

---

Interface Description

# Order Handling

for Bartrack 1.1



**Copyright Prevas AB 2004**

This document was produced by Prevas AB  
KS001B07/EN V2

**Prevas AB**  
Box 1909  
S-651 19 Karlstad

Phone: +46 54 147400  
Fax: +46 54 147499





# Contents

<b>Introduction</b>	<b>7</b>
Importing Orders .....	7
Exporting Order Information .....	7
Order Information .....	7
Order Handling .....	8
<b>Files and Directories</b>	<b>9</b>
File Handling .....	9
Import Process .....	9
File Delivery .....	10
Error Logging .....	10
<b>Message Contents</b>	<b>11</b>
Assumptions .....	11
Message Structure .....	11
/-record .....	13
A-record .....	14
T-record (1 and 2) .....	14
T-record (3) .....	15
T-record (4 and 5) .....	15
B-record .....	17
C-Record .....	18
<b>Glossary of Terms</b>	<b>19</b>



# Introduction

---

## Importing Orders

In Bartrack it is possible to attach an order number to an individual or individuals. The order needs to be prepared first, and some basic information about the order entered.

However, it is not possible to use the application itself to enter this information. The only way of doing this is by using this order import interface.

The purpose of the interface is to allow order systems, such as SAP/R3, to send order information to Bartrack. This will ensure automatic and correct order information in Bartrack. Normally, no human interaction is necessary except when the order is started by an operator.

---

## Exporting Order Information

It is possible to export order information in connection to the ship activity in Bartrack. The information that is extracted is shown later in the document. Basically you will receive the same information that was imported but it will be connected to an individual serial number.

Normally this feature will be disabled, to invoke it you need to set a parameter in the BARTRACK\_SETUP table in the database.

---

## Order Information

Along with the top parent of the order, some extra order information can be passed on. For example, configuration data or license information can be sent along with the order and stored in Bartrack. When an external system needs this information, it uses the SFT interface or the BarAPIx interface to retrieve it.

For more information on Bartrack interfaces, please refer to KS001b04/en “Interface Description - SFT for Bartrack” and KS001b05/en “Interface Description – BarAPIx”

For each top parent it is possible to define a number of order rows that will be associated with this parent. In effect, this means that a one-level product structure can be transferred from the order system to Bartrack.

The child level specified by the order can in its turn consist of many levels already prepared in Bartrack. This allows for easy configuration of the top-levels of a structure, while the lower parts are already put together in standard modules.

When the individuals produced according to a specific order is shipped, it is optional if the structure should be verified or not against the original order product structure.

## Order Handling

Each order being received from the order system is considered to have a unique order number. At least for the duration of the production of the order, and any historic storage after that point.

To allow for qualification of the order number, it is possible to specify an Item number. This number is used in conjunction with the order number to form a unique key for each order. If the item number is not used, you must supply the value -1 to indicate that so is the case.

When an order is received, Bartrack tries to store it in the database. Should there already be an order with the same unique key, it will be discarded, and the new information will be stored.

It is also possible to delete an order. Just supply the order number and item number in the delete record and the complete order will be deleted, including its product structure (children). There is only one exception; if the order has already been started in Bartrack, it will not be possible to delete it unless the orders is released from the individual associated with it.

To ensure that the order interface is working at all times, there are very few error situations. Most of these include moving the offending order row to an error file for safe-keeping until someone has the time to correct the problem.

Bartrack only supports one top individual per order.

# Files and Directories

---

## File Handling

The import interface is based on files being delivered to an in-directory. The filename must end with IN (**\*.IN**) and is a plain text file containing the specified records of this interface description.

### Import Process

The service Order Receive is responsible for polling the directory specified in the system variable `BAR_ORD_IN_DIR` for order files and interpret the records in the file.

During the interpretation, files containing faulty records will be put in an error directory (`BAR_ORD_ERROR_DIR`), and files containing only successful records will be put in a save directory(`BAR_ORD_SAVE_DIR`).

System variable	Comment
<code>BAR_ORD_IN_DIR</code>	The location of all incoming order files.
<code>BAR_ORD_SAVE_DIR</code>	Completely successful files.
<code>BAR_ORD_ERROR_DIR</code>	Failing files.

**N.B.** These three directories must be placed on the same physical disk. Moving the files is accomplished by a **RENAME** command, that only works on the same physical disk.

A system administrator might get some errors in Bartrack's alarm log. When the cause of the failure is corrected, the failed files can be tried again.

### Example

An order is received containing references to non-existing products in Bartrack. An error will be logged in the alarm log indicating the cause of the failure. The failing file will be put in the `BAR_ORD_ERROR` directory.

A preparer adds the missing products to Bartrack, and the system administrator just moves failed file back to the in-directory `BAR_ORD_IN_DIR`.

The next time the Order Receive service executes, it will read the same records one more time, but this time they will be successful. The successful file will then be moved to the `BAR_ORD_SAVE_DIR` directory. If any remaining failing records exists, the whole file will be moved to the `BAR_ORD_ERROR_DIR` directory again.

## File Delivery

There are many different ways to put a file in the **BAR\_ORD\_IN\_DIR** directory. You can use FTP, EDI or mail.

## Error Logging

There are normally only two occasions when an error is logged:

1. A product in the order header or order row does not exist in Bartrack.
2. Deleting of an order that is started in Bartrack.

# Message Contents

---

## Assumptions

All values are supposed to be left justified. If not, the contents of the field will be laid out as entered.

If a field contains a value that is shorter than the field, it must be filled with blanks to the length demanded by the record format. These blanks will not be present in Bartrack. Blanks inside the field will be kept.

Any blank rows will be skipped.

More than one record may be present in one message.

When an order is added to Bartrack, it is considered to be a collection of one A-record and zero or more B-records having the same Order number and Item.

All records must end with a Carriage Return (CR) and a Line Feed (LF). When a record is interpreted, the interpretation will stop when the CR/LF is encountered. The CR/LF does not have to be put at the end of the defined record; it can be put in an arbitrary position, as long as it doesn't interfere with the record format.

If there is extra data after the final position of a record, this data will not be processed.

The maximum length of a record is 512 characters (including the CR/LF at the end of the record).

If an order is sent two times, the last version will overwrite the previous version. No log or error will be generated.

A deletion of an order requires the key to be complete (order number and Item). If not, the deletion will not take place and no log or error will be generated.

Faulty records or duplicate records will be stored in the error directory. The good records will be stored in the save directory. A file containing both good and bad records will be split up with good records in the save directory etc.

---

## Message Structure

Each file sent to Bartrack may contain any number of records, as long as the sending system, transmission and receiving system limits are considered.

The order of AA-records and BA-records is important. An AA-record may be sent by itself, without B-records. A BA-record may not be sent by itself, without a preceding A-record.

A T(n)-record must be preceded by an AA-record.

Missing T(n)-record is equal to an empty T(n)-record.

An AD-record may be sent anytime. If there is nothing to delete, it will not be logged, and it will not generate an error.

Any number of empty rows or /-records may be sent.

<b>AD-record, order x</b>	
<b>AA-record, order 1</b>	
<b>T1-record, order 1</b>	
<b>T2-record, order 1</b>	
<b>T3-record, order 1</b>	
<b>T4-record, order 1</b>	
<b>T5-record, order 1</b>	
<b>BA-records for order 1</b>	
<b>AA-record, order 2</b>	
<b>T1-record, order 2</b>	
<b>T3-record, order 2</b>	
<b>T5-record, order 2</b>	
<b>BA-records for order 2</b>	
<b>AA-record, order n</b>	
<b>BA-records for order n</b>	
<b>AD-record, order y</b>	

If something is wrong with the details of an order, the whole order is cancelled, and the next record will be processed. An order is the collection of one AA-record and its optional BA-records.

---

## **/-record**

**Name** /-record

**Description** This record is a comment record. Everything from the record start to the next CR+LF is considered a comment.

**Length** Variable, maximum 512 bytes

**Record subtype** None

**Error handling**

**Miscellaneous**

<b>Parameter</b>	<b>Type</b>	<b>Size</b>	<b>Mandatory</b>	<b>Comment</b>
Record type	Char	1	Y	First position is always /.
Comment	Char	X		Any content except CR and LF.
Record end	Char	2	Y	Always Carriage Return (CR) and Line Feed (LF) characters.

---

## A-record

<b>Name</b>	A-record
<b>Description</b>	This record contains the overall information about an order. It is a header row for the following order structure or order rows.
<b>Length</b>	Fixed, 77 bytes
<b>Record subtype</b>	A: Add D: Delete
<b>Error handling</b>	<p>If the product number and/or R-state does not exist in Bartrack, the row is considered failed and an error will be logged in Batracks alarm log.</p> <p>If an add-record tries to add an already existing order in Bartrack, the old information will be discarded and the new information will be stored and the row is considered successful.</p> <p>If a delete-record tries to delete a non-existing order in Bartrack, the row is considered successful.</p> <p>All failing records will be stored in the <b>BAR_ORD_ERROR_DIR</b> directory.</p> <p>All successful records will be stored in the <b>BAR_ORD_SAVE_DIR</b> directory.</p>
<b>Miscellaneous</b>	If there is more than one individual per order you will create them by stepping the order item number ( 1-100).

Parameter	Type	Size	Mandatory	Comment
Record type	Char	1	Y	Always A.
Record action	Char	1	Y	A for Add D for Delete
Order number	Char	35	Y	Key. The order number.
Item	Char	4	Y	Key. The order number item. <b>N.B.</b> If item is not used, the value -1 is used instead.
Product number	Char	24		The product number for the order.
R-state	Char	7		The R-state for the order.
Quantity	Char	3		Denotes the number of individuals per order (max 100).
Record end	Char	2	Y	Always Carriage Return (CR) and Line Feed (LF) characters.

---

## T-record (1 and 2)

**Name** T-record  
**Description** This record contains the “Text1” or “Text2” information about an order.  
**Length** Fixed, 39 bytes  
**Record subtype** 1: Text1  
                   2: Text2  
**Error handling**  
**Miscellaneous**

Parameter	Type	Size	Mandatory	Comment
Record type	Char	1	Y	Always T.
Record action	Char	1	Y	1 for Text1 2 for Text2
Text	Char	35	Y	Site-specific data.
Record end	Char	2	Y	Always Carriage Return (CR) and Line Feed (LF) characters.

---

## T-record (3)

**Name** T-record  
**Description** This record contains the “Text3” information about an order.  
**Length** Fixed, 260 bytes  
**Record subtype** 3: Text3  
**Error handling**  
**Miscellaneous**

Parameter	Type	Size	Mandatory	Comment
Record type	Char	1	Y	Always T.
Record action	Char	1	Y	3 for Text3
Text	Char	256	Y	Site-specific data.
Record end	Char	2	Y	Always Carriage Return (CR) and Line Feed (LF) characters.

---

## T-record (4 and 5)

**Name** T-record

**Description** This record contains the “Text4” or “Text5” information about an order.

**Length** Variable

**Record subtype** 4: Text4  
5: Text5

**Error handling**

**Miscellaneous**

Parameter	Type	Size	Mandatory	Comment
Record type	Char	1	Y	Always T.
Record action	Char	1	Y	4 for Text4 5 for Text5
Text	Char	Var	Y	Site-specific data. Texts are delimited by “;” Example 1: A-record quantity = 1  T4AAA;  -----Text4 for order 1 is set to AAA  Example 2: A-record quantity = 3  T4AAA;BBB;CCC;  -----Text4 for order 1 is set to AAA -----Text4 for order 2 is set to BBB -----Text4 for order 3 is set to CCC
Record end	Char	2	Y	Always Carriage Return (CR) and Line Feed (LF) characters.

---

## B-record

<b>Name</b>	B-record
<b>Description</b>	This record contains the order row details for an order. It is a repeated row for each of the order rows.
<b>Length</b>	Fixed, 88 bytes
<b>Record subtype</b>	A: Add
<b>Error handling</b>	<p>If the product number and/or R-state does not exist in Bartrack, the row is considered failed and an error will be logged in Batracks alarm log.</p> <p>If an add-record tries to add an already existing position in Bartrack, the old information will be discarded and the new information will be stored and the row is considered successful.</p> <p>All failing records will be stored in the <b>BAR_ORD_ERROR_DIR</b> directory.</p> <p>All successful records will be stored in the <b>BAR_ORD_SAVE_DIR</b> directory.</p>
<b>Miscellaneous</b>	The information contained in B-records will be deleted along with the information in the associated A-record if an AD-record is received (order delete).

Parameter	Type	Size	Mandatory	Comment
Record type	Char	1	Y	Always B.
Record action	Char	1	Y	Always A for Add.
Order number	Char	35	Y	Key. The order number.
Item	Char	4	Y	Key. The order number item. <b>N.B.</b> If item is not used, the value -1 is used instead.
Position	Char	10	Y	Key. The position in the assembly/structure.
Product number	Char	24	Y	Key. The product number of a child.
R-state	Char	8	Y	Key. The R-state of a child.
Quantity	Char	3	Y	The quantity of children for a position.
Record end	Char	2	Y	Always Carriage Return (CR) and Line Feed (LF) characters.

---

## C-Record

<b>Name</b>	C-record
<b>Description</b>	This record contains the order row details for an order connected to a serial number.
<b>Length</b>	Fixed, 487 bytes
<b>Record subtype</b>	None
<b>Error handling</b>	None
<b>Miscellaneous</b>	The C-record will be saved as BAR_SAP_OUT_DIR:AAAAAXXXX.OUT. (AAAAAA equals the order number, XXXX equals the item number)

Parameter	Type	Size	Comment
Record type	Char	1	Always C.
Order number	Char	35	Key. The order number.
Item	Char	4	Key. The order number item. <b>N.B.</b> If item is not used, the value -1 must be used.
Serial Number	Char	20	Key. The individual serial number.
Product number	Char	24	Key. The product number of a child.
R-state	Char	7	Key. The R-state of a child.
Text 1	Char	35	Site-specific data.
Text 2	Char	35	Site-specific data.
Text 3	Char	256	Site-specific data.
Text 4	Char	35	Site-specific data.
Text 5	Char	35	Site-specific data.
Filler	Char	5	

# Glossary of Terms

<b>Order item</b>	A qualifier to the order number. Item and Order number together is considered to be unique for a substantial time period.
<b>Order number</b>	A string of digits and/or characters that may be used to identify an order.
<b>Position</b>	The position of a unit in PRIM.
<b>PRIM</b>	Ericsson's product database.
<b>Product number</b>	A string of characters and digits identifying a product.
<b>R-state</b>	A revision or version of a product