	1		Root element
@version	1	String	Protocol version, value must be "2.0".
order-accepted	1	Boolean	"true" if the order was accepted by the DMS, "false" in case of an error
item	0..n		Order items with feedback
item-id	1	String	Client specific item identifier given in request
message	1	String	Feedback message
@level	1	String	Severity of the feedback, valid values are "Info",

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-append-response version="2.0">
  <order-accepted>true</order-accepted>
</order-append-response>
```

In case of an error or if the DMS wants to give more feedback on some order items the response looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<order-append-response version="2.0">
  <order-accepted>false</order-accepted>
  <item>
    <item-id>1</item-id>
    <message level="Warning">Unknown VIN ABCDE26Y213112345</message>
  </item>
  <item>
    <item-id>2</item-id>
    <message level="Error">Unknown article number</message>
  </item>
</order-append-response>
```

## 4 Version history

### 4.1 What's new in version 2.0?

- more detailed communication security section
- new command "commands-available"
- support for billing and delivery address
- estimated ready for shipment date with optional time

## Appendix A Brand codes

The following list defines brand codes to be used by DMS commands.

A.M.G.	Hummer	Opel
Abarth	Hyundai	P.G.O.
Acura	Infiniti	Peugeot
AlfaRomeo	Isuzu	Plymouth
Alpina-BMW	Iveco	Pontiac
AstonMartin	IvecoUnic	Porsche
Audi	Jaguar	Proton
Autobianchi	Jeep	Renault
Bentley	Kia	RenaultV.I.
BMW	LADA	Rolls-Royce
Bugatti	Lamborghini	Rover
Buick	Lancia	Saab
Cadillac	Landrover	Saturn
Chevrolet	Landwind	Scania
Chrysler	Lexus	Seat
Citroen	Lincoln	Skoda
CitroenDS	Lotus	Smart
Dacia	MAN	SsangYong
Daewoo	Maserati	Subaru
DAF	Maybach	Suzuki
Daihatsu	Mazda	Tata
Daimler	Mercedes-Benz	Toyota
DeTomaso	Mercury	Triumph
Dodge	MG	Vauxhall
Ferrari	MINI	Volkswagen
Fiat	Mitsubishi	Volvo
Ford	Morgan	VolvoTrucks
Gruau	Nissan	VWTrucksAndBuses
Holden	NSU	VWCommercial
Honda	Oldsmobile	

## **Appendix B      Contact**

LexCom Informationssysteme GmbH  
Rüdesheimer Str. 23  
D-80686 Munich  
Germany

Phone: ++49 (0) 89 54715-0  
Fax:    ++49 (0) 89 54718-175  
Web:    [www.lexcom.de](http://www.lexcom.de)  
Email: [integration@lexcom.de](mailto:integration@lexcom.de)