
Interface Description

SFT for Bartrack 6.6

(Shop Floor Transaction)



© **Copyright Prevas AB 2006**

This description is produced by Prevas AB
KS001b03/en v12

Prevas AB
Box 1909
S-651 19 Karlstad

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

Contents

Introduction	8
About This Interface Description	8
Which Version.....	8
The Intended Audience.....	8
Scope of the Manual.....	8
Purpose of the Interface Description	9
Conventions.....	9
About Bartrack.....	9
What Does It Do?	9
System Overview.....	10
SFT Overview	11
Purpose of SFT	11
General Description.....	11
Usage	11
Message Sequences.....	12
Message Sequences	17
Transaction Overview.....	17
Report New Individual (NewInfo/NewInfo_1)	18
Report all Children for a Parent (AsmInfo/AsmInfo_1).....	18
Report Scrapped Individual (SepInfo).....	19
Report Updated Individual (UpdInfo)	19
Request Information about an Individual (IDReq/ IDReq_1/IDReq_2).....	19
Standard Sequence, Transaction SERDSP	20
Standard Sequence, Transaction ADDNOS	20
Error Sequence 1	20
Error Sequence 2	20
Request Structure Information about an Individual (IDStrReq/IDStrReq_1/IDStrReq_2).....	21
Standard Sequence.....	21
Error Sequence 1	21
Error Sequence 2	22
Request to Add or Remove a Child (AsmIDReq).....	22
Standard Sequence, Transaction SERPCA.....	22
Standard Sequence, Transaction SERPCD.....	22
Error Sequence 1	23
Error Sequence 2	23
Request to Store, Ship or Scrap an Individual (MoveIDReq)	23
Standard Sequence.....	24
Error Sequence 1	24
Error Sequence 2	24
Request to Ship (forced) an Individual (FShipIdReq)	24
Standard Sequence.....	25
Error Sequence 1	25
Error Sequence 2	25
Request to Reactivate an Individual (ReactIDReq).....	25
Standard Sequence.....	26

Error Sequence 1	26
Error Sequence 2	26
Request to Create Individuals (NewIDReq/NewIDReq_1/NewIDReq_2).....	26
Standard Sequence	27
Error Sequence 1	27
Error Sequence 2	27
Request to Update an Individual (UpdIDReq/UpdIDReq_1/UpdIDReq_2/UpdIDReq_3)27	
Standard Sequence 1, Always Reply.....	28
Standard Sequence 2, Never Reply	28
Error Sequence 1	28
Error Sequence 2	29
Request to Check completeness for an Individual structure (IDComplReq).....	29
Standard Sequence	29
Error Sequence 1	29
Error Sequence 2	30
Request to Login (LoginReq).....	30
Request for Structure Information of a Product.....	30
Standard Sequence	30
Error Sequence 1	31
Error Sequence 2	31
Request Information about an Order (OrdReq/OrdReq_1)	31
Standard Sequence	32
Error Sequence 1	32
Error Sequence 2	32
Request Test Status Information about an Individual (TstReq).....	33
Standard Sequence	33
Error Sequence 1	33
Error Sequence 2	33
Request Print-out of Labels (PrtLabReq).....	34
Standard Sequence	34
Error Sequence 1	34
Error Sequence 2	34
Request for Test Production Comment Information.....	35
Standard Sequence	35
Error Sequence 1	35
Error Sequence 2	35
Request for flow control on test station.....	36
Standard Sequence	36
Error Sequence 1	36
Error Sequence 2	36
Request individual data from the traceability system	37
Standard Sequence	37
Error Sequence 1	37
Error Sequence 2	37
Request data for additional numbers (defined but not generated)	38
Standard Sequence	38
Error Sequence 1	38
Error Sequence 2	38
Request to generate an additional number.....	39
Standard Sequence	39
Error Sequence 1	39
Error Sequence 2	39

Messages 40

About SFT Messages.....	40
Message contents.....	41
AsmIDAck.....	42
AsmIDErr	43
AsmIDReq.....	44

AsmInfo	45
AsmInfo_1	46
BarUnknown	47
ChkFlowErr	48
ChkFlowInfo	49
ChkFlowReq	50
FromTcyAck	51
FromTcyErr	52
FromTcyReq	53
FShipIdAck	54
FShipIdErr	55
FShipIdReq	56
GenAddAck	57
GenAddErr	58
GenAddReq	59
GetAddInfo	60
GetAddErr	61
GetAddReq	62
IDComplAck	63
IDComplErr	64
IDComplReq	65
IDErr	66
IDInfo1	67
IDInfo2	68
IDReq	69
IDErr_1	70
IDInfo_1	71
IDReq_1	73
IDErr_2	74
IDInfo_2	75
IDReq_2	77
IDStrErr_1	78
IDStrInfo_1	79
IDStrReq_1	81
IDStrErr_2	82
IDStrInfo_2	83
IDStrReq_2	85
LoginReply	86
LoginReq	87
MoveIDAck	88
MoveIDErr	89
MoveIDReq	90
MsgToFile	91
NewIDErr	92
NewIDInfo	93
NewIDReq	94
NewIDErr_1	96
NewIDInfo_1	97
NewIDReq_1	98
NewIDErr_2	100
NewIDInfo_2	101
NewIDReq_2	102
NewInfo	104
NewInfo_1	105
OrdErr	106
OrdInfo	107
OrdReq	109
OrdErr_1	110
OrdInfo_1	111

OrdReq_1	113
ProStrInfo	114
ProStrErr.....	115
ProStrReq	116
PrtLabAck.....	117
PrtLabErr	118
PrtLabReq.....	119
ReactIDAck	120
ReactIDErr.....	121
ReactIDReq	122
ScpInfo	123
TpcErr.....	124
TpcInfo	125
TpcReq	126
TstInfo	127
TstErr.....	128
TstReq.....	129
UpdIDAck	130
UpdIDErr.....	131
UpdIDReq.....	132
UpdIDAck_1	133
UpdIDErr_1	134
UpdIDReq_1.....	135
UpdIDAck_2	136
UpdIDErr_2.....	137
UpdIDReq_2.....	138
UpdIDAck_3	140
UpdIDErr_3.....	141
UpdIDReq_3.....	142
UpdInfo.....	144
Types	147
Carrier	149
Overview	149
Set-up of Buses and Groups	149
Create Queues.....	149
Receiving of Messages	149
Using CLS	150
References	151
Glossary of Terms	152
Lists	153
List of Messages in Alphabetical Order	153
List of Messages in DMQ Type Order	157
List of Pictures.....	159
List of Tables	159

Introduction

About This Interface Description

Which Version

This manual covers the 6.0 - 6.6 release of the Bartrack system. If you do not know which version you have, do the following:

1. Start Bartrack (either web or VT session)
2. From web: Choose the menu item About/About Bartrack
From VT: Press the **F10** key
3. A display box shows which version of Bartrack is installed.
4. Click on the button **<OK>** (web) or press the key **DO** (VT)

All messages that were available in Bartrack R5 are described in the "SFT for Bartrack R5" interface description. These messages are still valid, but are not described in this description. Some of the described messages are not valid for all Bartrack 6.x versions.

The Intended Audience

This manual is written for a reader with prior knowledge of software design and programming.

The Bartrack Shop Floor Transaction (SFT) interface is intended to be used by software developers that want to utilize the Bartrack functions from another system. The SFT interface is used by software capable of handling DmQ messages. If your software doesn't use DmQ, or if it is not feasible to use DmQ, the BarAPIx exists as a DLL for Windows PCs. Please refer to the "BarAPIx System Administrators Manual" for issues regarding the BarAPIx.

Scope of the Manual

This manual does not cover:

1. Programming techniques
2. Programming examples
3. Discussions on how to use the functions
4. Implementation and solutions of the necessary operations.
5. Usage of the Bartrack program. See User's Guide – KS001B01/EN for information on using Bartrack.
6. Descriptions of older messages. See "SFT for Bartrack R5".

Should you have any problems, and if you have a support agreement, you can reach the Prevas support by e-mail: support.sfc@prevas.se

You can also use fax or phone:

Fax: +46 703 869 144

Tel: +46 54 147 444

Depending on your support agreement with Prevas, this service may be open 24 hours a day, seven days a week.

Purpose of the Interface Description

The purpose of this document is to:

Define the interface between an external system and Bartrack.

Conventions

In order to keep the manual as easy to read as possible, the following conventions are used:

A request to press a key on the keyboard is written like this:

Press the **F10** key.

A request to choose a menu or menu item is written like this:

Choose the menu item **System/About**.

A request to click on an on-screen button is written like this:

Click on the button **Product structure**.

References to directories, files, processes etc. on the server is written like this:

The **BAR_TRACY** process scans the **BAR_TCY_OUT_DIR** directory.

A sequence of instructions to be carried out in order is written like this:

1. This is step 1
2. This is step 2
3. Etc.

A list of items is written like this:

- This is an item
- This is another item
- Etc.

About Bartrack

What Does It Do?

Bartrack is a traceability system for individuals produced in factories specialized in electronics manufacturing. The purpose of using Bartrack is to:

- Create serial numbers for the individuals
- Keep track of the assembled parts of an individual
- Print out bar-code labels

- Keep master traceability systems updated
- Monitor the product structure

Bartrack assigns every produced individual with traceability demands a serial number. To this unique number, assorted information about the individual is connected, for example product number, structure information, manufacturing time.

When an individual is ready to leave the factory, its information in Bartrack is deleted. But since Bartrack is one of the sources of traceability information, it first sends the information to the master traceability database, such as Tracy.

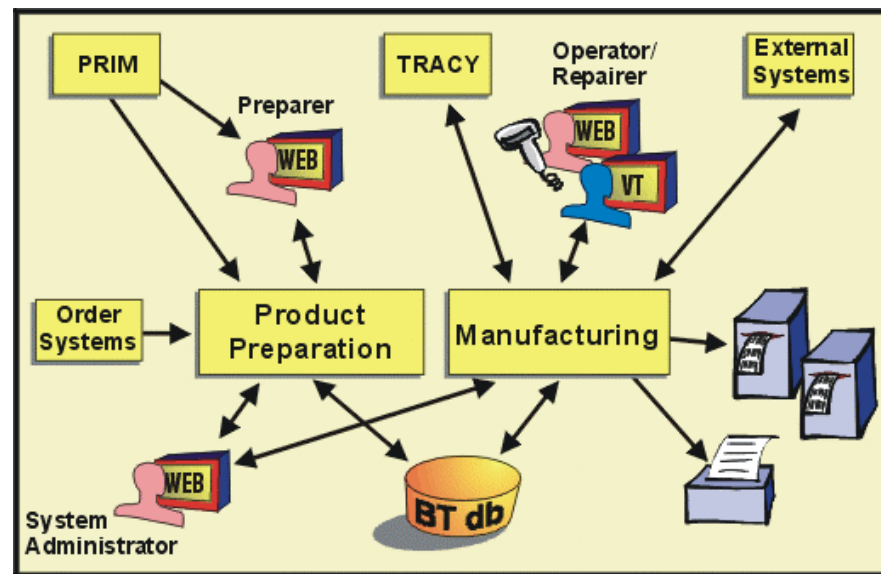
System Overview

The Bartrack system consists mainly of two units:

- The Prepare/Administration unit
- The Manufacturing unit

Both of these units have access to the Bartrack Rdb-database.

The communication between Bartrack and other systems all go through these two units, except for the direct access the System Administrator has to the Rdb-database.



A system overview

There are four main user categories that use Bartrack:

- Operator
- Repairer
- Preparer
- System Administrator

They put in information, request tasks and read output information.

In addition to the real users, there are a number of virtual users, for example the Tracy interface or the SFT interface.

SFT Overview

Purpose of SFT

When Bartrack is installed in a factory, there might be other systems present already. In order to make the Bartrack functions available to other systems, there is an interface called the Shop Floor Transaction interface, or SFT in short.

Having an interface to Bartrack ensures that all users of the Bartrack functions have the same definition of the functions, and that updates of Bartrack do not affect systems using the SFT interface (unless new functions are implemented).

An external system might want to use this interface in order to change or request information about individuals in the Bartrack system, for example, updating test status or requesting information.

Bartrack also sends information to external systems at specified events in Bartrack, for example individual information or structure information.

General Description

The SFT interface from Bartrack generates DmQ messages at certain occasions. The occasions and destination are defined in this document as well as the messages.

The SFT interface also receives requests from the external system, and performs the transaction and returns a reply message.

The information handled by the SFT interface is preferably used for direct manufacturing purposes at the shop floor. Typically information of individuals when they are created, tested, assembled and shipped.

The Bartrack interface SFT can act both as a master and a slave. When the SFT act as a master, a DmQ message is sent to a receiver. When the SFT act as a slave, the interface is waiting for a message and when it is received, the correct response message is sent to the sender.

When SFT acts as a slave there are five OpenVMS processes handling the reception of the messages, and when SFT acts as a master, there is one process that handles the sending of messages.

Usage

The functionality of Bartrack and the SFT interface is guaranteed as long as the guidelines, protocols and message formats in this interface description are followed.

If a software developer deviates from this interface description, the functionality of Bartrack cannot be guaranteed.

Message Sequences

Message sequence when SFT receives requests

Request	DMQ #	Transaction	Subtype	Process	Reply message	DMQ #	Transaction	Error message	DMQ #
NewIDReq	12067	SERCRT		SFT_CRT	NewIDInfo	12066	RTNMSG	NewIDErr	12065
NewIDReq_1	12106	SERCRT		SFT_CRT	NewIDInfo_1	12108	RTNMSG	NewIDErr_1	12107
NewIDReq_2	12138	SERCRT		SFT_CRT	NewIDInfo_2	12140	RTNMSG	NewIDErr_2	12139
ChkFlowReq	12120	CHKTST		SFT_IDI	ChkFlowInfo	12122	RTNMSG	ChkFlowErr	12121
IDReq	12054	SERDSP		SFT_IDI	IDInfo1	12052	RTNMSG	IDErr	12051
IDReq_1	12112	SERDSP		SFT_IDI	IDInfo_1	12114	RTNMSG	IDErr_1	12113
IDReq_2	12132	SERDSP		SFT_IDI	IDInfo_2	12134	RTNMSG	IDErr_2	12133
IDReq	12054	ADDNOS		SFT_IDI	IDInfo2	12053	RTNMSG	IDErr	12051
LoginReq	12040	BLOGIN		SFT_IDI	LoginReply	12041	RTNMSG	LoginReply	12041
OrdReq	12080	ORDREQ		SFT_IDI	OrdInfo	12079	RTNMSG	OrdErr	12078
OrdReq_1	12094	ORDREQ		SFT_IDI	OrdInfo_1	12096	RTNMSG	OrdErr_1	12095
TstReq	12100	TSTREQ		SFT_IDI	TstInfo	12102	RTNMSG	TstErr	12101
FromTcyReq	12123	TCYREQ		SFT_IDI	FromTcyAck	12125	RTNMSG	FromTcyErr	12124
GetAdd	12126	NOTGEN		SFT_IDI	GetAddInfo	12128	RTNMSG	GetAddErr	12127
GenAdd	12129	GENADD		SFT_IDI	GenAddAck	12131	RTNMSG	GenAddErr	12130
MoveIDReq	12064	SERMVE	SH	SFT_SHP	MoveIDAck	12063	RTNMSG	MoveIDErr	12062
MoveIDReq	12064	SERMVE	SC	SFT_SHP	MoveIDAck	12063	RTNMSG	MoveIDErr	12062
MoveIDReq	12064	SERMVE	ST	SFT_SHP	MoveIDAck	12063	RTNMSG	MoveIDErr	12062
ReactIDReq	12090	SERMVE		SFT_SHP	ReactIDAck	12092	RTNMSG	ReactIDErr	12091
PrtLabReq	12074	SERLBL		SFT_SHP	PrtLabAck	12073	RTNMSG	PrtLabErr	12075
AsmIDReq	12060	SERPCA		SFT_SHP	AsmIDAck	12059	RTNMSG	AsmIDErr	12058
AsmIDReq	12060	SERPCD		SFT_SHP	AsmIDAck	12059	RTNMSG	AsmIDErr	12058
IDComplReq	12081	SERCPL		SFT_SHP	IDComplAck	12082	RTNMSG	IDComplErr	12083
FShipIdReq	12103	SERFSH		SFT_SHP	FShipIdAck	12105	RTNMSG	FShipIdErr	12104
UpdIDReq	12071	SERTST		SFT_UPD	UpdIDAck	12072	RTNMSG	UpdIDErr	12069
UpdIDReq	12071	SERCOM		SFT_UPD	UpdIDAck	12072	RTNMSG	UpdIDErr	12069
UpdIDReq	12071	SEREXE		SFT_UPD	UpdIDAck	12072	RTNMSG	UpdIDErr	12069

UpdIDReq_1	12087	SERTST		SFT_UPD	UpdIDAck_1	12089	RTNMSG	UpdIDErr_1	12088
UpdIDReq_1	12087	SERCOM		SFT_UPD	UpdIDAck_1	12089	RTNMSG	UpdIDErr_1	12088
UpdIDReq_1	12087	SEREXE		SFT_UPD	UpdIDAck_1	12089	RTNMSG	UpdIDErr_1	12088
UpdIDReq_1	12087	SERPRO		SFT_UPD	UpdIDAck_1	12089	RTNMSG	UpdIDErr_1	12088
UpdIDReq_2	12097	SERTST		SFT_UPD	UpdIDAck_2	12099	RTNMSG	UpdIDErr_2	12098
UpdIDReq_2	12097	SERCOM		SFT_UPD	UpdIDAck_2	12099	RTNMSG	UpdIDErr_2	12098
UpdIDReq_2	12097	SEREXE		SFT_UPD	UpdIDAck_2	12099	RTNMSG	UpdIDErr_2	12098
UpdIDReq_2	12097	SERPRO		SFT_UPD	UpdIDAck_2	12099	RTNMSG	UpdIDErr_2	12098
UpdIDReq_3	12141	SERTST		SFT_UPD	UpdIDAck_3	12143	RTNMSG	UpdIDErr_3	12142
UpdIDReq_3	12141	SERCOM		SFT_UPD	UpdIDAck_3	12143	RTNMSG	UpdIDErr_3	12142
UpdIDReq_3	12141	SEREXE		SFT_UPD	UpdIDAck_3	12143	RTNMSG	UpdIDErr_3	12142
UpdIDReq_3	12141	SERPRO		SFT_UPD	UpdIDAck_3	12143	RTNMSG	UpdIDErr_3	12142
UpdIDReq_3	12141	SERINF		SFT_UPD	UpdIDAck_3	12143	RTNMSG	UpdIDErr_3	12142
IDStrReq	12057	SERSTP		SFT_STR	IDStrInfo	12056	RTNMSG	IdStrErr	12055
IDStrReq_1	12115	SERSTP		SFT_STR	IDStrInfo_1	12117	RTNMSG	IdStrErr_1	12116
IDStrReq_2	12135	SERSTP		SFT_STR	IDStrInfo_2	12137	RTNMSG	IdStrErr_2	12136
ProStrReq	12084	PROREQ		SFT_PRO	ProStrInfo	12085	RTNMSG	ProStrErr	12086
TpcReq	12109	TPCREQ		SFT_PRO	TpcInfo	12111	RTNMSG	TpcErr	12110
								BarUnknown	12093
MsgToFile	12090			DTF					

Bartrack uses a switch in the **EXTERNAL_SYSTEM** table where each system connected to Bartrack can have its own setting. The setting can be either 5 or 6. 5 mean that the R5 messages are sent to that system, and 6 mean that the 6.0 messages are sent.

Message sequence when SFT sends automatic version 5 messages

Bartrack event	Message sent	Transaction	#	Explanation
Create	MsA_CreIDInfo	SERCRT	12023	The message contains the data of the individual. It is sent when the Create event occurs in Bartrack.
Register Manually	MsA_CreIDInfo	SEREXT	12023	The message contains the data of the individual. It is sent when the Register Manually event occurs in Bartrack.
Register from Tracy				No message is sent.
Assemble	MsA_AsmIDInfo_2	SERPCA	12024	The message contains the data of the individual and all its children. It is sent when the Assemble event occurs in Bartrack.
Change Data	MsA_UpdIDInfo	SERREV	12013	The message contains the data of the individual. It is sent when the Change Data event occurs in Bartrack.
Change Structure				No message is sent.
Store	MsA_UpdIDInfo	SERSTO	12013	The message contains the data of the individual. It is sent when the Store event occurs in Bartrack.
Ship	MsA_UpdIDInfo	SERSHP	12013	The message contains the data of the individual. It is sent when the Ship event occurs in Bartrack.
Scrap	MsA_ScpIDInfo	SERSCP	12022	The message contains scrap information. It is sent when the Scrap event occurs in Bartrack.

Message sequence when SFT sends automatic version 6 messages

Bartrack event	Message sent	Transaction	#	Explanation
Create	NewInfo	SERCRT	12061	The message contains the data of the individual. It is sent when the Create event occurs in Bartrack.
Register Manually	NewInfo	SEREXT	12061	The message contains the data of the individual. It is sent when the Register Manually event occurs in Bartrack.
Register from Tracy				No message is sent.
Assemble	AsmInfo	SERPCA	12050	The message contains the data of the individual and all its children. It is sent when the Assemble event occurs in Bartrack.
Change Data	UpdInfo	SERREV	12070	The message contains the data of the individual. It is sent when the Change Data event occurs in Bartrack.
Change Structure				No message is sent.
Store	UpdInfo	SERSTO	12070	The message contains the data of the individual. It is sent when the Store event occurs in Bartrack.
Ship	UpdInfo	SERSHP	12070	The message contains the data of the individual. It is sent when the Ship event occurs in Bartrack.
Scrap	ScpInfo	SERSCP	12068	The message contains scrap information. It is sent when the Scrap event occurs in Bartrack.

Message sequence when SFT sends automatic version 7 messages

Bartrack event	Message sent	Transaction	#	Explanation
Create	NewInfo_1	SERCRT	12118	The message contains the data of the individual. It is sent when the Create event occurs in Bartrack.
Register Manually	NewInfo_1	SEREXT	12118	The message contains the data of the individual. It is sent when the Register Manually event occurs in Bartrack.
Register from Tracy				No message is sent.
Assemble	AsmlInfo_1	SERPCA	12119	The message contains the data of the individual and all its children. It is sent when the Assemble event occurs in Bartrack.
Change Data	UpdInfo	SERREV	12070	The message contains the data of the individual. It is sent when the Change Data event occurs in Bartrack.
Change Structure				No message is sent.
Store	UpdInfo	SERSTO	12070	The message contains the data of the individual. It is sent when the Store event occurs in Bartrack.
Ship	UpdInfo	SERSHP	12070	The message contains the data of the individual. It is sent when the Ship event occurs in Bartrack.
Scrap	ScpInfo	SERSCP	12068	The message contains scrap information. It is sent when the Scrap event occurs in Bartrack.

Message Sequences

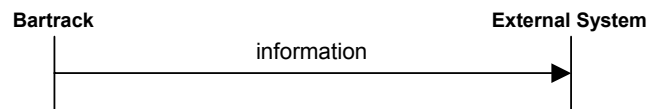
Transaction Overview

All transaction sequences are described in the same way; Bartrack on the left side and other systems on the right side. Other systems are always referred to as "External system".

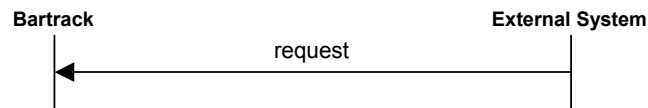
Before you can use any of the request messages, you must use the **LoginReq** message to login.

There are four different types of sequences. The contents may vary, but the flow is always the same.

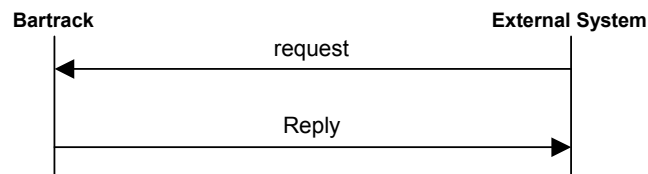
The first one is initiated by Bartrack:



The second one is initiated by the external system:

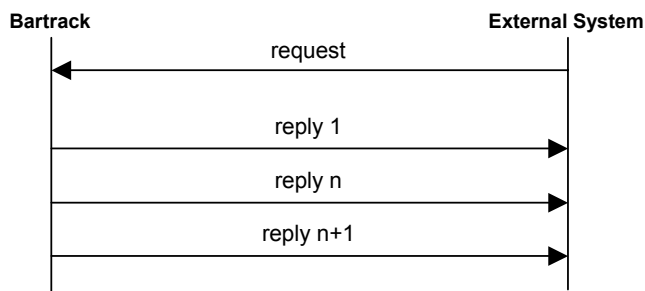


The third one is initiated by the external system, and Bartrack replies:



The reply could be the requested information or an error.

The fourth one is initiated by the external system, and the requested information does not fit in one single reply message:



The last reply contains empty information to inform the receiver that the sequence has ended.

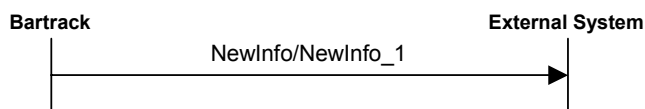
Report New Individual (NewInfo/NewInfo_1)

The purpose of this sequence is to inform an external system when a new individual has been created. Bartrack may automatically initiate the sequence when a user creates a serial number.

Preconditions:

- An individual has been created and the event shall be reported to an external system.
- An individual has been registered and the event shall be reported to an external system.

No error routines are defined.



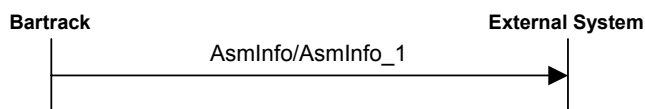
Report all Children for a Parent (AsmInfo/AsmInfo_1)

The purpose of this sequence is to inform an external system about what children those currently are connected to an individual. Bartrack may automatically initiate the sequence when a user has completed an assembly.

Preconditions:

- An individual has been changed and the event shall be reported to the external system.

No error routines are defined.



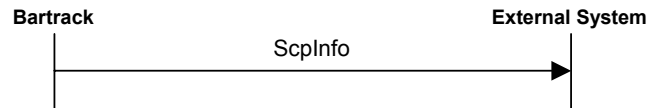
Report Scrapped Individual (ScplInfo)

The purpose of this sequence is to inform an external system when an individual is scrapped in Bartrack. Bartrack may automatically initiate the sequence when a user scraps a serial number.

Preconditions:

- An individual has been changed and the event shall be reported to the external system.

No error routines are defined.



Report Updated Individual (UpdInfo)

The purpose of this sequence is to inform an external system when information for an individual has been changed. Bartrack may automatically initiate the sequence when a user changes the information for a serial number.

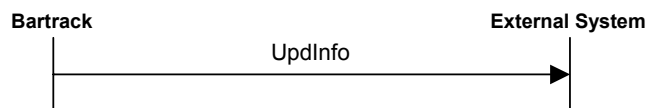
The changes that trigger the message are:

- Product Number
- R-state
- Store
- Ship

Preconditions:

- An individual has been changed and the event shall be reported to the external system.

No error routines are defined.



Request Information about an Individual (IDReq/IDReq_1/IDReq_2)

The purpose of this sequence is to let an external system request information about an individual.

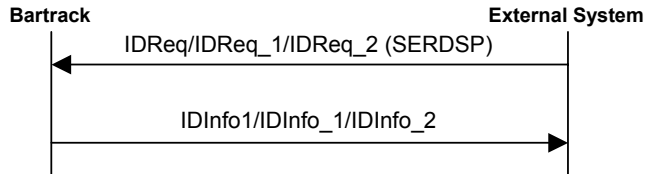
Preconditions:

- If the individual exists in Bartrack, the standard sequence is executed.
- If the individual does not exist in Bartrack, the error sequence is executed.

Standard Sequence, Transaction SERDSP

The external system sends the **IDReq**/**IDReq_1**/**IDReq_2** message to Bartrack with a request for information about an individual.

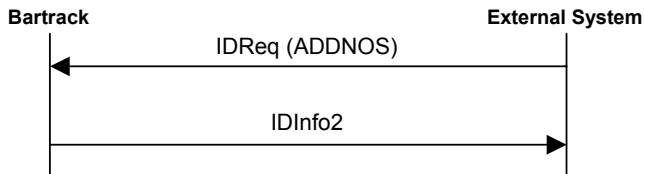
If the requested individual is valid and the transaction is **SERDSP**, Bartrack responds with an **IDInfo1**/**IDInfo_1**/**IDInfo_2** message for the individual.



Standard Sequence, Transaction ADDNOS

The external system sends the **IDReq** message to Bartrack with a request for information about an individual.

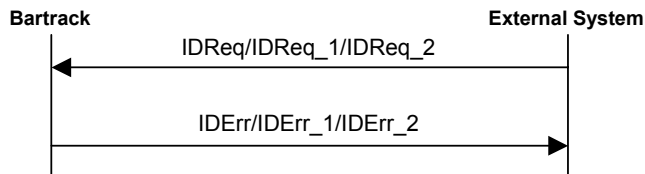
If the requested individual is valid and the transaction is **ADDNOS**, Bartrack responds with an **IDInfo2** message for the individual.



Error Sequence 1

The external system sends the **IDReq**/**IDReq_1**/**IDReq_2** message to Bartrack with a request for information about an individual.

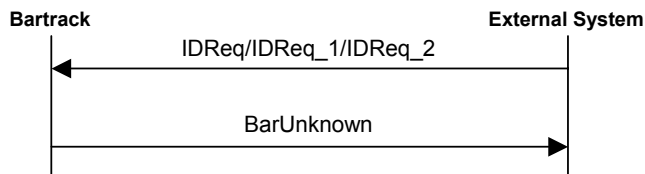
If the requested individual is invalid, Bartrack responds with an **IDErr**/**IDErr_1**/**IDErr_2** message.



Error Sequence 2

The external system sends the **IDReq**/**IDReq_1**/**IDReq_2** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request Structure Information about an Individual (IDStrReq/IDStrReq_1/IDStrReq_2)

This message sequence lets an external system request what children an individual has.

Preconditions:

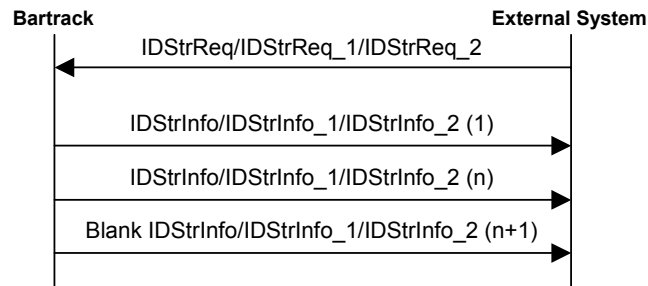
- If the individual exists in Bartrack, the standard sequence is executed.
- If the individual does not exist in Bartrack, the error sequence is executed.

Standard Sequence

The external system sends the **IDStrReq/IDStrReq_1/IDStrReq_2** message to Bartrack with a request for information about an individual's structure.

If the requested individual is valid, Bartrack responds with a number of **IDStrInfo/IDStrInfo_1/IDStrInfo_2** messages for the individual, one for each child.

After the last message, (n), an extra message (n+1) is sent, containing a blank child field (no serial number) to indicate that there are no more children for the individual.

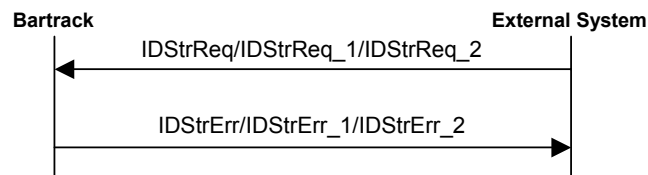


Bartrack gives only the first level of children. If the external system needs several levels, it has to send a new request for each one of the children.

Error Sequence 1

The external system sends the **IDStrReq/IDStrReq_1/IDStrReq_2** message to Bartrack with a request for information about an individual's structure.

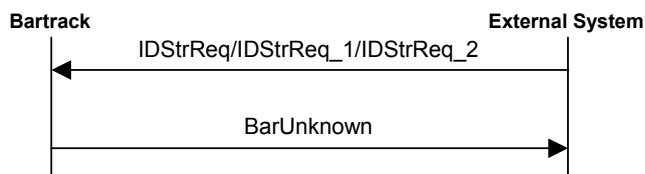
If the requested individual is invalid, Bartrack responds with an **IDStrErr/IDStrErr_1/IDStrErr_2** message.



Error Sequence 2

The external system sends the **IDStrReq**/**IDStrReq_1**/**IDStrReq_2** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to Add or Remove a Child (AsmIDReq)

This message sequence lets an external system add or remove an individual's children. The transaction indicates whether an addition or a removal is to be performed.

Preconditions:

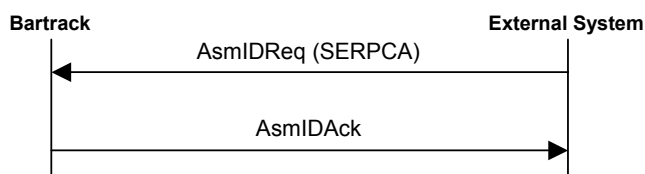
- If the transaction is allowed, the standard sequence is executed.
- If the transaction is not allowed for the individual, the error sequence is executed.

Standard Sequence, Transaction SERPCA

The external system sends the **AsmIDReq** message to Bartrack with a request to connect two individuals. If the requested individuals are valid to connect, Bartrack responds with the **AsmIDAck** message for the individual.

Preconditions:

- The parent individual and the child exist in Bartrack and are valid.
- The product structure constraints are met.



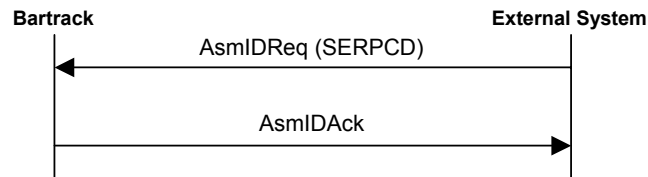
Standard Sequence, Transaction SERPCD

The external system sends the **AsmIDReq** message to Bartrack with a request to disconnect two individuals. If the requested individuals are valid to disconnect, Bartrack responds with the **AsmIDAck** message for the individual.

Note. The **AsmIDReq** message has a flag that will override the **ASSEMBLE_DONE** marking in Bartrack. If you always want to be able to disconnect a child, this flag must be used.

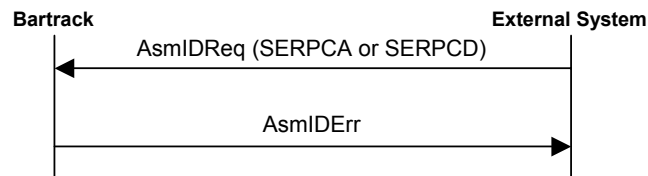
Preconditions:

- The parent individual and the child exist in Bartrack and are valid.
- The **ASSEMBLE_DONE** is either over-ridden or not set.



Error Sequence 1

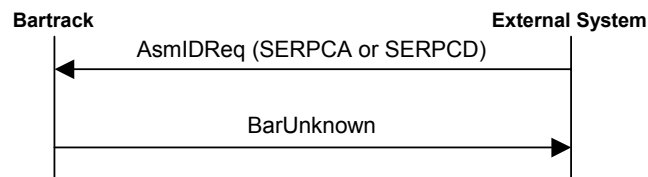
The external system sends the **AsmIDReq** message to Bartrack with a request containing two individuals. If the requested individuals are invalid to connect or disconnect, Bartrack responds with an **AsmIDErr** for the individual.



Error Sequence 2

The external system sends the **IDReq** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to Store, Ship or Scrap an Individual (MoveIDReq)

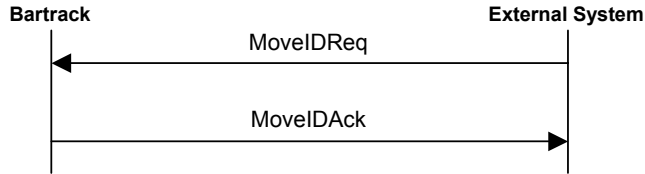
The purpose of the sequence is to let the external system move an individual. The subtype of the **MoveIDReq** message indicates whether a store, ship or scrap transaction is requested.

Preconditions:

- If the transaction is allowed for the individual, the standard sequence is executed.
- If the transaction is not allowed for the individual, the error sequence is executed.

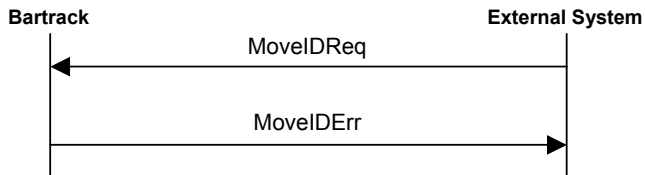
Standard Sequence

The external system sends the **MoveIDReq** message to Bartrack with a move request for an individual. If the requested individual is valid to store, ship or scrap, Bartrack responds with the **MoveIDAck** message for the individual.



Error Sequence 1

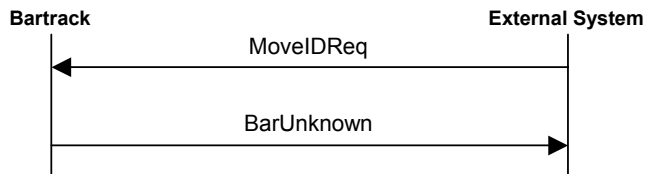
The external system sends the **MoveIDReq** message to Bartrack with a request for an individual. If the requested individual is not valid to store, ship or scrap, Bartrack will respond with the **MoveIDErr** for the individual.



Error Sequence 2

The external system sends the **MoveIDReq** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to Ship (forced) an Individual (FShipIdReq)

The purpose of the sequence is to let the external system force a ship for an individual.

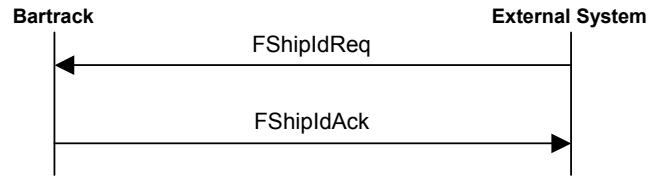
The forced ship skips the structure completeness check and the test status check.

Preconditions:

- If the transaction is allowed for the individual, the standard sequence is executed.
- If the transaction is not allowed for the individual, the error sequence is executed.

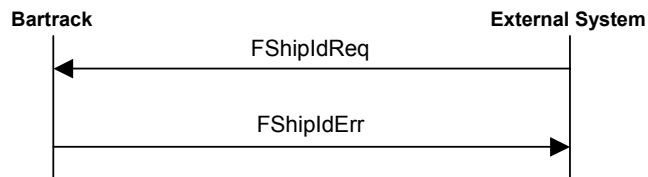
Standard Sequence

The external system sends the **FShipIdReq** message to Bartrack with a forced ship request for an individual. If the requested individual is valid to force ship, Bartrack responds with the **FShipIdAck** message for the individual.



Error Sequence 1

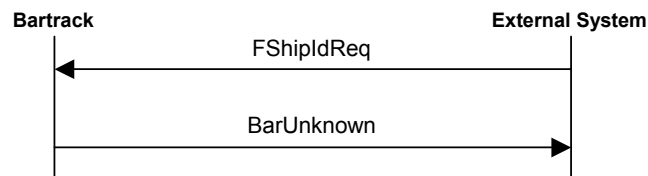
The external system sends the **FShipIdReq** message to Bartrack with a request for an individual. If the requested individual is not valid to store, ship, force ship or scrap, Bartrack will respond with the **FShipIdErr** for the individual.



Error Sequence 2

The external system sends the **FShipIdReq** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to Reactivate an Individual (ReactIDReq)

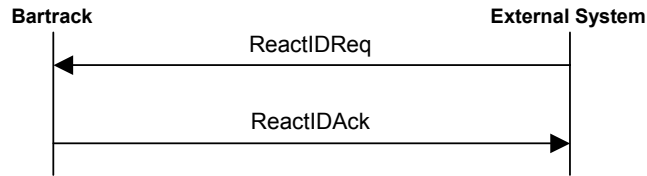
The purpose of the sequence is to let the external system reactivate an individual.

Preconditions:

- If the transaction is allowed for the individual, the standard sequence is executed.
- If the transaction is not allowed for the individual, the error sequence is executed.

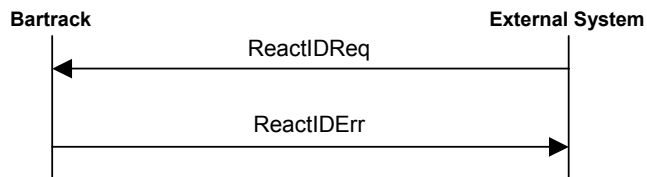
Standard Sequence

The external system sends the **ReactIDReq** message to Bartrack with a reactivate request for an individual. If the requested individual is valid to reactivate, Bartrack responds with the **ReactIDAck** message for the individual.



Error Sequence 1

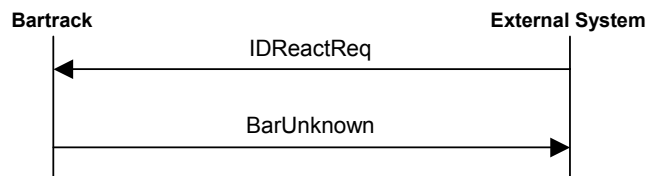
The external system sends the **ReactIDReq** message to Bartrack with a request for an individual. If the requested individual is not valid to reactivate, Bartrack will respond with the **ReactIDErr** for the individual.



Error Sequence 2

The external system sends the **ReactIDReq** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to Create Individuals (NewIDReq/NewIDReq_1/NewIDReq_2)

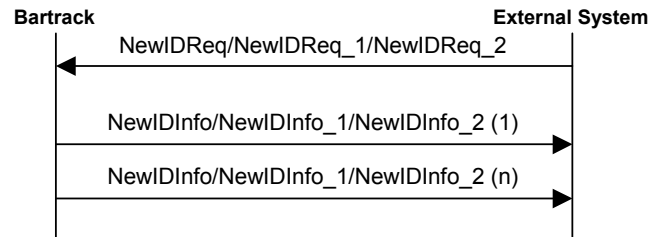
The purpose of the sequence is to let the external system request the creation of new individuals.

Preconditions:

- If the information is correct to create individuals in Bartrack, the standard sequence is executed.
- If the information is not correct to create individuals in Bartrack, the error sequence is executed.

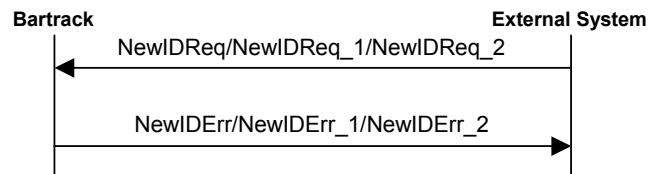
Standard Sequence

The external system sends the **NewIDReq/NewIDReq_1/NewIDReq_2** message to Bartrack. If the request is valid, Bartrack responds with the requested number (n) of **NewIDInfo/NewIDInfo_1/NewIDInfo_2** messages, one for each created individual.



Error Sequence 1

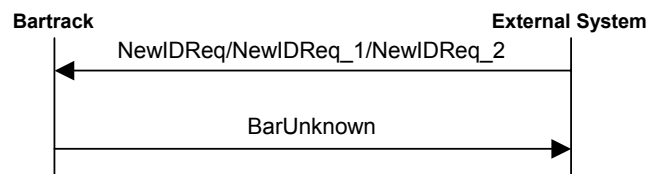
The external system sends the **NewIDReq/NewIDReq_1/NewIDReq_2** message to Bartrack with a request to create an individual. If the request is not valid, Bartrack will respond with the **NewIDErr/NewIDErr_1/NewIDErr_2** for the individual.



Error Sequence 2

The external system sends the **NewIDReq/NewIDReq_1/NewIDReq_2** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to Update an Individual (UpdIDReq/UpdIDReq_1/UpdIDReq_2/UpdIDReq_3)

The purpose of the sequence is to let an external system update an individual.

You will always receive a message back if the field **Reply wanted** = T. The message will be **UpdIDAck/UpdIDAck_1/UpdIDAck_2/UpdIDAck_3** for

successful updates and `UpdIDErr/UpdIDErr_1/UpdIdErr_2/UpdIdErr_3` for unsuccessful updates.

If you set the field `Reply wanted = N`, you will not get any messages back, and if you set the field `Reply wanted = Y`, you will only get error messages back.

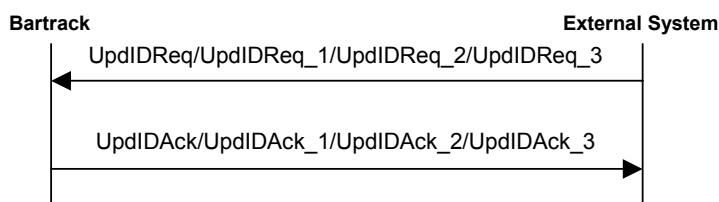
There are five transactions for this message. SERTST, SERCOM, SEREXE, SERPRO and SERINF. They work in exactly the same way except that they deal with different fields in the message.

Preconditions:

- If the transaction is allowed for the individual, the standard sequence is executed.
- If the transaction is not allowed for the individual, the error sequence is executed.

Standard Sequence 1, Always Reply

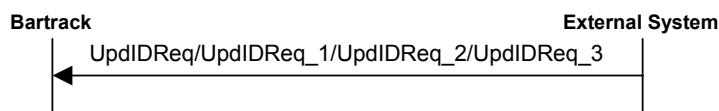
The external system sends the `UpdIDReq/UpdIDReq_1/UpdIDReq_2/UpdIDReq_3` message to Bartrack with a request to update an individual with the field `Reply wanted = T` and the requested transaction is valid for the individual.



The `UpdIDErr/UpdIDErr_1/UpdIDErr_2/UpdIDErr_3` is replied instead of the `UpdIDAck/UpdIDAck_1/UpdIDAck_2/UpdIDAck_3` if an error occurs.

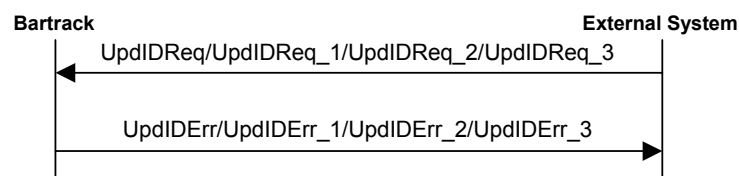
Standard Sequence 2, Never Reply

The external system sends the `UpdIDReq/UpdIDReq_1/UpdIDReq_2/UpdIDReq_3` message to Bartrack with a request to update an individual with the field `Reply wanted =N` and the requested transaction is valid for the individual. Bartrack will send no reply.



Error Sequence 1

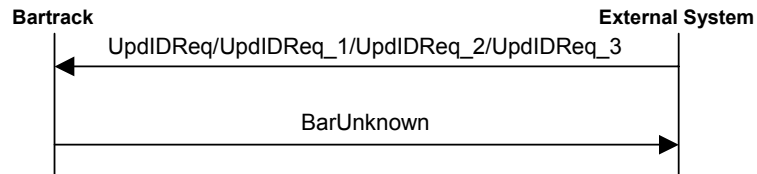
The external system sends the `UpdIDReq/UpdIDReq_1/UpdIDReq_2/UpdIDReq_3` message to Bartrack with a request for an individual. The requested update is not valid and Bartrack responds with an `UpdIDErr/UpdIDErr_1/UpdIDErr_2/UpdIDErr_3` for the individual if the field `Reply wanted` is T or Y.



Error Sequence 2

The external system sends the **UpdIDReq/UpdIDReq_1/UpdIDReq_2/UpdIDReq_3** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to Check completeness for an Individual structure (IDComplReq)

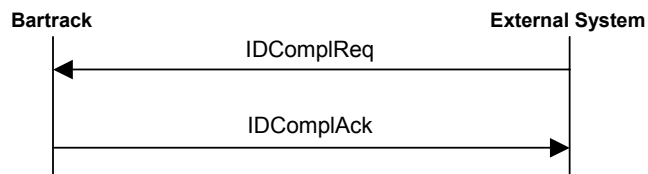
The purpose of the sequence is to send a request to ask Bartrack if the structure is complete for an individual.

Preconditions:

- If the transaction is allowed for the individual, the standard sequence is executed.
- If the transaction is not allowed for the individual, the error sequence is executed.

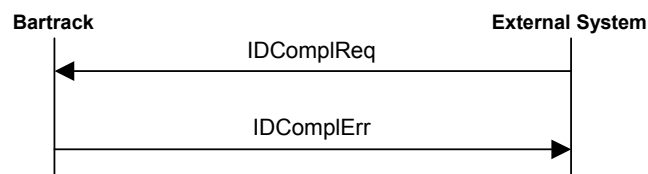
Standard Sequence

The external system sends the **IDComplReq** message to Bartrack with a request for the individual structure completeness status. If the requested individual exists, Bartrack respond with the **IDComplAck** message.



Error Sequence 1

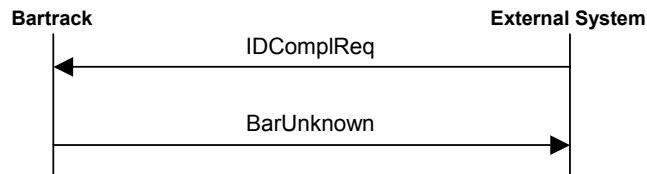
The external system sends the **IDComplReq** message to Bartrack with a request to get an answer if individual structure is complete. If the request is not valid, Bartrack will respond with the **IDComplErr** for the individual.



Error Sequence 2

The external system sends the **IDComp1Req** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



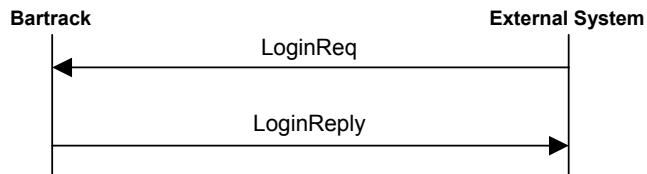
Request to Login (LoginReq)

The purpose of the sequence is to send a request to ask Bartrack to verify the user-ID and password. The field **Login success** will contain a Y if the login was successful and a N if not.

Preconditions:

- None

No error messages are defined.



Request for Structure Information of a Product

The message sequence lets an external system request what children a known product has.

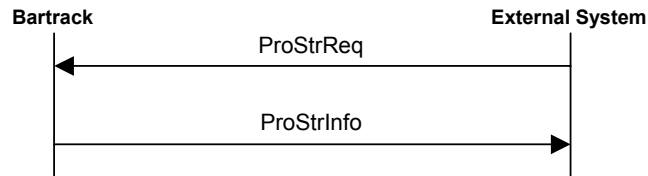
Preconditions:

- If the product exists in Bartrack, the standard sequence is executed.
- If the product does not exist in Bartrack, the error sequence is executed.

Standard Sequence

The external system sends the **ProStrReq** message to Bartrack with a request for information about a product's structure.

If the requested product is valid, Bartrack responds with a **ProStrInfo** message for the product.

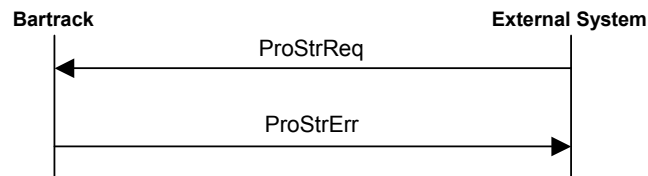


Bartrack gives only the first level of children. If the external system needs several levels, it has to send a new request for each one of the children.

Error Sequence 1

The external system sends the **ProStrReq** message to Bartrack with a request for information about a products' structure.

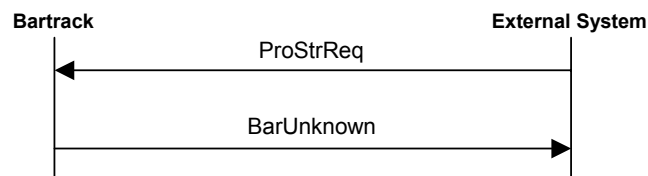
If the requested product is invalid, Bartrack responds with a **ProStrErr** message.



Error Sequence 2

The external system sends the **ProStrReq** message to Bartrack with a request for information about a product.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request Information about an Order (OrdReq/OrdReq_1)

This message sequence lets an external system request the information for a specific order.

Preconditions:

- If the order number exists in Bartrack, the standard sequence is executed.

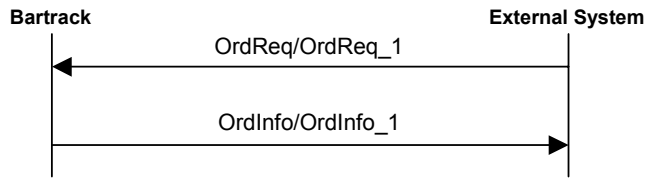
- If the order number does not exist in Bartrack, the error sequence is executed.

Standard Sequence

The external system sends the **OrdReq/OrdReq_1** message to Bartrack with a request for information about an order.

If the requested order number is valid, Bartrack responds with an **OrdInfo/OrdInfo_1** message for the order.

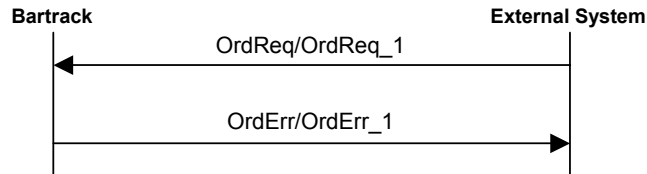
Bartrack gives only the first level of children. If the external system needs several levels, it has to send a new request for each one of the children.



Error Sequence 1

The external system sends the **OrdReq/OrdReq_1** message to Bartrack with a request for information about an order.

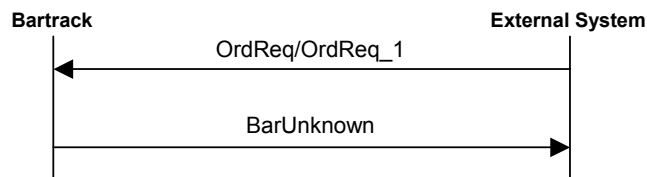
If the requested order number is invalid, Bartrack responds with an **OrdErr/OrdErr_1** message.



Error Sequence 2

The external system sends the **OrdReq/OrdReq_1** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request Test Status Information about an Individual (TstReq)

The purpose of this sequence is to let an external system request test status information about an individual.

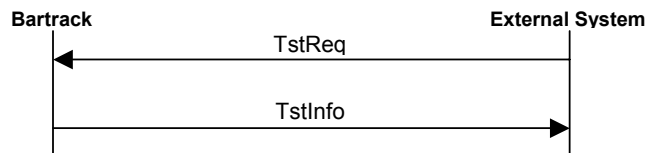
Preconditions:

- If the individual exists in Bartrack, the standard sequence is executed.
- If the individual does not exist in Bartrack, the error sequence is executed.

Standard Sequence

The external system sends the **TstReq** message to Bartrack with a request for test status information about an individual.

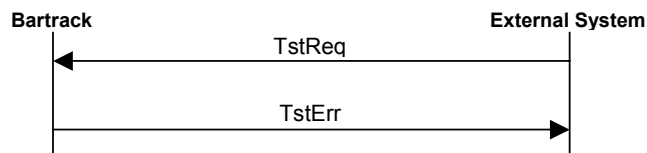
If the requested individual exists, Bartrack responds with an **TstInfo** message for the individual.



Error Sequence 1

The external system sends the **OrdReq/OrdReq_1** message to Bartrack with a request for information about an order.

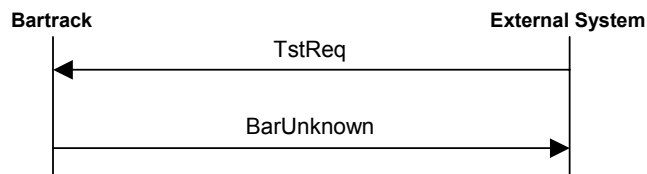
If the requested order number is invalid, Bartrack responds with an **OrdErr/OrdErr_1** message.



Error Sequence 2

The external system sends the **OrdReq/OrdReq_1** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request Print-out of Labels (PrtLabReq)

This message sequence lets an external system request the information for a specific order.

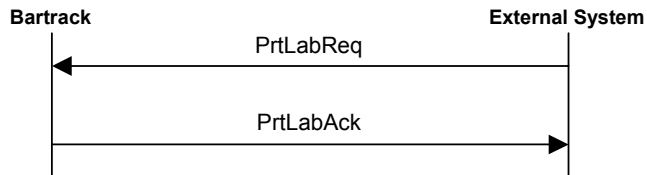
Preconditions:

- If the serial number and the label exist in Bartrack, the standard sequence is executed.
- If the serial number or the label does not exist in Bartrack, the error sequence is executed.

Standard Sequence

The external system sends the **PrtLabReq** message to Bartrack with a request to print out labels for an individual.

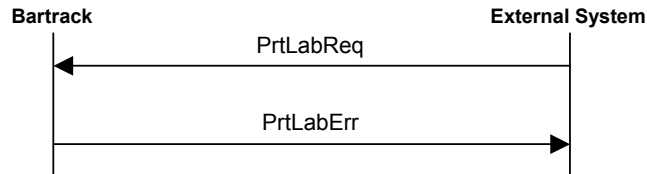
If the requested serial number is valid, Bartrack responds with a **PrtLabAck** message.



Error Sequence 1

The external system sends the **PrtLabReq** message to Bartrack with a request to print out labels for an individual.

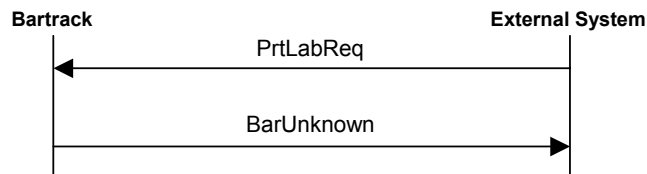
If the requested serial number is invalid or the label does not exist, Bartrack responds with a **PrtLabErr** message.



Error Sequence 2

The external system sends the **PrtLabReq** message to Bartrack with a request for information about an individual.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request for Test Production Comment Information

The message sequence lets an external system request what test production comment a known product has.

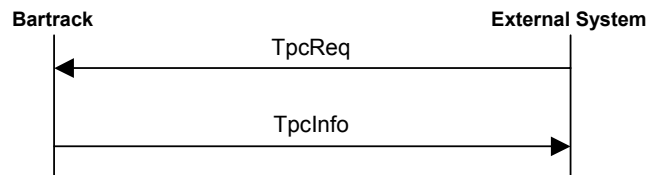
Preconditions:

- If the product exists in Bartrack, the standard sequence is executed.
- If the product does not exist in Bartrack, the error sequence is executed.

Standard Sequence

The external system sends the **TpcReq** message to Bartrack with a request for information about a product's test production comment.

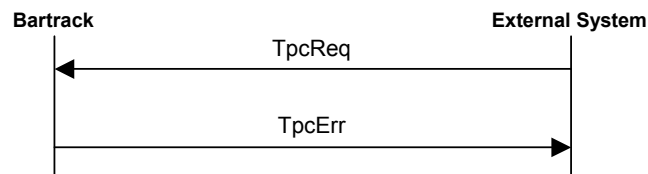
If the requested product or order is valid, Bartrack responds with a **TpcInfo** message for the product.



Error Sequence 1

The external system sends the **TpcReq** message to Bartrack with a request for information about a product's test production comment.

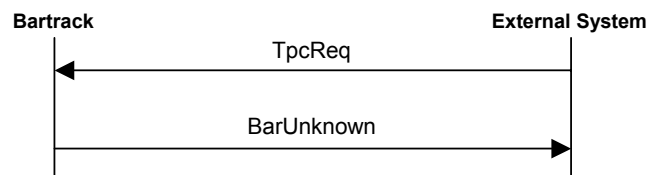
If the requested product is invalid, Bartrack responds with a **TpcErr** message.



Error Sequence 2

The external system sends the **TpcReq** message to Bartrack with a request for information about a product's test production comment.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request for flow control on test station

The message sequence lets an external system request a check if an update of the test status for an individual is allowed according to the defined production flow.

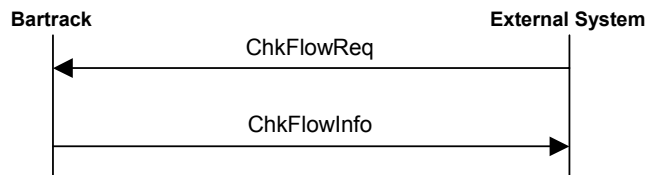
Preconditions:

- If the product exists in Bartrack, the standard sequence is executed.
- If the product does not exist in Bartrack, the error sequence is executed.

Standard Sequence

The external system sends the **ChkFlowReq** message to Bartrack with a request to check the production flow.

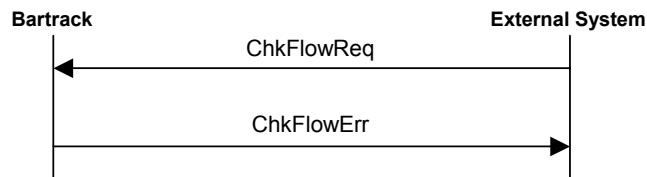
If the requested individual is valid, Bartrack responds with a **ChkFlowInfo** message for the individual.



Error Sequence 1

The external system sends the **ChkFlowReq** message to Bartrack with a request to check the production flow.

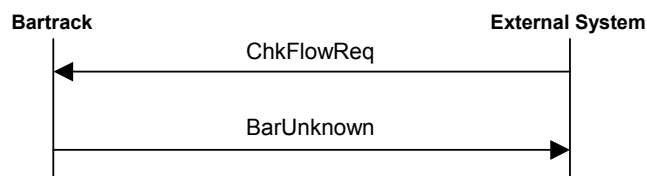
If the requested individual is invalid, Bartrack responds with a **ChkFlowErr** message.



Error Sequence 2

The external system sends the **ChkFlowReq** message to Bartrack with a request to check the production flow.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request individual data from the traceability system

The message sequence lets an external system request individual data from the traceability system.

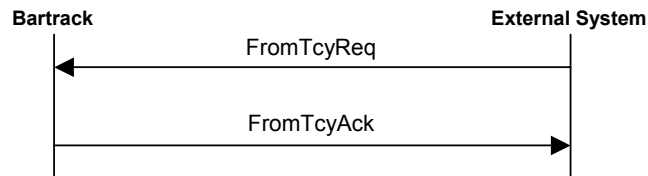
Preconditions:

- If it is legal to request individual data from the traceability system, the standard sequence is executed.
- If not, the error sequence is executed.

Standard Sequence

The external system sends the **FromTcyReq** message to Bartrack with a request to send a request for individual data from the traceability system.

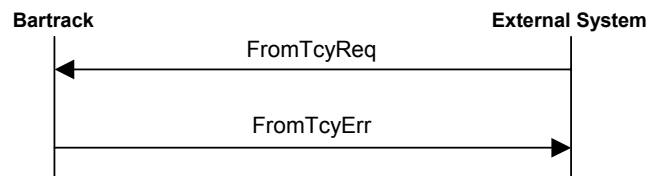
If the request to the traceability system is legal, Bartrack responds with a **FromTcyAck** message for the individual.



Error Sequence 1

The external system sends the **FromTcyReq** message to Bartrack with a request to send a request for individual data from the traceability system.

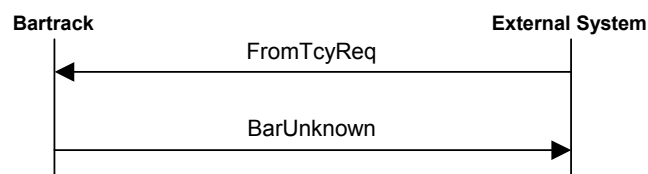
If the request to the traceability system is not legal, Bartrack responds with a **FromTcyErr** message.



Error Sequence 2

The external system sends the **FromTcyReq** message to Bartrack with a request to send a request for individual data from the traceability system.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request data for additional numbers (defined but not generated)

The message sequence lets an external system request data for additional numbers that are defined but not generated for an individual.

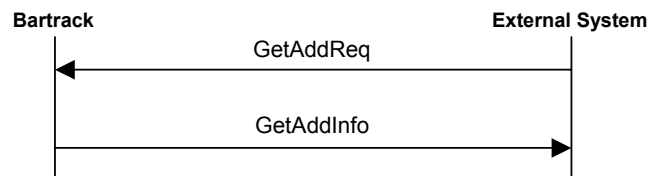
Preconditions:

- If the serial number exist in Bartrack, the standard sequence is executed.
- If not, the error sequence is executed.

Standard Sequence

The external system sends the **GetAddReq** message to Bartrack with a request for additional number information.

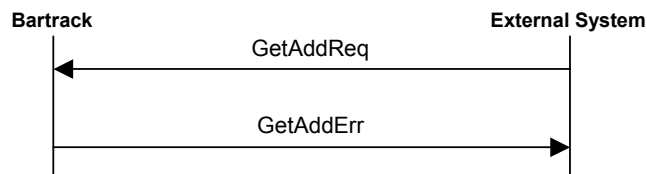
If the request is legal, Bartrack responds with a **GetAddInfo** message for the individual.



Error Sequence 1

The external system sends the **GetAddReq** message to Bartrack with a request for additional number information.

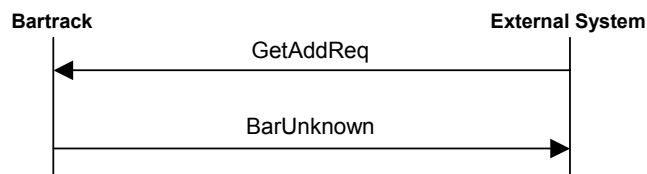
If the request is not legal, Bartrack responds with a **GetAddErr** message.



Error Sequence 2

The external system sends the **GetAddReq** message to Bartrack with a request for additional number information.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Request to generate an additional number

The purpose of the sequence is to let the external system generate an additional number for an individual.

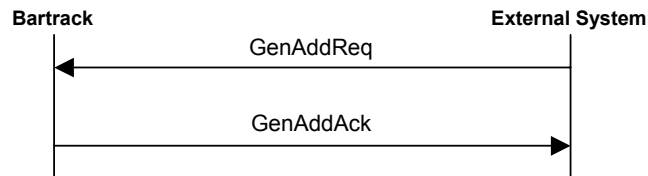
Preconditions:

- If the serial number exist and it is legal to generate the additional number in Bartrack, the standard sequence is executed.
- If not, the error sequence is executed.

Standard Sequence

The external system sends the **GenAddReq** message to Bartrack with a request to generate an additional number.

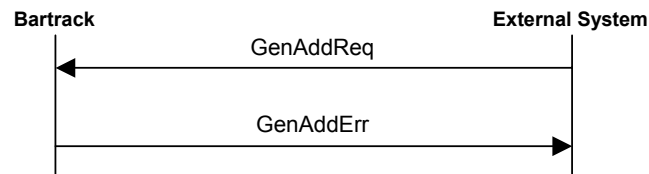
If the request is legal, Bartrack responds with a **GenAddAck** message for the individual.



Error Sequence 1

The external system sends the **GenAddReq** message to Bartrack with a request to generate an additional number.

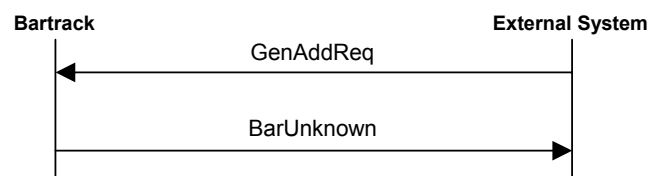
If the request is not legal, Bartrack responds with a **GenAddErr** message.



Error Sequence 2

The external system sends the **GenAddReq** message to Bartrack with a request to generate an additional number.

If the request is invalid, e.g. mismatch between SFT and Bartrack version, Bartrack responds with a **BarUnknown** message.



Messages

About SFT Messages

Each message is defined by a number of parameters:

Name	This is the name of the message.
Description	A short description of what the message does or contain.
DmQ message Type	A unique number that identifies the message.
DmQ message class:	The message class is the same for all SFT messages; 12000.
DmQ message priority:	When the priority is set to "High", the message is read before other messages in the target queue, that have the priority set to "Low". Normally set to "Low".
DmQ interest point:	DECmessageQ has several methods of message recovery services that indicate how the message shall be secured before sent call returns. The methods used in Bartrack are: MEM: The message shall be received by the target queue before the DmQ call returns. This is the fastest method but most insecure. DQF, UMA_SAF: The message shall be written to disk at the receiving side before the DmQ call returns. If this fails, the message is written to the local SAF file. This is a slower method but more secure.
Length:	A message consists of a specific number of bytes. The parameters specify the exact number of bytes in each message. The length could be fixed for the message type or varying depending of the data in the message.
Transaction:	Each message has a field called "Transaction". This field always contain the code for the requested type of message. For instance, the NewIDReq message has the transaction code SERCRT, which actually stands for SERIAL number CreaTe. Some of the messages may have more than one transaction code, and thus have multiple meanings. In addition to the transaction code, a subtype exists for some of the messages. The subtype specifies what special version of transaction is to be performed. N.B. Always consider the transaction when you are testing for message contents.
Message sequence:	Message can either be sent by themselves, or as a reply. The sequence for each message is shown with Bartrack on the left-hand side, and an external system on the right-hand side. The arrows between these systems illustrate which party initiates the message sequence, starting from the top.

Message contents

For each of the messages, there is a table defining which fields the message consists of, the type (length or size) and the order of the fields within the message.

All fields shall be allocated. Only in a few number of the messages, it is possible to have a varying number of fields, and in these messages, the number of fields are specified before they occur.

If the field doesn't have any relevant information, it shall be filled with blanks or spaces.

The table describing the fields contain the following information:

Name	Type	M	Description
The name of the field. Normally, the field-name is specific to the message. But some of the fields occur in many messages.	The type of contents in the field. Look at the table in section "Types" on page 147 for a description of the types.	M means mandatory. A "Y" in this column indicates that the information is mandatory. A blank indicates that the information in the field is optional. N.B. A mandatory field must be filled in, regardless who the initiator is.	A more thorough description of the field. The type defines most of the allowed values, but sometimes, extra information can be found here.

Some common fields

There are some fields that are present in most or all of the messages:

Message identifier	By supplying a unique value in this field, the returning message will have the same value as the request. This makes it possible for the calling application to identify the reply.
Transaction Initiator	Described on the previous page. Indicates what type of system the initiator of a request was. Always R for auto-generated messages from Bartrack. Always blank for requests and replied messages.
Serial number	Most messages have the serial number field, because most of the messages are dealing with the serialized individuals in one way or another.
User	This value is used in the Bartrack log to log the action of the request messages.

More information on the kind of information that is intended to be put in the fields can be found in section "Types" on page 147.

AsmIDAck

Name AsmIDAck

Description Sent by Bartrack when the **AsmIDReq** message is received with a parent and a child serial number that is valid.

Req. Bartrack version 6.0 -

DmQ message Type 12059

DmQ message class: 12000

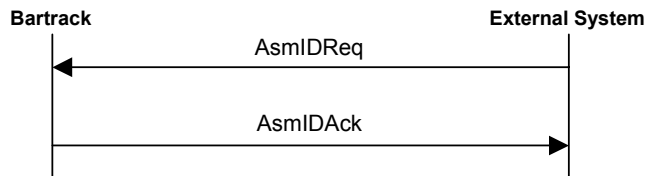
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



If an error occurs, **AsmIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

AsmIDErr

Name AsmIDErr

Description Sent by Bartrack when the **AsmIDReq** message is received with a parent or child serial number that is not valid.

Req. Bartrack version 6.0 -

DmQ message Type 12058

DmQ message class: 12000

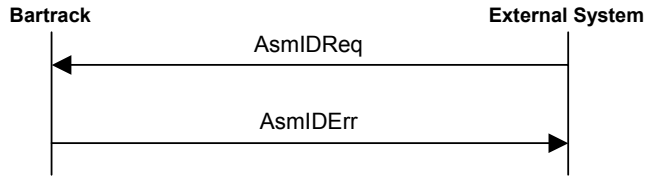
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

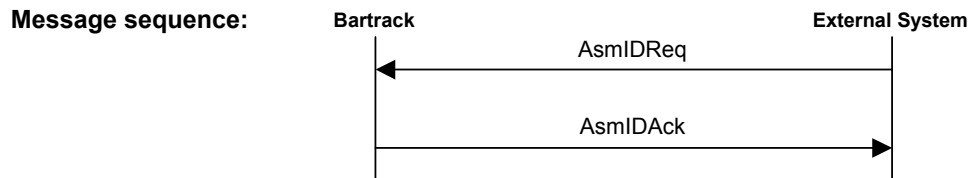
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

AsmIDReq

Name AsmIDReq
Description Request to add or remove a child to a parent.
Req. Bartrack version 6.0 -
DmQ message Type 12060
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Fixed, 255 bytes
Transaction: SERPCA: Add a child to the parent (assemble or connect)
 SERPCD: Delete a child from the parent (disassemble or disconnect)



If an error occurs, **AsmIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERPCA or SERPCD.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Parent	Identity_20	Y	The parent serial number.
Child	Identity_20	Y	The child serial number.
User	User_12		User-ID of the operator.
Soft disconnect	YesNo_1	Y	Y: Soft. The ASSEMBLE_DONE field in Bartrack is regarded. If it is "Y", the disconnect fails. N: Hard. The ASSEMBLE_DONE field in Bartrack is disregarded.
Assemble done	YesNo_1	Y	Y: The ASSEMBLE_DONE field in Bartrack is set to Y (yes). N: The ASSEMBLE_DONE field in Bartrack is set to N (no).
Structure check	YesNo_1	Y	Y: The structure of the parent must be compatible with the product structure in Bartrack. N: The structure of the parent is allowed to have missing children. N.B. If structure check is disabled, it will not be possible to assemble children conflicting with the product structure in Bartrack.
Filler	Filler_169		To fill up message length (169 characters).

AsmInfo

Name AsmInfo

Description The structure has changed for the parent. One or more children have been added or removed in the structure. All children in the new structure are included in the message.

Req. Bartrack version 6.0 -

DmQ message Type 12050

DmQ message class: 12000

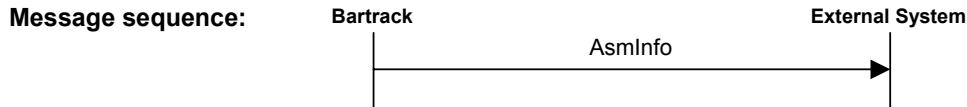
DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Varying, 154 + 20 x [No of Children]
Maximum 999 children means that the maximum size is 20.134 bytes.

N.B. The maximum number of children in Bartrack may be higher or lower.

Transaction: SERPCA



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERPCA.
Initiator	Initiator_1	Y	Indicates what type of system the initiator of a request was. Always R.
Serial number	Identity_20	Y	The parent serial number.
Order number	OrdNum_35	Y	Order number of the parent.
Order item	OrdItem_4	Y	Order number item of the parent.
Product number	ProdNum_24	Y	Product number of the parent.
R-state	R_State_7	Y	R-state of the parent.
Comment	Comment_30		Free text.
No of Children	Numeric_3	Y	The number of children following this field.
Child Serial No[1]	Identity_20		1:st child.
Child Serial No[2]	Identity_20		2:nd child.
Child Serial No[n]	Identity_20		n:th child.

AsmInfo_1

Name AsmInfo_1

Description The structure has changed for the parent. One or more children have been added or removed in the structure. All children in the new structure are included in the message.

Req. Bartrack version 6.4 -

DmQ message Type 12119

DmQ message class: 12000

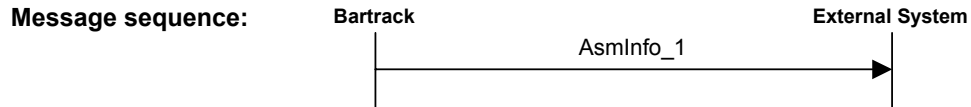
DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Varying, 184 + 20 x [No of Children]
Maximum 999 children means that the maximum size is 20.164 bytes.

N.B. The maximum number of children in Bartrack may be higher or lower.

Transaction: SERPCA



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERPCA.
Initiator	Initiator_1	Y	Indicates what type of system the initiator of a request was. Always R.
Serial number	Identity_20	Y	The parent serial number.
Order number	OrdNum_35	Y	Order number of the parent.
Order item	OrdItem_4	Y	Order number item of the parent.
Product number	ProdNum_24	Y	Product number of the parent.
R-state	R_State_7	Y	R-state of the parent.
Comment	Comment_30		Free text.
TPComment	TPComment_30		Test Production Comment.
No of Children	Numeric_3	Y	The number of children following this field.
Child Serial No[1]	Identity_20		1:st child.
Child Serial No[2]	Identity_20		2:nd child.
Child Serial No[n]	Identity_20		n:th child.

BarUnknown

Name BarUnknown

Description Sent by Bartrack when a message is received and the request is not understood. That is, the message type and/or class are not defined in the BARSFT interface description.

Req. Bartrack version 6.2 -

DmQ message Type 12093

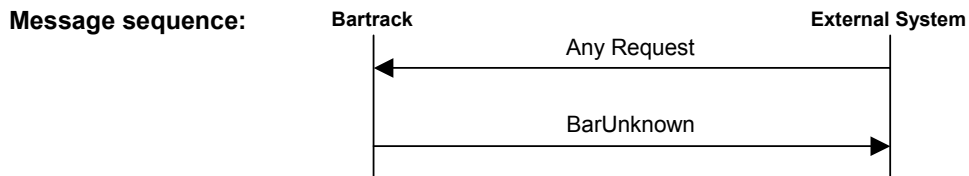
DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed 255 bytes

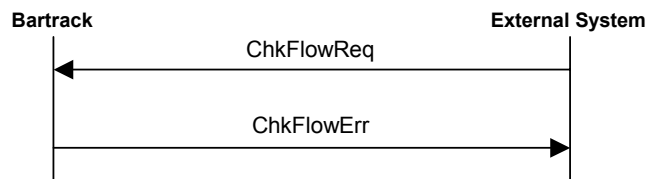
Transaction: RTNMSG



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

ChkFlowErr

Name ChkFlowErr
Description Sent by Bartrack when the ChkFlowReq failed.
Req. Bartrack version 6.5 -
DmQ message Type 12121
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

ChkFlowInfo

Name ChkFlowInfo

Description Result of the request to check if an update of the test status for an individual is allowed according to the defined production flow.

Req. Bartrack version 6.5 -

DmQ message Type 12122

DmQ message class: 12000

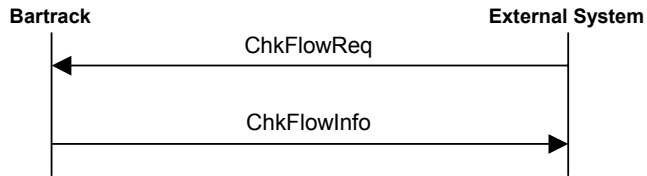
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



If an error occurs, **ChkFlowErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Result	YesNo_1	Y	Y = OK to update test status N = Not OK to update test status
Filler	Filler_223		To fill up message length (223 characters).

ChkFlowReq

Name ChkFlowReq

Description Request to check if an update of the test status for an individual is allowed according to the defined production flow.

Req. Bartrack version 6.5 -

DmQ message Type 12120

DmQ message class: 12000

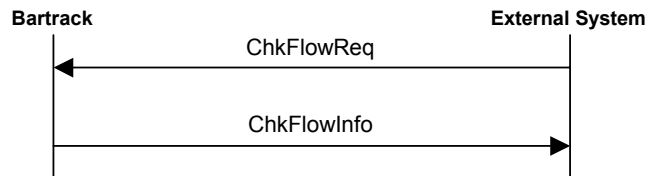
DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Fixed, 255 bytes

Transaction: CHKTST

Message sequence:

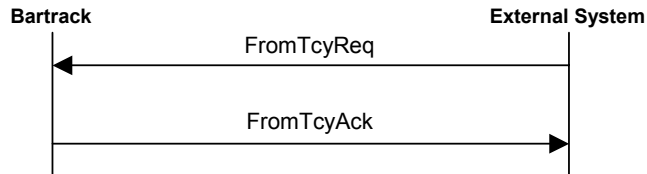


If an error occurs, **ChkFlowErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	CHKTST.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number.
Test id	Test_Id_6	Y	The test station identity.
User	User_12		User-ID of the operator.
Filler	Filler_186		To fill up message length (186 characters).

FromTcyAck

- Name** FromTcyAck
- Description** Sent by Bartrack when the request for individual data from the traceability system has been successfully delivered.
- Req. Bartrack version** 6.6 -
- DmQ message Type** 12125
- DmQ message class:** 12000
- DmQ message priority:** Low
- DmQ interest point:** MEM
- Length:** Fixed, 255 bytes
- Transaction:** RTNMSG
- Message sequence:**

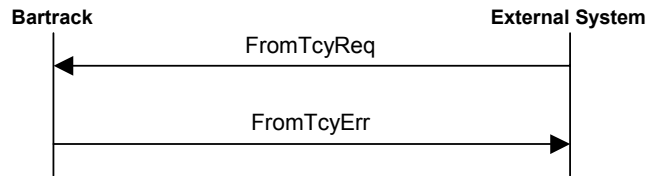


If an error occurs, **FromTcyErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

FromTcyErr

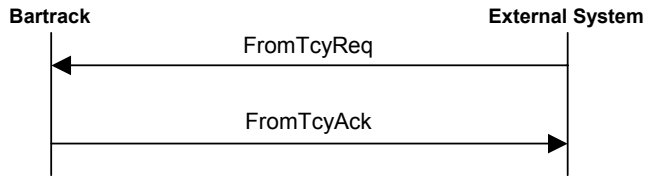
Name FromTcyErr
Description Sent by Bartrack when the request for individual data from the traceability system has failed.
Req. Bartrack version 6.6 -
DmQ message Type 12124
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

FromTcyReq

Name FromTcyReq
Description Request to send a request for individual data from the traceability system.
Req. Bartrack version 6.6 -
DmQ message Type 12123
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Fixed, 255 bytes
Transaction: TCYREQ
Message sequence:

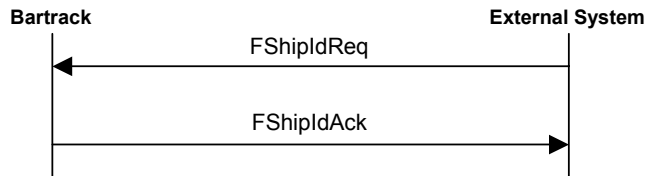


If an error occurs, **FromTcyErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always TCYREQ.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	Serial number of the individual.
Individual data	Yes_No_1		Y = Request for individual data. Other = Do not request for individual data.
Additional data	Yes_No_1		Y = Request for additional numbers. Other = Do not request for additional numbers.
Retail data	Yes_No_1		Y = Request for retail products. Other = Do not request for retail products.
No of levels	Alphanum_1		1,2,3 = Request structure information for 1, 2 or 3 levels. A = Request structure information for all levels. Other = Do not request for structure information.
User	User_12		User-ID of the operator.
Filler	Filler_188		To fill up message length (188 characters).

FShipIdAck

Name FShipIdAck
Description Sent by Bartrack when the **FShipIdReq** message is received with a valid serial number.
Req. Bartrack version 6.3.1 -
DmQ message Type 12105
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:

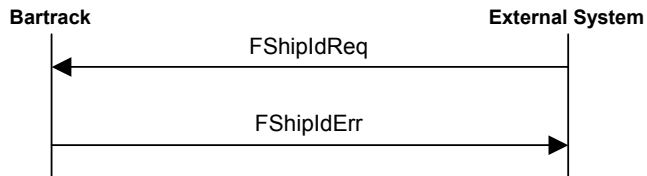


If an error occurs, **FShipIdErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

FShipIdErr

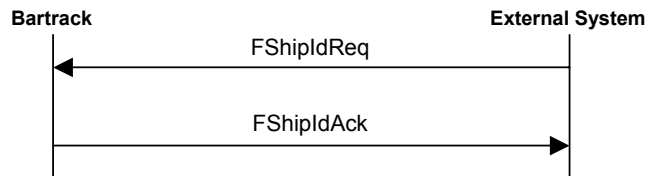
Name FShipIdErr
Description Sent by Bartrack when the **FShipIdReq** message is received with an illegal serial number of individual.
Req. Bartrack version 6.3.1 -
DmQ message Type 12104
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

FShipIdReq

Name FShipIdReq
Description Request to force ship an individual.
Req. Bartrack version 6.3.1 -
DmQ message Type 12103
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Fixed, 255 bytes
Transaction: SERFSH
Message sequence:



If an error occurs, **FShipIdErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERFSH.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	Serial number of the individual.
Destination	Destination_6		The shipping destination in Bartrack.
Label	Label_22		Label name in Bartrack.
No of Labels	Numeric_3		Number of label copies to print.
Printer	Printer_22		Printer name in Bartrack.
User	User_12		User-ID of the operator.
Filler	Filler_139		To fill up message length (139 characters).

GenAddAck

Name GenAddAck

Description Sent by Bartrack when the **GenAddReq** message is received with a valid serial number.

Req. Bartrack version 6.6 -

DmQ message Type 12131

DmQ message class: 12000

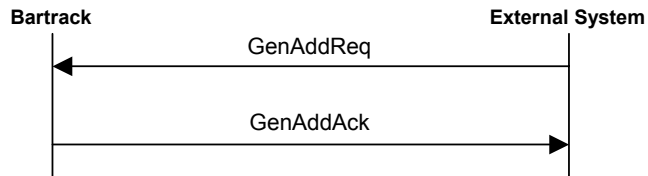
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:

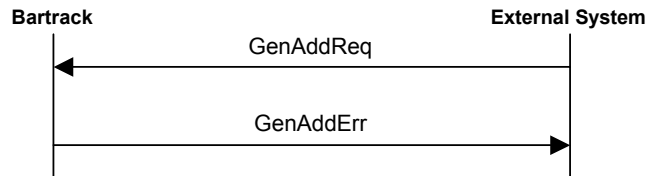


If an error occurs, **GenAddErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Warning message	Text_80		Warning message from Bartrack.
Additional warning message data	Text_20		Additional warning message data.
Filler	Filler_124		To fill up message length (124 characters).

GenAddErr

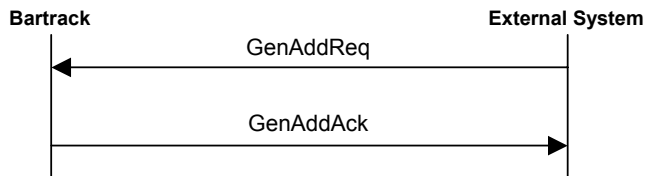
Name GenAddErr
Description Sent by Bartrack when the **GenAddReq** message is received with an illegal serial number of individual.
Req. Bartrack version 6.6 -
DmQ message Type 12130
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

GenAddReq

Name GenAddReq
Description Request to generate an additional number.
Req. Bartrack version 6.6 -
DmQ message Type 12129
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: GENADD
Message sequence:



If an error occurs, **GenAddErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always GENADD.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	Serial number of the individual.
Position	Pos_10		Position.
User	User_12		User-ID of the operator.
Filler	Filler_182		To fill up message length (182 characters).

GetAddInfo

Name GetAddInfo

Description Sent by Bartrack when the **GetAddReq** message is received with a valid serial number.

Req. Bartrack version 6.6 -

DmQ message Type 12128

DmQ message class: 12000

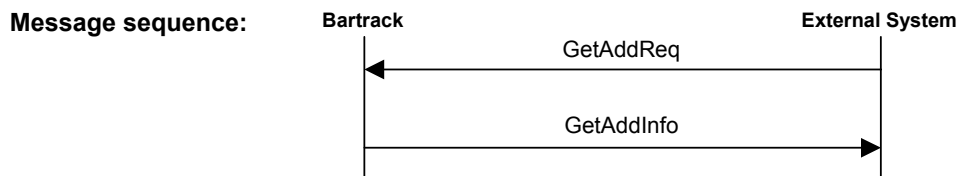
DmQ message priority: Low

DmQ interest point: MEM

Length: Varying, 54 + 74 x [No of addnos]
 Maximum 128 additional numbers means that the maximum size is 9.526 bytes.

N.B. The maximum number of additional numbers in Bartrack may be higher or lower.

Transaction: RTNMSG



If an error occurs, **GetAddErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	Serial number of the individual.
No of addnos	Numeric_3	Y	Number of additional numbers.
Position[1]	Pos_10		Position 1.
Family[1]	Family_12		Family 1.
Type[1]	Type_20		Type 1.
Description[1]	Type_32		Description 1.
Position[2]	Pos_10		Position 2.
Family[2]	Family_12		Family 2.
Type[2]	Type_20		Type 2.
Description[2]	Type_32		Description 2.
Position[n]	Pos_10		Position n.
Family[n]	Family_12		Family n.
Type[n]	Type_20		Type n.
Description[n]	Type_32		Description n.

GetAddErr

Name GetAddErr

Description Sent by Bartrack when the **GetAddReq** message is received with an illegal serial number of individual.

Req. Bartrack version 6.6 -

DmQ message Type 12127

DmQ message class: 12000

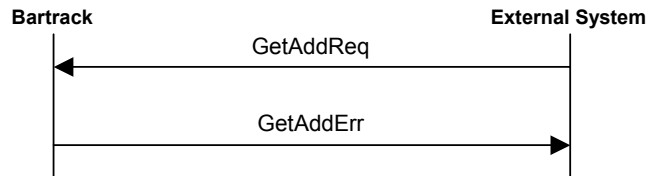
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

GetAddReq

Name GetAddReq

Description Request to fetch data of all defined but not generated additional numbers.

Req. Bartrack version 6.6 -

DmQ message Type 12126

DmQ message class: 12000

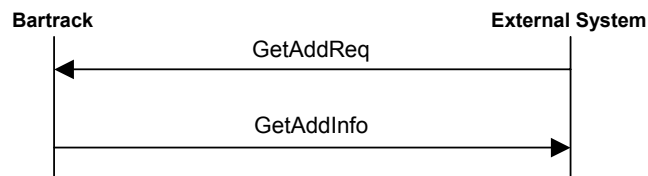
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: NOTGEN

Message sequence:

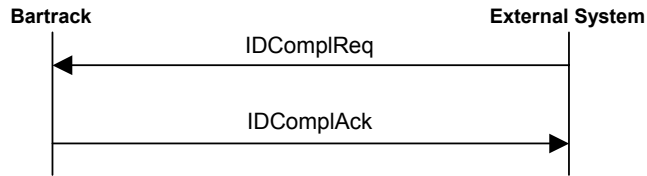


If an error occurs, **GetAddErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always NOTGEN.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	Serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

IDComplAck

Name	IDComplAck
Description	Sent by Bartrack when the IDComp1Req message is received with a valid serial number.
Req. Bartrack version	6.2 -
DmQ message Type	12082
DmQ message class:	12000
DmQ message priority:	Low
DmQ interest point:	MEM
Length:	Fixed, 255 bytes
Transaction:	RTNMSG
Message sequence:	



If an error occurs, **IDComp1Err** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

IDComplErr

Name IDComplErr

Description Sent by Bartrack when a **IDComplReq** is received with for a non-existing serial number.

Req. Bartrack version 6.2 -

DmQ message Type 12083

DmQ message class: 12000

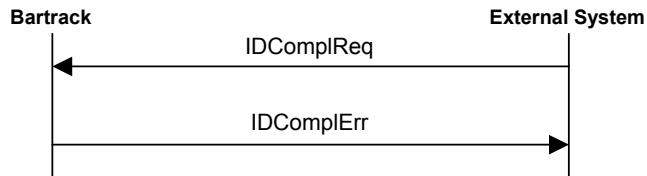
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

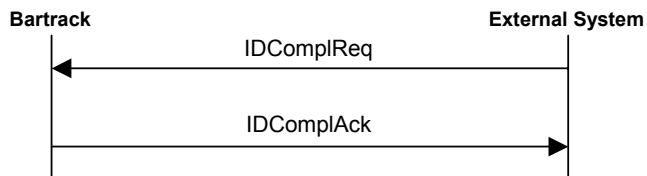
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

IDComplReq

Name IDComplReq
Description Request to check the completeness of an individual structure.
Req. Bartrack version 6.2 -
DmQ message Type 12081
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Fixed, 255 bytes
Transaction: SERCPL
Message sequence:



If an error occurs, **IDComplErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERCPL.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

IDErr

Name IDErr

Description Sent by Bartrack when an **IDReq** is received with for a non-existing serial number.

Req. Bartrack version 6.0 -

DmQ message Type 12051

DmQ message class: 12000

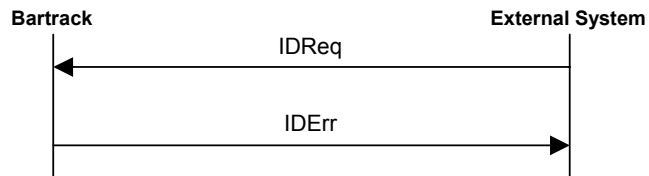
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

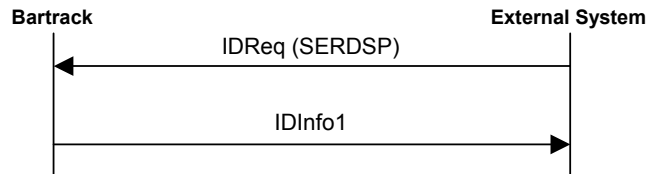
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

IDInfo1

Name IDInfo1
Description Basic information about a requested individual. This is the reply of an **IDReq**.
Req. Bartrack version 6.0 -
DmQ message Type 12052
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



If an error occurs, **IDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Order number	OrdNum_35	Y	Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Product name	ProdName_25		Name of the product.
Parent serial number	Identity_20		The current parent serial number.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Status	IndStatus_2	Y	The manufacturing status for the individual.
Created date	Date_10		The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Test status	Test_status_2	Y	Test status, only the first position is relevant: "P"= Passed, "F"= Failed, "N"=Not tested.
No of req. Children	Numeric_3		Number of required children.
No of current children	Numeric_3		Current number of children.
Comment	Comment_30		Free text.
Filler	Filler_29		To fill up message length (29 characters).

IDInfo2

Name IDInfo2

Description Basic information about a requested individual. This is the reply of the **IDReq** message.

Req. Bartrack version 6.0 -

DmQ message Type 12053

DmQ message class: 12000

DmQ message priority: Low

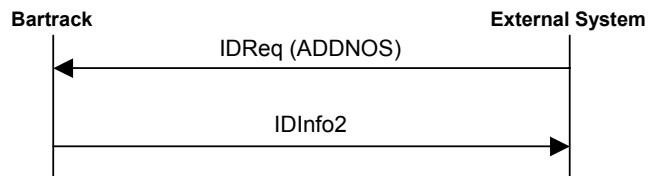
DmQ interest point: MEM

Length: Varying, 54 + 30 x [No of addnos]
 Maximum 128 additional numbers means that the maximum size is 3.894 bytes.

N.B. The maximum number of additional numbers in Bartrack may be higher or lower.

Transaction: RTNMSG

Message sequence:

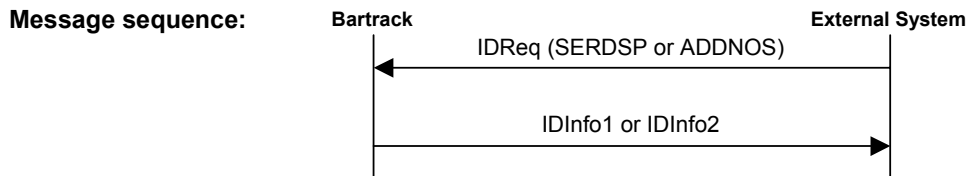


If an error occurs, **IDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
No of addnos	Numeric_3		Number of additional numbers.
Additional number[1]	Addno_20		Additional number 1.
Position[1]	Pos_10		Position 1.
Additional number[2]	Addno_20		Additional number 2.
Position[2]	Pos_10		Position 2.
Additional number[n]	Addno_20		Additional number n.
Position[n]	Pos_10		Position n.

IDReq

Name IDReq
Description The external system requests the information of the specified individual.
Req. Bartrack version 6.0 -
DmQ message Type 12054
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERDSP: The message IDInfo1 is replied
 ADDNOS: The message IDInfo2 is replied (includes additional numbers)

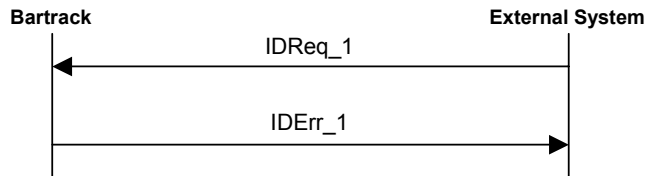


If an error occurs, **IDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERDSP or ADDNOS.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

IDErr_1

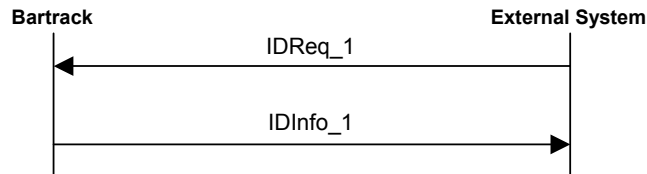
Name IDErr_1
Description Sent by Bartrack when an **IDReq_1** is received with for a non-existing serial number.
Req. Bartrack version 6.4 -
DmQ message Type 12113
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

IDInfo_1

Name IDInfo_1
Description Basic information about a requested individual. This is the reply of an **IDReq_1**.
Req. Bartrack version 6.4 -
DmQ message Type 12114
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 511 bytes
Transaction: RTNMSG
Message sequence:



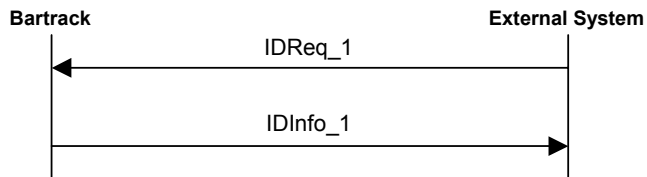
If an error occurs, **IDErr_1** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Order number	OrdNum_35		Order number of the individual.
Order item	OrdItem_4		Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Product name	ProdName_25		Name of the product.
Parent serial number	Identity_20		The current parent serial number.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Status	IndStatus_2	Y	The manufacturing status for the individual.
Created date	Date_10	Y	The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Test status	Test_status_2	Y	Test status, only the first position is relevant: "P"= Passed, "F"= Failed, "N"=Not tested.
No of req. Children	Numeric_3	Y	Number of required children.
No of current children	Numeric_3	Y	Current number of children.
Comment	Comment_30		Free text.
Exemption	Exemption_1	Y	Exemption. (Y,N,U)

TPC_flag	TPC_flag_1	Y	C = Created I = Inherited N = No Test Production comment.
TPComment	TPComment_30		Test Production Comment. Only valid if TPC_flag is C or I
Filler	Filler_253		To fill up message length (253 characters).

IDReq_1

Name IDReq_1
Description The external system requests the information of the specified individual.
Req. Bartrack version 6.4 -
DmQ message Type 12112
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERDSP
Message sequence:



If an error occurs, **IDErr_1** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERDSP.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

IDErr_2

Name IDErr_2

Description Sent by Bartrack when an **IDReq_2** is received with for a non-existing serial number.

Req. Bartrack version 6.6 -

DmQ message Type 12133

DmQ message class: 12000

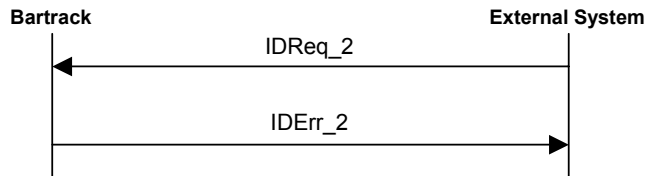
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

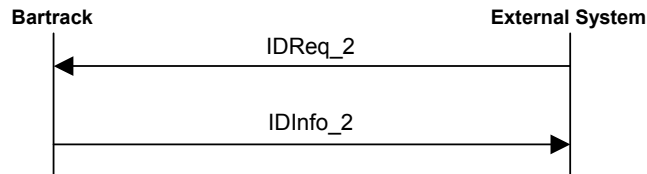
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

IDInfo_2

Name	IDInfo_2
Description	Basic information about a requested individual. This is the reply of an IDReq_2 .
Req. Bartrack version	6.6 -
DmQ message Type	12134
DmQ message class:	12000
DmQ message priority:	Low
DmQ interest point:	MEM
Length:	Fixed, 511 bytes
Transaction:	RTNMSG
Message sequence:	



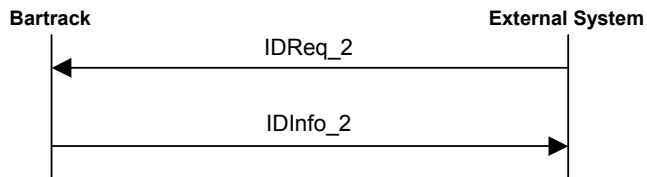
If an error occurs, **IDErr_2** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Order number	OrdNum_35		Order number of the individual.
Order item	OrdItem_4		Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Product name	ProdName_25		Name of the product.
Parent serial number	Identity_20		The current parent serial number.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Status	IndStatus_2	Y	The manufacturing status for the individual.
Created date	Date_10	Y	The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Test status	Test_status_2	Y	Test status, only the first position is relevant: "P"= Passed, "F"= Failed, "N"=Not tested.
No of req. Children	Numeric_3	Y	Number of required children.
No of current children	Numeric_3	Y	Current number of children.
Comment	Comment_30		Free text.
Exemption	Exemption_1	Y	Exemption. (Y,N,U)

TPC_flag	TPC_flag_1	Y	C = Created I = Inherited N = No Test Production comment.
TPComment	TPComment_30		Test Production Comment. Only valid if TPC_flag is C or I
Information	Information_35		Information.
Filler	Filler_218		To fill up message length (218 characters).

IDReq_2

Name IDReq_2
Description The external system requests the information of the specified individual.
Req. Bartrack version 6.6 -
DmQ message Type 12132
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERDSP
Message sequence:



If an error occurs, **IDErr_2** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERDSP.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

IDStrErr_1

Name IDStrErr_1

Description Sent by Bartrack when an **IDStrReq_1** is received with for a non-existing serial number.

Req. Bartrack version 6.4 -

DmQ message Type 12116

DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

IDStrInfo_1

Name IDStrInfo_1

Description Basic information about a child of the requested individual. This is the reply of an **IDStrReq_1**.
 One message is sent for each child
 No more individuals = serial number is filled with space.

Req. Bartrack version 6.4 -

DmQ message Type 12117

DmQ message class: 12000

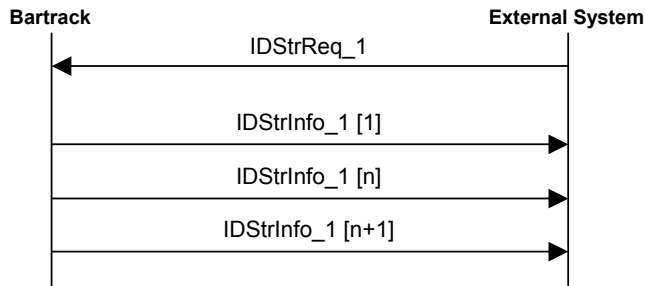
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 511 bytes

Transaction: RTNMSG

Message sequence:



Where n is the number of current children.

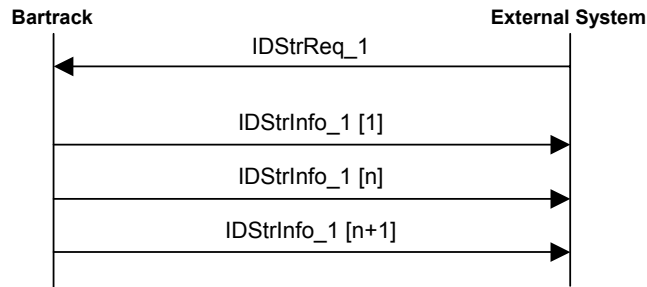
If an error occurs, **IDErr_1** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual. Filled with space = No more children found.
Order number	OrdNum_35		Order number of the individual.
Order item	OrdItem_4		Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Product name	ProdName_25		Name of the product.
Parent serial number	Identity_20		The current parent serial number.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Status	IndStatus_2	Y	The manufacturing status for the individual.
Created date	Date_10	Y	The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).

No of Addnos	Numeric_3	Y	The number of additional numbers.
Test status	Test_status_2	Y	Test status, only the first position is relevant: "P"= Passed "F"= Failed "N"=Not tested.
No of req. Children	Numeric_3	Y	Number of required children.
No of current children	Numeric_3	Y	Current number of children.
Comment	Comment_30		Free text.
Exemption	Exemption_1	Y	Exemption. (Y,N,U)
TPC_flag	TPC_flag_1	Y	C = Created I = Inherited N = No Test Production comment.
TPComment	TPComment_30		Test Production Comment. Only valid if TPC_flag is C or I.
Filler	Filler_253		To fill up message length (253 characters).

IDStrReq_1

Name IDStrReq_1
Description The External system requests the information about children (1 level) for the specified parent
Req. Bartrack version 6.4 -
DmQ message Type 12115
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERSTP
Message sequence:



Where n is the number of current children.

If an error occurs, **IDStrErr_1** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERSTP.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

IDStrErr_2

Name IDStrErr_2

Description Sent by Bartrack when an **IDStrReq_2** is received with for a non-existing serial number.

Req. Bartrack version 6.6 -

DmQ message Type 12136

DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

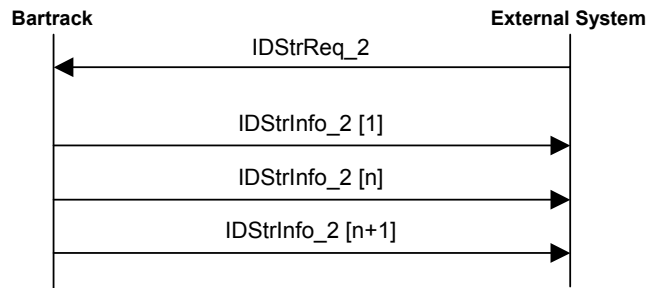
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

IDStrInfo_2

Name	IDStrInfo_2
Description	Basic information about a child of the requested individual. This is the reply of an IDStrReq_2 . One message is sent for each child No more individuals = serial number is filled with space.
Req. Bartrack version	6.6 -
DmQ message Type	12137
DmQ message class:	12000
DmQ message priority:	Low
DmQ interest point:	MEM
Length:	Fixed, 511 bytes
Transaction:	RTNMSG
Message sequence:	



Where n is the number of current children.

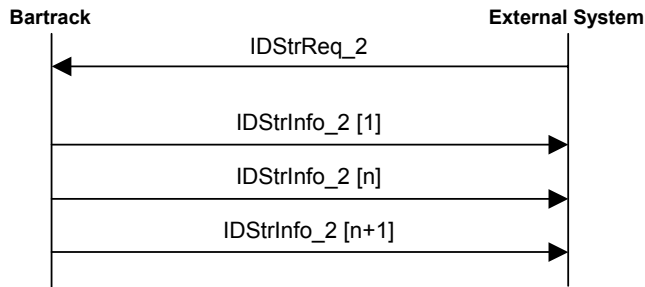
If an error occurs, **IDErr_2** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual. Filled with space = No more children found.
Order number	OrdNum_35		Order number of the individual.
Order item	OrdItem_4		Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Product name	ProdName_25		Name of the product.
Parent serial number	Identity_20		The current parent serial number.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Status	IndStatus_2	Y	The manufacturing status for the individual.
Created date	Date_10	Y	The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).

No of Addnos	Numeric_3	Y	The number of additional numbers.
Test status	Test_status_2	Y	Test status, only the first position is relevant: "P"= Passed "F"= Failed "N"=Not tested.
No of req. Children	Numeric_3	Y	Number of required children.
No of current children	Numeric_3	Y	Current number of children.
Comment	Comment_30		Free text.
Exemption	Exemption_1	Y	Exemption. (Y,N,U)
TPC_flag	TPC_flag_1	Y	C = Created I = Inherited N = No Test Production comment.
TPComment	TPComment_30		Test Production Comment. Only valid if TPC_flag is C or I.
Information	Information_35		Information.
Filler	Filler_218		To fill up message length (218 characters).

IDStrReq_2

Name IDStrReq_2
Description The External system requests the information about children (1 level) for the specified parent
Req. Bartrack version 6.6 -
DmQ message Type 12135
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERSTP
Message sequence:



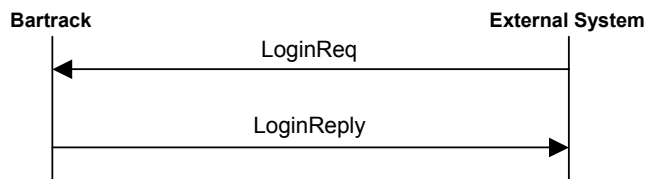
Where n is the number of current children.

If an error occurs, **IDStrErr_2** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERSTP.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

LoginReply

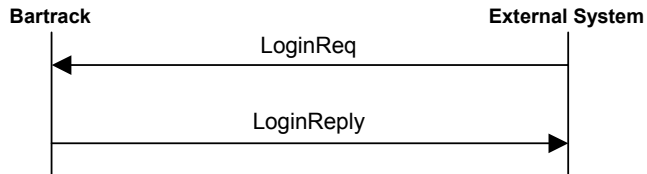
Name LoginReply
Description Sent by Bartrack to report whether the user-ID and/or password were valid.
Req. Bartrack version 6.0 -
DmQ message Type 12041
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Login success	YesNo_1	Y	The result of the login. Y: The user-ID/password was OK. N: The user-ID/password doesn't match.
Message	Text_80		If the request was unsuccessful, this message contains the error.
Filler	Filler_143		To fill up message length (143 characters).

LoginReq

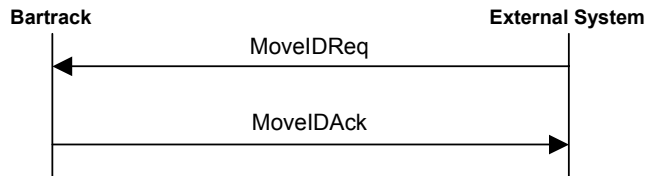
Name LoginReq
Description Sent by an external system to request verify the user-ID and password to login to Bartrack.
Req. Bartrack version 6.0 -
DmQ message Type 12040
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: BLOGIN
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always BLOGIN
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
User-ID	User_12		The user-ID of the operator.
Password	Password_20		The password for the user-ID.
Filler	Filler_192		To fill up message length (192 characters).

MoveIDAck

Name MoveIDAck
Description Sent by Bartrack when the **MoveIDReq** message is received with a valid serial number.
Req. Bartrack version 6.0 -
DmQ message Type 12063
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



If an error occurs, **MoveIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

MoveIDErr

Name MoveIDErr

Description Sent by Bartrack when the **MoveIDReq** message is received with an illegal serial number of individual.

Req. Bartrack version 6.0 -

DmQ message Type 12062

DmQ message class: 12000

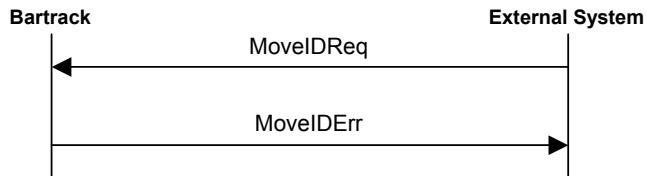
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

MoveIDReq

Name MoveIDReq

Description Request to store, ship or scrap an individual. The subtype determines the type of request.

Req. Bartrack version 6.0 -

DmQ message Type 12064

DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

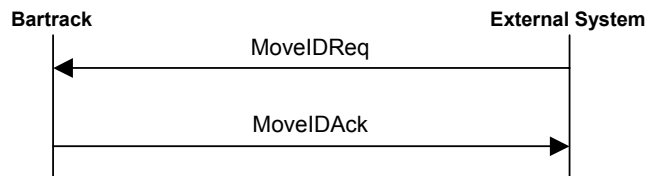
Length: Fixed, 255 bytes

Transaction: SERMVE

Subtype The field "Action" shall contain one of the following values:
 ST: Store the individual
 SH: Ship the individual
 SC: Scrap the individual

Note: If a unit is scrapped, it's parent and children are disconnected before the scrapping. If a whole structure is to be scrapped, it is the responsibility of the requesting application to scrap all individuals intended to scrap.

Message sequence:

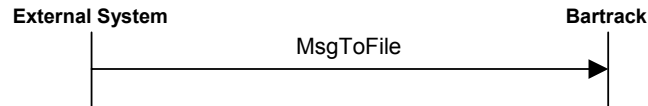


If an error occurs, **MoveIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERMVE.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	Serial number of the individual.
Action	Action_2	Y	ST: Store SH: Ship SC: Scrap
Destination	Destination_6		The shipping/storing destination in Bartrack.
Label	Label_22		Label name in Bartrack.
No of Labels	Numeric_3		Number of label copies to print.
Printer	Printer_22		Printer name in Bartrack.
User	User_12		User-ID of the operator.
Filler	Filler_137		To fill up message length (137 characters).

MsgToFile

Name MsgToFile
Description Request to create a file in a given directory with a given filename.
Req. Bartrack version 6.2 -
DmQ message Type 12090
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Varying, 24 + 255 + contents length
Transaction:
Subtype
Message sequence:



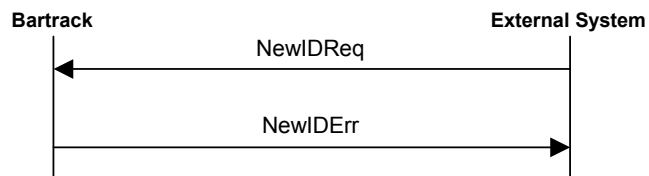
Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Path and filename	Text_255	Y	The OpenVMS path and filename to use. For example: BAR_ORDER_DIR:021123.ORD
Contents	Text_x		Contents to be put in the file. May be zero length.

This message is a bit different from the other messages. No error handling or return messages exist. The only purpose of this message is to create a file with the content of the message.

Typical use for this message is to receive information from an external system with DmQ and transform it to a file for the file interfaces of Bartrack. An example of such a file interface is the order import from order systems, or Tracy files from Tracy.

NewIDErr

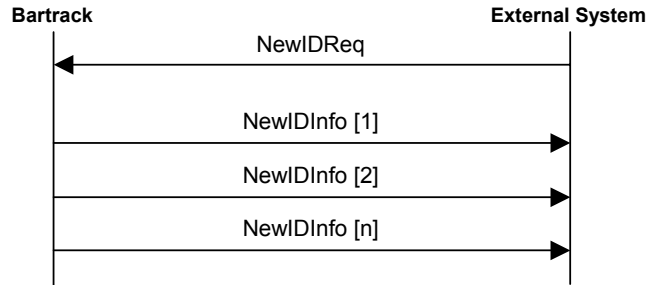
Name NewIDErr
Description Sent by Bartrack when a **NewIDReq** is received with invalid information.
Req. Bartrack version 6.0 -
DmQ message Type 12065
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

NewIDInfo

Name NewIDInfo
Description Sent by Bartrack when the **NewIDReq** message is received with valid information.
Req. Bartrack version 6.0 -
DmQ message Type 12066
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:

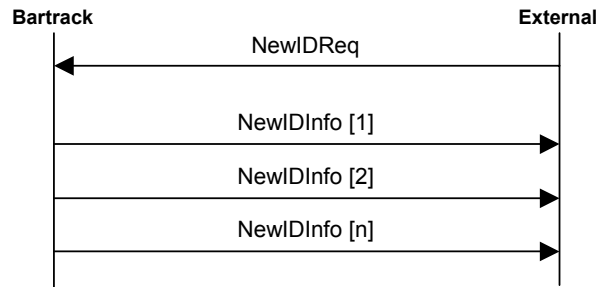


Where n is the number of requested number of individuals.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20		Serial number of the individual.
Order number	OrdNum_35	Y	Order number of the parent.
Order item	OrdItem_4	Y	Order number item of the parent.
Product Number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Created date	Date_10		The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Filler	Filler_120		To fill up message length (120 characters).

NewIDReq

Name NewIDReq
Description A request to generate one or several serial numbers.
Req. Bartrack version 6.0 -
DmQ message Type 12067
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERCRT
Message sequence:



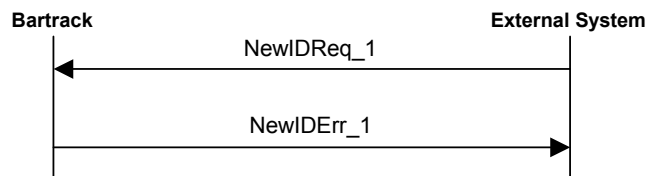
Where n is the number of requested number of individuals.
 If an error occurs, **NewIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERCRT.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35		Order number of the parent. If this value is given, the orders product number and revision will be used.
Order item	OrdItem_4		Order number item of the parent. If this value is given, the orders product number and revision will be used.
Product Number	ProdNum_24	Y	Product number of the individual. If order number is supplied, this value has no effect.
Revision State	R_State_7	Y	R-state of the individual. If order number is supplied, this value has no effect.
Comment	Comment_30		Free text.
Exemption	YesNo_1		If the individual is produced with an exemption.
Req number	Numeric_3		Requested number of serial numbers. If order number is supplied, this value has no effect.
Label name	Label_22		Label name in Bartrack.
Label copies	Numeric_3		Number of copies of labels.
Printer	Printer_22		Printer name in Bartrack.
User	User_12		User-ID of the operator.

Filler	Filler_61		To fill up message length (61 characters).
--------	-----------	--	--

NewIDErr_1

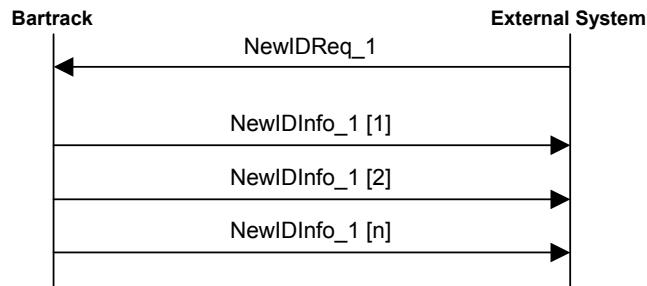
Name NewIDErr_1
Description Sent by Bartrack when a **NewIDReq_1** is received with invalid information.
Req. Bartrack version 6.4 -
DmQ message Type 12107
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

NewIDInfo_1

Name NewIDInfo_1
Description Sent by Bartrack when the **NewIDReq_1** message is received with valid information.
Req. Bartrack version 6.4 -
DmQ message Type 12108
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:

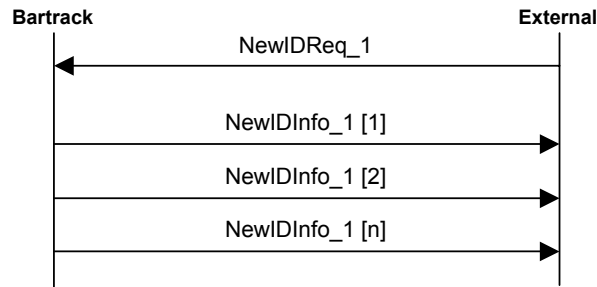


Where n is the number of requested number of individuals.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20		Serial number of the individual.
Order number	OrdNum_35	Y	Order number of the parent.
Order item	OrdItem_4	Y	Order number item of the parent.
Product Number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Created date	Date_10		The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Filler	Filler_120		To fill up message length (120 characters).

NewIDReq_1

Name NewIDReq_1
Description A request to generate one or several serial numbers.
Req. Bartrack version 6.4 -
DmQ message Type 12106
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERCRT
Message sequence:



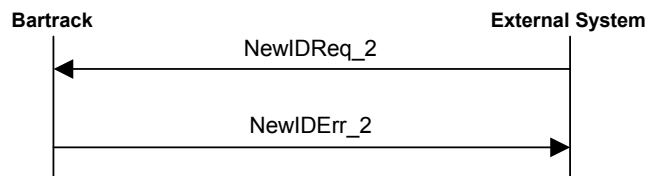
Where n is the number of requested number of individuals.
 If an error occurs, **NewIDErr_1** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERCRT.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35		Order number of the parent. If this value is given, the orders product number and revision will be used.
Order item	OrdItem_4		Order number item of the parent. If this value is given, the orders product number and revision will be used.
Product Number	ProdNum_24	Y	Product number of the individual. If order number is supplied, this value has no effect.
Revision State	R_State_7	Y	R-state of the individual. If order number is supplied, this value has no effect.
Comment	Comment_30		Free text.
Exemption	YesNo_1		If the individual is produced with an exemption.
Test production comment id	Numeric_6		= 0 - No test production. > 0 – Individual(s) is created with a test production comment. The test production comment id for a product or order is fetched with the TpcReq message
Req number	Numeric_3		Requested number of serial numbers. If order number is supplied, this value has no effect.

Label name	Label_22		Label name in Bartrack.
Label copies	Numeric_3		Number of copies of labels.
Printer	Printer_22		Printer name in Bartrack.
User	User_12		User-ID of the operator.
Filler	Filler_55		To fill up message length (55 characters).

NewIDErr_2

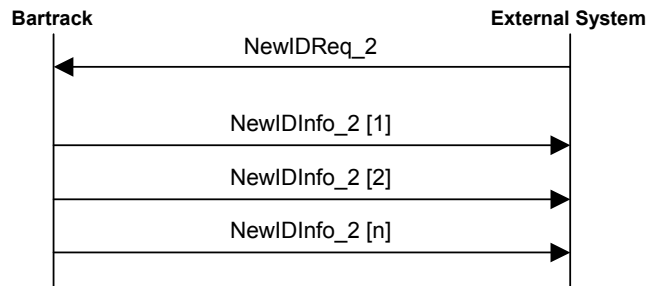
Name NewIDErr_2
Description Sent by Bartrack when a **NewIDReq_2** is received with invalid information.
Req. Bartrack version 6.6 -
DmQ message Type 12139
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

NewIDInfo_2

Name	NewIDInfo_2
Description	Sent by Bartrack when the NewIDReq_2 message is received with valid information.
Req. Bartrack version	6.6 -
DmQ message Type	12140
DmQ message class:	12000
DmQ message priority:	Low
DmQ interest point:	MEM
Length:	Fixed, 255 bytes
Transaction:	RTNMSG
Message sequence:	

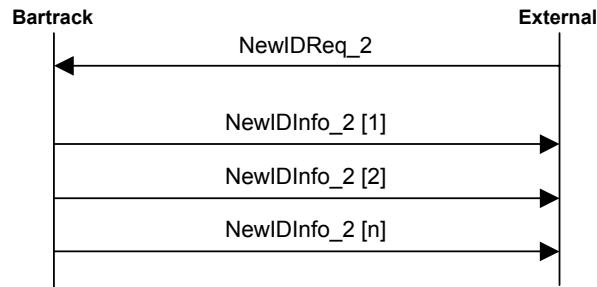


Where n is the number of requested number of individuals.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20		Serial number of the individual.
Order number	OrdNum_35	Y	Order number of the parent.
Order item	OrdItem_4	Y	Order number item of the parent.
Product Number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Created date	Date_10		The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Warning message	Text_80		Warning message from Bartrack.
Additional warning message data	Text_20		Additional warning message data.
Filler	Filler_20		To fill up message length (20 characters).

NewIDReq_2

Name NewIDReq_2
Description A request to generate one or several serial numbers.
Req. Bartrack version 6.6 -
DmQ message Type 12138
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: SERCRT
Message sequence:



Where n is the number of requested number of individuals.
 If an error occurs, **NewIDErr_2** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERCRT.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35		Order number of the parent. If this value is given, the orders product number and revision will be used.
Order item	OrdItem_4		Order number item of the parent. If this value is given, the orders product number and revision will be used.
Product Number	ProdNum_24	Y	Product number of the individual. If order number is supplied, this value has no effect.
Revision State	R_State_7	Y	R-state of the individual. If order number is supplied, this value has no effect.
Comment	Comment_30		Free text.
Exemption	YesNo_1		If the individual is produced with an exemption.
Test production comment id	Numeric_6		= 0 - No test production. > 0 – Individual(s) is created with a test production comment. The test production comment id for a product or order is fetched with the TpcReq message
Information	Information_35		Information.
Req number	Numeric_3		Requested number of serial numbers. If order number is

		supplied, this value has no effect.
Label name	Label_22	Label name in Bartrack.
Label copies	Numeric_3	Number of copies of labels.
Printer	Printer_22	Printer name in Bartrack.
User	User_12	User-ID of the operator.
Filler	Filler_20	To fill up message length (20 characters).

NewInfo

Name NewInfo

Description A new identity has been entered in Bartrack. The serial number is either created or registered manually.

Req. Bartrack version 6.0 -

DmQ message Type 12061

DmQ message class: 12000

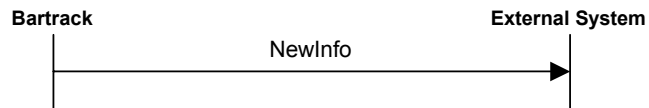
DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Fixed, 255 bytes

Transaction: SERCRT: The serial number was created
SEREXT: The serial number was registered manually (not registered from Tracy)

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERCRT or SEREXT.
Initiator	Initiator_1	Y	Indicates what type of system the initiator of a request was. Always R.
Serial number	Identity_20	Y	The serial number of the individual.
Order number	OrdNum_35	Y	Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Created date	Date_10		The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Comment	Comment_30		Free text.
User	User_12		User-ID of the operator.
Filler	Filler_72		To fill up message length (72 characters).

NewInfo_1

Name NewInfo_1

Description A new identity has been entered in Bartrack. The serial number is either created or registered manually.

Req. Bartrack version 6.4 -

DmQ message Type 12118

DmQ message class: 12000

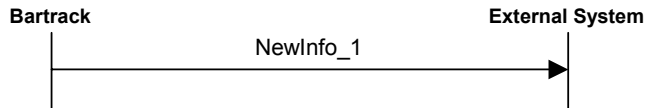
DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Fixed, 255 bytes

Transaction: SERCRT: The serial number was created
SEREXT: The serial number was registered manually (not registered from Tracy)

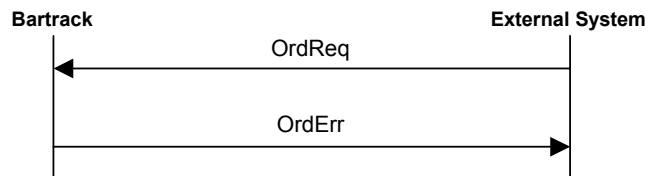
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERCRT or SEREXT.
Initiator	Initiator_1	Y	Indicates what type of system the initiator of a request was. Always R.
Serial number	Identity_20	Y	The serial number of the individual.
Order number	OrdNum_35	Y	Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Created date	Date_10		The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Addnos	Numeric_3	Y	The number of additional numbers.
Comment	Comment_30		Free text.
TPComment	TPComment_30		Test Production Comment.
User	User_12		User-ID of the operator.
Filler	Filler_42		To fill up message length (42 characters).

OrdErr

Name OrdErr
Description Sent by Bartrack when the **OrdReq** message is received with a non-existing order.
Req. Bartrack version 6.0 -
DmQ message Type 12078
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

OrdInfo

Name OrdInfo

Description Order information about a requested order.

Req. Bartrack version 6.0 -

DmQ message Type 12079

DmQ message class: 12000

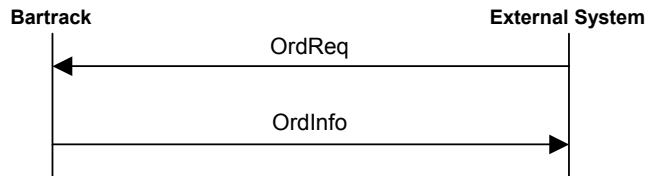
DmQ message priority: Low

DmQ interest point: MEM

Length: Varying, 280 + 44 x [No of children]
 Maximum 600 children mean that the maximum size is 26.680 bytes.
N.B. The maximum number of children in Bartrack may be higher or lower.

Transaction: RTNMSG

Message sequence:

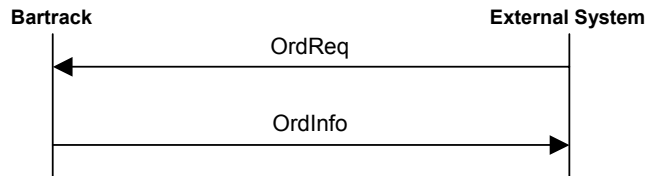


Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35	Y	Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7		R-state of the individual.
Used	YesNo_1	Y	Y = Order used N = Order not used
Text1	Text_35		Site-specific information.
Text2	Text_35		Site-specific information.
Text3	Text_35		Site-specific information.
Text4	Text_35		Site-specific information.
Text5	Text_35		Site-specific information.
No of children	Numeric_3	Y	Number of children in order structure.
Position[1]	Position_10	Y	The identifier for the order row.
Child Product number[1]	ProdNum_24	Y	Product number.
Child Revision State[1]	R_State_7		R-state.
Child quantity[1]	Numeric_3	Y	Quantity of units for the order row.
Position[2]	Position_10	Y	The identifier for the order row.
Child Product number[2]	ProdNum_24	Y	Product number.

Child Revision State[2]	R_State_7		R-state.
Child quantity[2]	Numeric_3	Y	Quantity of units for the order row.
Position[n]	Position_10	Y	The identifier for the order row.
Child Product number[n]	ProdNum_24	Y	Product number.
Child Revision State[n]	R_State_7		R-state.
Child quantity[n]	Numeric_3	Y	Quantity of units for the order row.

OrdReq

Name OrdReq
Description The External system requests the information about the specified order.
Req. Bartrack version 6.0 -
DmQ message Type 12080
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: ORDREQ
Message sequence:

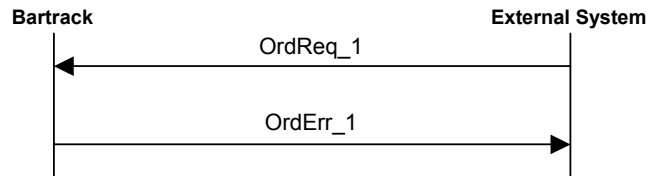


If an error occurs, **OrdErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always ORDREQ.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35	Y	Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_173		To fill up message length (173 characters).

OrdErr_1

Name OrdErr_1
Description Sent by Bartrack when the **OrdReq_1** message is received with a non-existing order.
Req. Bartrack version 6.3 -
DmQ message Type 12095
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

OrdInfo_1

Name OrdInfo_1

Description Order information about a requested order.

Req. Bartrack version 6.3 -

DmQ message Type 12096

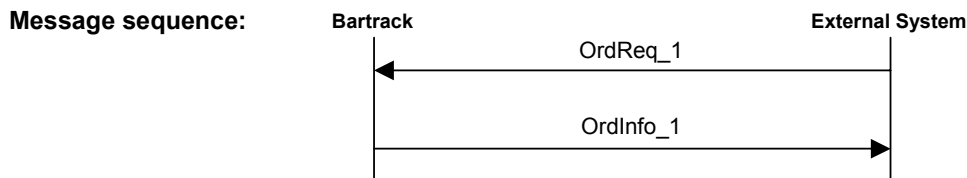
DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: MEM

Length: Varying, 315 + 44 x [No of children]
 Maximum 600 children mean that the maximum size is 26.715 bytes.
N.B. The maximum number of children in Bartrack may be higher or lower.

Transaction: RTNMSG

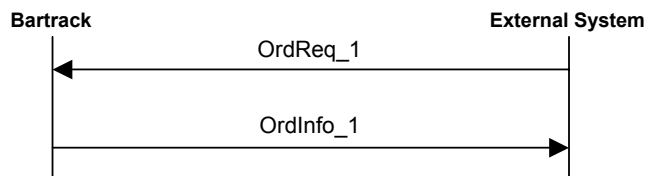


Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35	Y	Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7		R-state of the individual.
Used	YesNo_1	Y	Y = Order used N = Order not used
Text1	Text_35		Site-specific information.
Text2	Text_35		Site-specific information.
Text3	Text_70		Site-specific information.
Text4	Text_35		Site-specific information.
Text5	Text_35		Site-specific information.
No of children	Numeric_3	Y	Number of children in order structure.
Position[1]	Position_10	Y	The identifier for the order row.
Child Product number[1]	ProdNum_24	Y	Product number.
Child Revision State[1]	R_State_7		R-state.
Child quantity[1]	Numeric_3	Y	Quantity of units for the order row.

Position[2]	Position_10	Y	The identifier for the order row.
Child Product number[2]	ProdNum_24	Y	Product number.
Child Revision State[2]	R_State_7		R-state.
Child quantity[2]	Numeric_3	Y	Quantity of units for the order row.
Position[n]	Position_10	Y	The identifier for the order row.
Child Product number[n]	ProdNum_24	Y	Product number.
Child Revision State[n]	R_State_7		R-state.
Child quantity[n]	Numeric_3	Y	Quantity of units for the order row.

OrdReq_1

Name OrdReq_1
Description The External system requests the information about the specified order.
Req. Bartrack version 6.3 -
DmQ message Type 12094
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: ORDREQ
Message sequence:



If an error occurs, **OrdErr_1** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always ORDREQ.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35	Y	Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_173		To fill up message length (173 characters).

ProStrInfo

Name ProStrInfo

Description Sent by Bartrack when the **ProStrReq** message is received with a valid product number.

Req. Bartrack version 6.2 -

DmQ message Type 12085

DmQ message class: 12000

DmQ message priority: Low

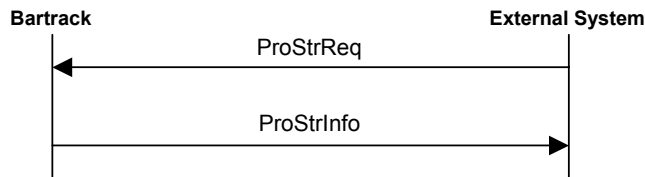
DmQ interest point: MEM

Length: Varying, 34 + 48 x [No of children]
 Maximum 100 children mean that the maximum size is 4.834 bytes.

Note. The maximum number of children in Bartrack may be higher or lower.

Transaction: RTNMSG

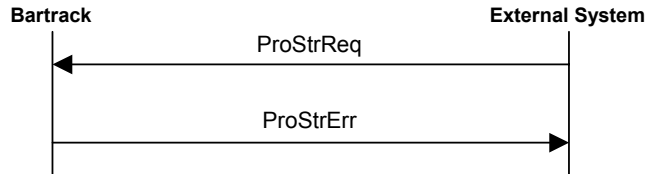
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
No of products	Numeric_3	Y	Number of products in the product structure
Position[n]	Position_10	Y	The identifier of the structure row
Child Product number[n]	ProdNum_24	Y	Product number.
Child Revision State[n]	R_State_7		R-state.
Minimum quantity[n]	Numeric_3	Y	Minimum quantity of units for the child product
Maximum quantity[n]	Numeric_3	Y	Maximum quantity of units for the child product
Exact revision[n]	YesNo_1	Y	Exact revision for the child product

ProStrErr

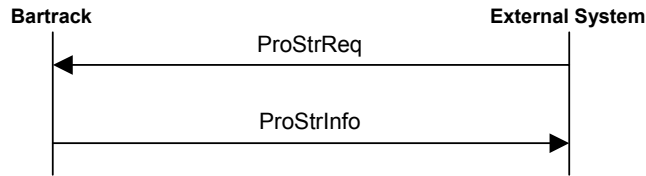
Name ProStrErr
Description Sent by Bartrack when a **ProStrReq** is received with for a non-existing product number.
Req. Bartrack version 6.2 -
DmQ message Type 12086
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

ProStrReq

Name ProStrReq
Description The External system requests the information about children (1 level) for the specified product.
Req. Bartrack version 6.2 -
DmQ message Type 12084
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: PROREQ
Message sequence:

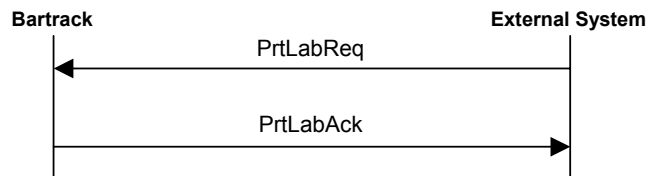


If an error occurs, **ProStrErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	PROREQ.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Product number	ProdNum_24	Y	The product number of the product.
Revision	R_State_7	Y	The product r-state of the product.
User	User_12		User-ID of the operator.
Filler	Filler_181		To fill up message length (181 characters).

PrtLabAck

Name PrtLabAck
Description The confirming or error message for a **PrtLabReq** message.
Req. Bartrack version 6.0 -
DmQ message Type 12073
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

PrtLabErr

Name PrtLabErr

Description Sent by Bartrack when the **PrtLabReq** message is received with an illegal serial number.

Req. Bartrack version 6.0 -

DmQ message Type 12075

DmQ message class: 12000

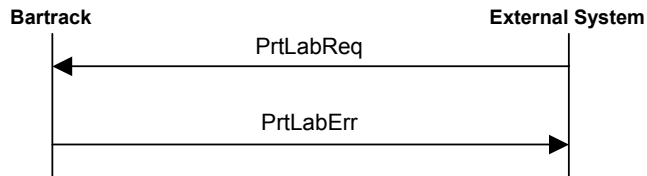
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

PrtLabReq

Name PrtLabReq

Description This request is used when you want to print out a label on a specified printer for specified individual.

Req. Bartrack version 6.0 -

DmQ message Type 12074

DmQ message class: 12000

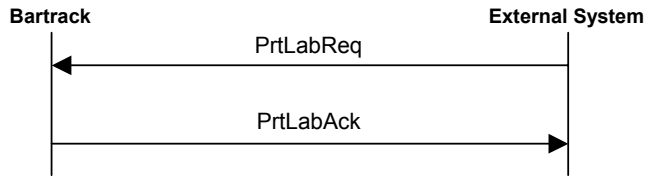
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: SERLBL

Message sequence:

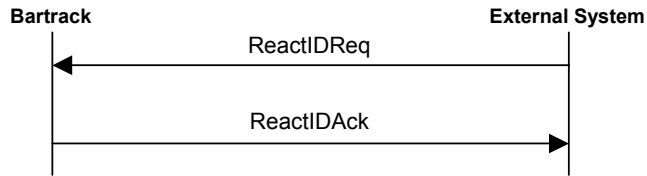


If an error occurs, **PrtLabErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERLBL.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Label name	Label_22	Y	Label name in Bartrack.
Label copies	Numeric_3		Number of copies of labels.
Printer	Printer_22	Y	Printer name in Bartrack.
User	User_12		User-ID of the operator.
Filler	Filler_145		To fill up message length (145 characters).

ReactIDAck

Name ReactIDAck
Description Sent by Bartrack when the **ReactIDReq** message is received with a valid serial number.
Req. Bartrack version 6.2 -
DmQ message Type 12092
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



If an error occurs, **ReactIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

ReactIDErr

Name ReactIDErr

Description Sent by Bartrack when a **ReactIDReq** is received with for a non-existing serial number.

Req. Bartrack version 6.2 -

DmQ message Type 12091

DmQ message class: 12000

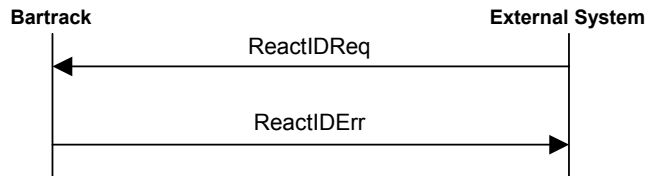
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

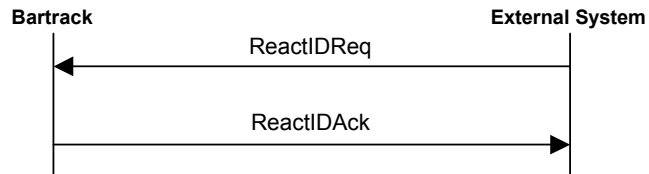
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

ReactIDReq

Name ReactIDReq
Description Request to reactivate an individual.
Req. Bartrack version 6.2 -
DmQ message Type 12090
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Fixed, 255 bytes
Transaction: SERMVE
Message sequence:

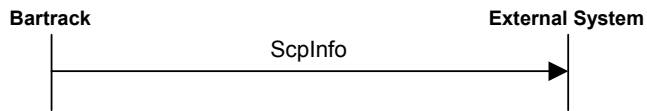


If an error occurs, **ReactIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERMVE.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

ScpInfo

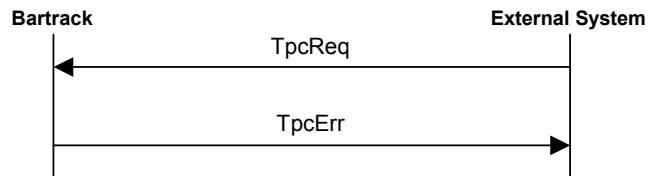
Name ScpInfo
Description This serial number has been deleted from Bartrack database.
Req. Bartrack version 6.0 -
DmQ message Type 12068
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Fixed, 255 bytes
Transaction: SERSCP
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always SERSCP.
Initiator	Initiator_1	Y	Indicates what type of system the initiator of a request was. Always R.
Serial number	Identity_20	Y	Serial number of the individual.
Filler	Filler_204		To fill up message length (204 characters).

TpcErr

Name TpcErr
Description Sent by Bartrack when a **TpcReq** is received with invalid information.
Req. Bartrack version 6.4 -
DmQ message Type 12110
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



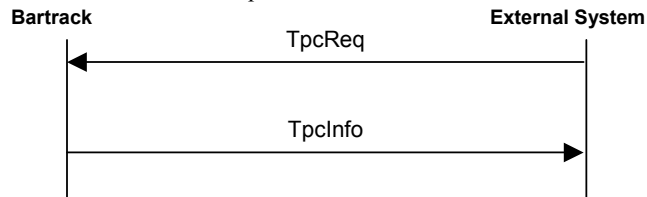
Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

TpcInfo

Name TpcInfo
Description Sent by Bartrack when the **TpcReq** message is received with valid information.
Req. Bartrack version 6.4 -
DmQ message Type 12111
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Varying, 34 + 286 x [No of TPC]
 Maximum 50 TPC mean that the maximum size is 14.334 bytes

Transaction: RTNMSG

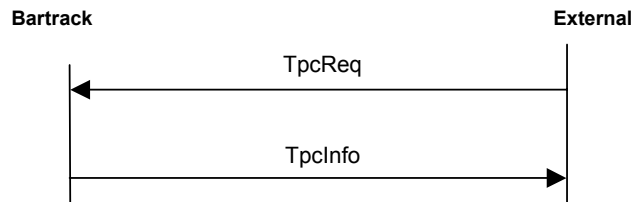
Message sequence: Where n is the number of requested number of individuals.



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
No of TPC	Numeric_3	Y	Number of test production comments
TPC id[1]	Numeric_6	Y	Test production comment id = 0 - No test production. > 0 – Test production comment defined. The test production comment id could be used in the NewIDReq_1 message.
Comment[1]	Comment_30		Comment.
Description[1]	Description_250		Description.
TPC id[2]	Numeric_6	Y	See TPC id[1].
Comment[2]	Comment_30		Comment.
Description[2]	Description_250		Description.
TPC id [n]	Numeric_6	Y	See TPC id[1]
Comment[n]	Comment_30		Comment.
Description[n]	Description_250		Description.

TpcReq

Name TpcReq
Description A request for a product's test production comment.
Req. Bartrack version 6.4 -
DmQ message Type 12109
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: TPCREQ
Message sequence: Where n is the number of requested number of individuals.
 If an error occurs, **TpcErr** is replied instead.



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always TPCREQ.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Order number	OrdNum_35		Order number. If this value is given, the orders product number and revision will be used.
Order item	OrdItem_4		Order number item. If this value is given, the orders product number and revision will be used.
Product Number	ProdNum_24	Y	Product number. If order number is supplied, this value has no effect.
Revision	R_State_7	Y	The product r-state of the product.
User	User_12		User-ID of the operator.
Filler	Filler_142		To fill up message length (142 characters).

TstInfo

Name TstInfo

Description Sent by Bartrack when the **TstReq** message is received with a valid serial number.

Req. Bartrack version 6.3.1 -

DmQ message Type 12102

DmQ message class: 12000

DmQ message priority: Low

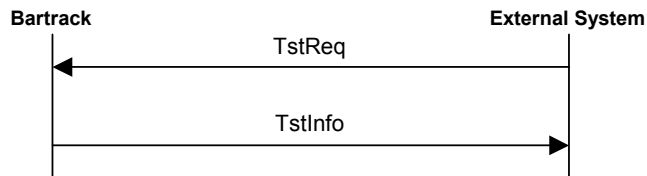
DmQ interest point: MEM

Length: Varying, 34 + 42 x [No of test status]
 Maximum 10 test status mean that the maximum size is 454 bytes.

Note. The maximum number of test status in Bartrack may be higher or lower.

Transaction: RTNMSG

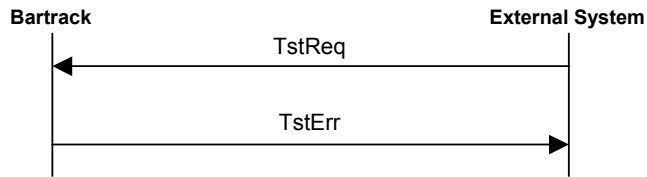
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
No of test status	Numeric_3	Y	Number of test status. NOTE! If the individual is not tested no test status is received.
Test status[n]	Test_status_1	Y	Test status: "P"= Passed "F"= Failed
Test id[n]	Test_id_6		Test identity
Test description[n]	Test_desc_16		Test description.
Time stamp[n]	Time_stamp_19		The time when the test status was updated.

TstErr

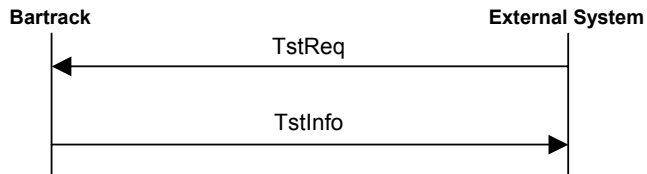
Name TstErr
Description Sent by Bartrack when a **TstReq** is received with for a non-existing serial number.
Req. Bartrack version 6.3.1 -
DmQ message Type 12101
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

TstReq

Name TstReq
Description Request all test status about an individual.
Req. Bartrack version 6.3.1 -
DmQ message Type 12100
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: DQF, UMA_SAF
Length: Fixed, 255 bytes
Transaction: TSTREQ
Message sequence:

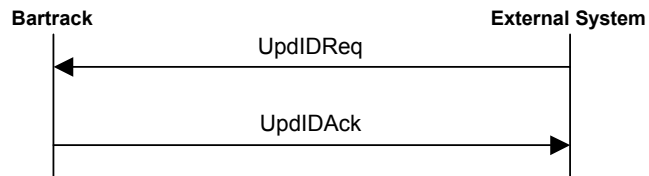


If an error occurs, **TstErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	TSTREQ.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
User	User_12		User-ID of the operator.
Filler	Filler_192		To fill up message length (192 characters).

UpdIDAck

Name UpdIDAck
Description Information message back if the update of the requested individual was successful.
Req. Bartrack version 6.0 -
DmQ message Type 12070
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

UpdIDErr

Name UpdIDErr

Description Sent by Bartrack when the **UpdIDReq** message is received with an illegal serial number.

Req. Bartrack version 6.0 -

DmQ message Type 12069

DmQ message class: 12000

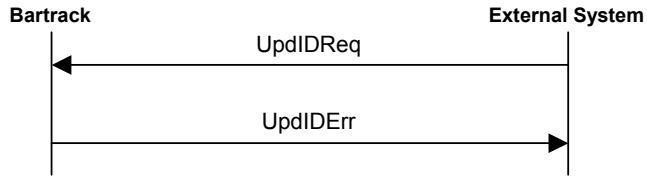
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

UpdIDReq

Name UpdIDReq

Description Request to change Information for an individual. The field **Reply wanted** indicate whether a message shall be sent back or not.

Req. Bartrack version 6.0 -

DmQ message Type 12071

DmQ message class: 12000

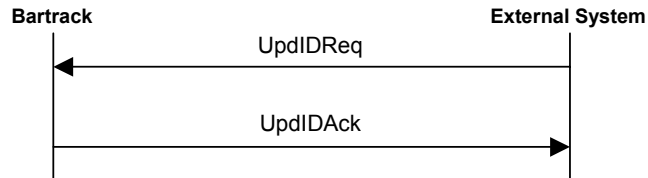
DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Fixed, 255 bytes

Transaction: SERTST: Change of test status
 SERCOM: Change of comment
 SEREXE: Change of exemption

Message sequence:

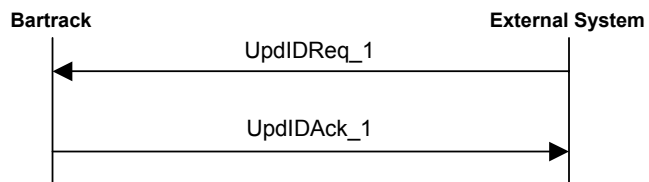


If an error occurs, **UpdIDErr** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERTST, SERCOM or SEREXE.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Test flags	Test_status_2		Used for test results, only the first position is relevant: "P"= Passed "F"= Failed "N"=Not tested Not specified = Not tested
Comment	Comment_30		Free text.
Exemption	YesNo_1		If the individual is produced with an exemption.
User	User_12		Identity of the operator.
Reply wanted	Ret_msg_1		Included in the request to determine if the initiator wants an error message back: Y = Return error message on error (Err) T = Always return something (Ack or Err) N = Never return anything
Filler	Filler_158		To fill up message length (108 characters).

UpdIDAck_1

Name UpdIDAck_1
Description Information message back if the update of the requested individual was successful.
Req. Bartrack version 6.2 -
DmQ message Type 12089
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

UpdIDErr_1

Name UpdIDErr_1

Description Sent by Bartrack when the **UpdIDReq_1** message is received with an illegal serial number.

Req. Bartrack version 6.2 -

DmQ message Type 12088

DmQ message class: 12000

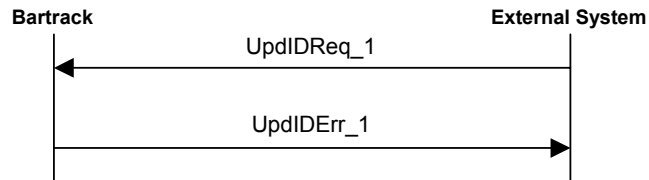
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

UpdIDReq_1

Name UpdIDReq_1

Description Request to change Information for an individual. The field **Reply wanted** indicate whether a message shall be sent back or not.

Req. Bartrack version 6.2 -

DmQ message Type 12087

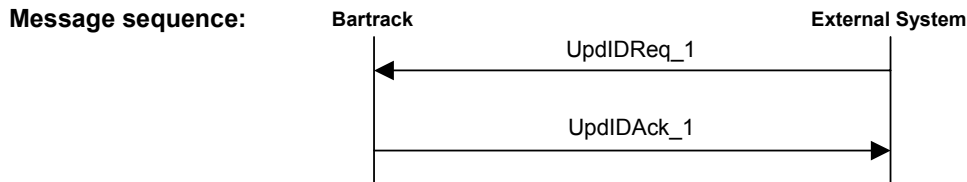
DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Fixed, 255 bytes

Transaction: SERTST: Change of test status
 SERCOM: Change of comment
 SEREXE: Change of exemption
 SERPRO: Change of product data

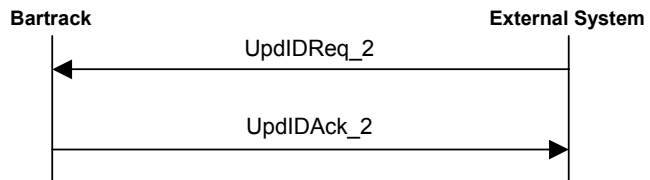


If an error occurs, **UpdIDErr_1** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERTST, SERCOM, SEREXE or SERPRO.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Test flags	Test_status_2		Used for test results, only the first position is relevant: "P"= Passed "F"= Failed "N"=Not tested Not specified = Not tested
Time stamp	Time_stamp_19		The time when the individual test status was updated.
Comment	Comment_30		Free text.
Exemption	YesNo_1		If the individual is produced with an exemption.
Product number	ProdNum_24		New product number for the individuals' product
Revision	R_State_7		New Revision for the individuals' product
User	User_12		Identity of the operator.
Reply wanted	Ret_msg_1		Included in the request to determine if the initiator wants an error message back: Y = Return error message on error (Err) T = Always return something (Ack or Err) N = Never return anything
Filler	Filler_108		To fill up message length (108 characters).

UpdIDAck_2

Name UpdIDAck_2
Description Information message back if the update of the requested individual was successful.
Req. Bartrack version 6.3.1 -
DmQ message Type 12099
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

UpdIDErr_2

Name UpdIDErr_2

Description Sent by Bartrack when the **UpdIDReq_2** message is received with an illegal serial number.

Req. Bartrack version 6.3.1 -

DmQ message Type 12098

DmQ message class: 12000

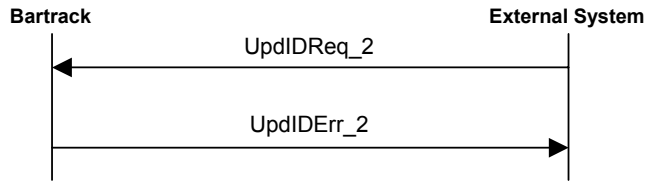
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

UpdIDReq_2

Name UpdIDReq_2

Description Request to change Information for an individual. The field **Reply wanted** indicate whether a message shall be sent back or not.

Req. Bartrack version 6.3.1 -

DmQ message Type 12097

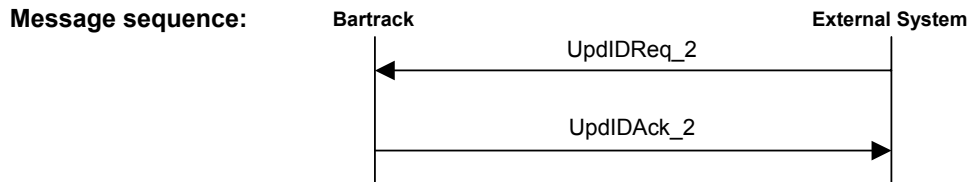
DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Fixed, 255 bytes

Transaction: SERTST: Change of test status
 SERCOM: Change of comment
 SEREXE: Change of exemption
 SERPRO: Change of product data



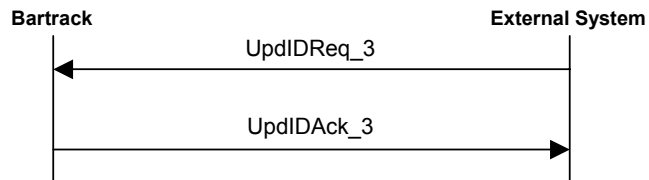
If an error occurs, **UpdIDErr_2** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERTST, SERCOM, SEREXE or SERPRO.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Test flags	Test_status_2		Used for test results, only the first position is relevant: "P"= Passed "F"= Failed "N"=Not tested Not specified = Not tested
Time stamp	Time_stamp_19		The time when the individual test status was updated.
Test id	Test_id_6		Test identity
Test description	Test_desc_16		Test description
Comment	Comment_30		Free text.
Exemption	YesNo_1		If the individual is produced with an exemption.
Product number	ProdNum_24		New product number for the individuals' product
Revision	R_State_7		New Revision for the individuals' product
User	User_12		Identity of the operator.
Reply wanted	Ret_msg_1		Included in the request to determine if the initiator wants an error message back: Y = Return error message on error (Err) T = Always return something (Ack or Err) N = Never return anything

Filler	Filler_86		To fill up message length (86 characters).
--------	-----------	--	--

UpdIDAck_3

Name UpdIDAck_3
Description Information message back if the update of the requested individual was successful.
Req. Bartrack version 6.6 -
DmQ message Type 12143
DmQ message class: 12000
DmQ message priority: Low
DmQ interest point: MEM
Length: Fixed, 255 bytes
Transaction: RTNMSG
Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Filler	Filler_224		To fill up message length (224 characters).

UpdIDErr_3

Name UpdIDErr_3

Description Sent by Bartrack when the **UpdIDReq_3** message is received with an illegal serial number.

Req. Bartrack version 6.6 -

DmQ message Type 12142

DmQ message class: 12000

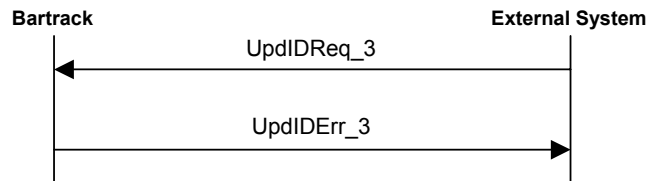
DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: RTNMSG

Message sequence:



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	Always RTNMSG.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Message	Text_80		Error message from Bartrack.
Filler	Filler_144		To fill up message length (144 characters).

UpdIDReq_3

Name UpdIDReq_3

Description Request to change Information for an individual. The field **Reply wanted** indicate whether a message shall be sent back or not.

Req. Bartrack version 6.6 -

DmQ message Type 12141

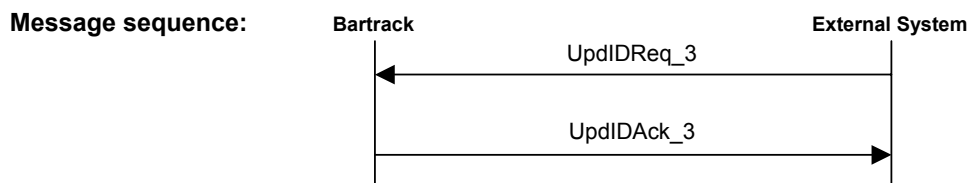
DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: DQF, UMA_SAF

Length: Fixed, 255 bytes

Transaction: SERTST: Change of test status
 SERCOM: Change of comment
 SEREXE: Change of exemption
 SERPRO: Change of product data
 SERINF: Change of information data



If an error occurs, **UpdIDErr_3** is replied instead.

Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERTST, SERCOM, SEREXE , SERPRO or SERINF.
Initiator	Initiator_1		Indicates what type of system the initiator of a request was. Always blank.
Serial number	Identity_20	Y	The serial number of the individual.
Test flags	Test_status_2		Used for test results, only the first position is relevant: "P"= Passed "F"= Failed "N"=Not tested Not specified = Not tested
Time stamp	Time_stamp_19		The time when the individual test status was updated.
Test id	Test_id_6		Test identity
Test description	Test_desc_16		Test description
Comment	Comment_30		Free text.
Exemption	YesNo_1		If the individual is produced with an exemption.
Product number	ProdNum_24		New product number for the individuals' product
Revision	R_State_7		New Revision for the individuals' product
Information	Information_35		New Information.
User	User_12		Identity of the operator.
Reply wanted	Ret_msg_1		Included in the request to determine if the initiator wants an error message back:

		Y = Return error message on error (Err) T = Always return something (Ack or Err) N = Never return anything
Filler	Filler_51	To fill up message length (51 characters).

UpdInfo

Name UpdInfo

Description Information about an individual that has been changed in Bartrack.

Req. Bartrack version 6.0 -

DmQ message Type 12070

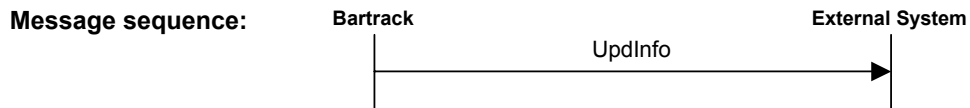
DmQ message class: 12000

DmQ message priority: Low

DmQ interest point: MEM

Length: Fixed, 255 bytes

Transaction: SERSTO: The individual was stored
 SERSHP: The individual was shipped.
 SERREV: The individual was revised (only product number and/or R-state)



Name	Type	M	Description
Message identifier	MsgID_24		Message identifier.
Transaction	BAR_Trans_6	Y	SERSTO, SERSHP, SERREV.
Initiator	Initiator_1	Y	Indicates what type of system the initiator of a request was. Always R.
Serial number	Identity_20	Y	The serial number.
Order number	OrdNum_35		Order number of the individual.
Order item	OrdItem_4	Y	Order number item of the individual.
Product number	ProdNum_24	Y	Product number of the individual.
Revision State	R_State_7	Y	R-state of the individual.
Product name	ProdName_25		Name of the product.
Parent serial number	Identity_20		The current parent serial number.
Prefix	Prefix_6	Y	The prefix in the individual's serial number. Normally the same as the factory code.
Status	IndStatus_2	Y	The manufacturing status for the individual.
Created date	Date_10	Y	The date when the individual was created.
Filler	Filler_1		To fill up message length (1 character).
No of Additional numbers	Numeric_3	Y	The number of additional numbers defined for the product of the individual. Note: The actual number of existing additional numbers may be smaller.
Test status	Test_status_2	Y	Test status, only the first position is relevant: "P"= Passed "F"= Failed "N"=Not tested.
Exemption	YesNo_1	Y	If the individual is produced with an exemption.
Destination	Destination_6	Y	The destination of the individual.

Comment	Comment_30		Free text.
User	User_12		User-ID of the operator.
Filler	Filler_16		To fill up message length (16 characters).

Types

Name of Type	Format	Description	Example
BAR_Trans_6	Char 6	Indicates the subtype of message.	SERCRT
Comment_30	Char 30	Free text.	This is for ...
Date_10	Char 10	Date in the format YYYY-MM-DD.	2002-11-23
Filler_x	Char x	A filler to make the message the desired length. The X indicates the number of positions.	
Identity_20	Char 20	Serial number of the individual.	A510103GHN
IndStatus_2	Char 2	The manufacturing status for the individual. The possible states are: CR: Created CR: Registered RV: Revised ST: Stored SC: Scrapped SH: Shipped	CR
Initiator_1	Char 1	Indicates what type of system the initiator of a request came from.	R
Label_22	Char 22	Label name in Bartrack.	PKD_11
MsgID_24	Char 24	This identifier is a "key" that can be used to identify replies from Bartrack. Bartrack copies the content of this request into the message identifier of the reply or error. The initiator can use this key to verify that the correct reply message is taken care of.	2h\$&/␣\$ ab194`56 `237jhf
Numeric_1	Char 1	Used for numeric valued 0-9.	7
Numeric_2	Char 2	Used for numeric values 0-99.	42
Numeric_3	Char 3	Used for numeric values 0-999.	372
OrdItem_4	Char 4	An item for an order number. A unique key together with the order number.	101
OrdNum_35	Char 35	An order number. A unique key together with the order number item.	AC1611764367
Password_20	Char 20	Used for verification of the login-request. Encrypted.	/Phy/&GPp
Prefix_6	Char 6	The first 6 positions of the prefix. N.B. The prefix can be 19 characters in Bartrack.	A57
Printer_22	Char 22	Printer in Bartrack.	ZEBRA_347
ProdDes_32	Char 32	Description of the product.	Front panel ...
ProdName_25	Char 25	Product name.	TRU 1900
ProdNum_24	Char 24	Product number.	ROA 119 2345
R_State_7	Char 7	R-state of the individual.	R5C

Ret_msg_1	Char 1	Included in request to determine if the initiator wants an error message back: Y = Return error message on error T = Always return something N = Never return anything	T
Test_status_2	Char 2	Test status, only the first position is relevant: "P " = Passed "F " = Failed "N " = Not tested.	P
Text_x	Char x	A message text or any other free text. The X indicates the number of positions.	Individual not found
Time_Stamp_19	Char 19	A time stamp containing date and time	2004-01-15 13:21.00
User_12	Char 12	User-ID of the operator.	ERIABCD
YesNo_1	Char 1	A Yes or No field: Y = Yes N = No	Y

Carrier

Overview

DECmessageQ carries the messages. When MRS is used the "Store And Forward" technique (SAF) is used to secure that no messages is lost. The destination for outgoing messages in the interface is set up by configuration of each instance.

The destination may be determined by using the source queue address for the incoming message. When an external System send DmQ message to Bartrack we recommend what DmQ Interest Point (DIP) should be used. If the External System wants to use MRS when we have recommended MEM Bartrack will support that.

Set-up of Buses and Groups

As there is no possibility to communicate between buses in DmQ, only applications that should interact shall belong to the same bus. Every manufacturing unit have at least one bus for live manufacturing and one bus for test environment.

Every application in one node shall have a unique group. If an application is running in several nodes, every node must also have a unique group for the application. This is especially important for client/server solutions with OpenVMS clients. As an exception, some platforms like MS-DOS and MS Windows do not need to have separate groups.

DECmessageQ for OpenVMS AXP must be configured to start up the bus and group stated in ref. ref. 1. The OpenVMS process where the interface software runs must also belong to the bus and group stated in ref. ref. 1.

Every queue and group (for external system) that Bartrack shall send to shall be declared in the cross group list for the Bartrack group.

Create Queues

One receiving queue is created for each SFT process. The queue is a "Multi Reader Queue", or MRQ for short; so several SFTx (instance of the interface) may read from the same receiving queue to increase performance.

Receiving of Messages

Each SFTx process reads the configured receive queue for requests, perform the request and when the request is secured, confirm the message if MRS is used.

The receiving software has to allocate a memory buffer large enough to hold the largest possible message.

Using CLS

Most of the DmQ messages that are sent from external systems to Bartrack require a response message. These response messages are always sent without MRS. When Bartrack acts as a master and send DmQ messages to the external system, most of these are sent with MRS.

References

- Ref. 1 PROLAB-96:002 Rev. A. Use of DEC-message-Queue at Ericsson production
- Ref. 2 DECmessageQ Installation and Configuration Guide for OpenVMS

Glossary of Terms

BEA	BEA Systems.
BmQ	BEAMessageQ – Message oriented middleware from BEA Systems. The same as DmQ.
Carrier	The carrier is the media that is used to transfer the messages. Typical carriers are DmQ, file transfer, etc.
DEC	Digital Equipment Corporation
DIP	DmQ Interest Point. The address where DmQ should deliver the message
DmQ	DECmessageQ – Message oriented middleware from BEA Systems. The same as BmQ.
External Interface	The message and carrier together make up the external interface.
External System	An external system is a system that Bartrack is communicating with, for example Testnet.
Individual	The physical realisation of a product. It is a unique item with traceability demands that is going through or is created in the manufacturing process. For example a mobile phone.
Messages	The information that is transferred between the users of the interface. The messages should as far as possible be independent of the carrier.
MRQ	Multi Reader Queue. Several processes can read an MRQ queue at the same time.
MRS	Message Recovery Service. A way to send messages in a safe way.
Origin notation	The Ericsson term "Origin notation for manufacturing units". Also known as "Factory code" (English) and "Ursprungsbeteckning" (Swedish).
Product	The description of a production item (individual). Includes product number + R-state for a specific product.
Serial number	A unique identification number used to identify an individual.
SFT	Short for "Shop Floor Transaction". The interface described in this document.
Tracy	Ericsson's global traceability database where all individuals produced and delivered are stored

Lists

List of Messages in Alphabetical Order

This is a list of all messages in alphabetical order. Some messages are used in Bartrack R5 and earlier, and some are introduced in Bartrack 6.0. Some new messages are added for the Bartrack 6.2 version.

Message Name	DMQ type	Version	Comment
AsmIDErr	12058	6.0	
AsmIDInfo	12059	6.0	
AsmIDReq	12060	6.0	
AsmInfo	12050	6.0	
AsmInfo_1	12119	6.4	
BarUnknown	12093	6.2	
ChkFlowErr	12121	6.5	
ChkFlowInfo	12122	6.5	
ChkFlowReq	12120	6.5	
FromTcyAck	12125	6.6	
FromTcyErr	12124	6.6	
FromTcyReq	12123	6.6	
FshipIdAck	12105	6.3.1	
FshipIdErr	12104	6.3.1	
FshipIdReq	12103	6.3.1	
GenAddAck	12131	6.6	
GenAddErr	12130	6.6	
GenAddReq	12129	6.6	
GetAddErr	12127	6.6	
GetAddInfo	12128	6.6	
GetAddReq	12126	6.6	
IDComplAck	12082	6.2	
IDComplErr	12083	6.2	
IDComplReq	12081	6.2	
IDErr	12051	6.0	
IDErr_1	12113	6.4	
IDErr_2	12133	6.6	
IDInfo_1	12114	6.4	

IDInfo_2	12134	6.6	
IDInfo1	12052	6.0	
IDInfo2	12053	6.0	
IDReq	12054	6.0	
IDReq_1	12112	6.4	
IDReq_2	12132	6.6	
IDStrErr	12055	6.0	
IDStrErr_1	12116	6.4	
IDStrErr_2	12136	6.6	
IDStrInfo	12056	6.0	
IDStrInfo_1	12117	6.4	
IDStrInfo_2	12137	6.6	
IDStrReq	12057	6.0	
IDStrReq_1	12115	6.4	
IDStrReq_2	12135	6.6	
LoginReply	12041	6.0	The same message is used in all versions
LoginReq	12040	6.0	The same message is used in all versions
MoveIDAck	12063	6.0	
MoveIDErr	12062	6.0	
MoveIDReq	12064	6.0	
MsA_AsmIDInfo_2	12024	R5 and earlier	
MsA_BIDErr	12016	R5 and earlier	
MsA_BIDInfo	12001	R5 and earlier	
MsA_BIDInfo_3	12038	R5 and earlier	
MsA_BIDReq	12002	R5 and earlier	
MsA_BIDStrErr	12017	R5 and earlier	
MsA_BIDStrInfo	12003	R5 and earlier	
MsA_BIDStrReq	12004	R5 and earlier	
MsA_CoupleIDErr	12018	R5 and earlier	
MsA_CoupleIDInfo	12007	R5 and earlier	
MsA_CoupleIDReq	12008	R5 and earlier	
MsA_CreIDInfo	12023	R5 and earlier	
MsA_DRegIDErr	12019	R5 and earlier	
MsA_DRegIDInfo	12009	R5 and earlier	
MsA_DRegIDReq	12010	R5 and earlier	
MsA_LoginReply	12041	R5 and earlier	The same message is used in all versions
MsA_LoginReq	12040	R5 and earlier	The same message is used in all versions
MsA_NewIDErr	12020	R5 and earlier	
MsA_NewIDInfo_3	12035	R5 and earlier	
MsA_NewIDReq	12012	R5 and earlier	
MsA_ScpIDInfo	12022	R5 and earlier	
MsA_UpdIDAck	12039	R5 and earlier	
MsA_UpdIDErr	12026	R5 and earlier	
MsA_UpdIDInfo	12013	R5 and earlier	
MsA_UpdIDReq	12025	R5 and earlier	
MsA_XXXInfo	12015	R5 and earlier	

MsA_XXXReq	12015	R5 and earlier	
MsgToFile	12090	6.2	
NewIDErr	12065	6.0	
NewIDErr_1	12107	6.4	
NewIDErr_2	12139	6.6	
NewIDInfo	12066	6.0	
NewIDInfo_1	12108	6.4	
NewIDInfo_2	12140	6.6	
NewIDReq	12067	6.0	
NewIDReq_1	12106	6.4	
NewIDReq_2	12138	6.6	
NewInfo	12061	6.0	
NewInfo_1	12118	6.4	
OrdErr	12078	6.0	
OrdErr_1	12095	6.3	
OrdInfo	12079	6.0	
OrdInfo_1	12096	6.3	
OrdReq	12080	6.0	
OrdReq_1	12094	6.3	
ProStrInfo	12085	6.2	
ProStrErr	12086	6.2	
ProStrReq	12084	6.2	
PrtLabAck	12073	6.0	
PrtLabErr	12075	6.0	
PrtLabReq	12074	6.0	
ReactIDAck	12092	6.2	
ReactIDErr	12091	6.2	
ReactIDReq	12090	6.2	
ScpInfo	12068	6.0	
TpcInfo	12111	6.4	
TpcErr	12110	6.4	
TpcReq	12109	6.4	
TstErr	12101	6.3.1	
TstInfo	12102	6.3.1	
TstReq	12100	6.3.1	
UpdIDAck	12072	6.0	
UpdIDAck_1	12089	6.2	
UpdIDAck_2	12099	6.3.1	
UpdIDAck_3	12143	6.6	
UpdIDErr	12069	6.0	
UpdIDErr_1	12088	6.2	
UpdIDErr_2	12098	6.3.1	
UpdIDErr_3	12142	6.6	
UpdIDReq	12071	6.0	
UpdIDReq_1	12087	6.2	
UpdIDReq_2	12097	6.3.1	

UpdIDReq_3	12141	6.6	
UpdInfo	12070	6.0	

List of Messages in DMQ Type Order

This is a list of all messages in DmQ type order. The minimum Version column indicates the version of Bartrack that supports the message.

DMQ type	Message Name	Page #	Minimum Version	Comment
12001	MsA_BIDInfo		R5 and earlier	
12002	MsA_BIDReq		R5 and earlier	
12003	MsA_BIDStrInfo		R5 and earlier	
12004	MsA_BIDStrReq		R5 and earlier	
12007	MsA_CoupleIDInfo		R5 and earlier	
12008	MsA_CoupleIDReq		R5 and earlier	
12009	MsA_DRegIDInfo		R5 and earlier	
12010	MsA_DRegIDReq		R5 and earlier	
12012	MsA_NewIDReq		R5 and earlier	
12013	MsA_UpdIDInfo		R5 and earlier	
12015	MsA_XXXInfo		R5 and earlier	
12015	MsA_XXXReq		R5 and earlier	
12016	MsA_BIDErr		R5 and earlier	
12017	MsA_BIDStrErr		R5 and earlier	
12018	MsA_CoupleIDErr		R5 and earlier	
12019	MsA_DRegIDErr		R5 and earlier	
12020	MsA_NewIDErr		R5 and earlier	
12022	MsA_ScpIDInfo		R5 and earlier	
12023	MsA_CreIDInfo		R5 and earlier	
12024	MsA_AsmIDInfo_2		R5 and earlier	
12025	MsA_UpdIDReq		R5 and earlier	
12026	MsA_UpdIDErr		R5 and earlier	
12035	MsA_NewIDInfo_3		R5 and earlier	
12038	MsA_BIDInfo_3		R5 and earlier	
12039	MsA_UpdIDAck		R5 and earlier	
12040	MsA_LoginReq		R5 and earlier	The same message is used in all versions
12040	LoginReq	50	6.0	The same message is used in all versions
12041	MsA_LoginReply		R5 and earlier	The same message is used in all versions
12041	LoginReply	49	6.0	The same message is used in all versions
12050	AsmInfo	36	6.0	
12051	IDErr	41	6.0	
12052	IDInfo1	42	6.0	
12053	IDInfo2	43	6.0	
12054	IDReq	44	6.0	
12055	IDStrErr	45	6.0	
12056	IDStrInfo	796	6.0	
12057	IDStrReq	48	6.0	
12058	AsmIDErr	34	6.0	
12059	AsmIDAck	33	6.0	
12060	AsmIDReq	35	6.0	

12061	NewInfo	59	6.0	
12062	MoveIDErr	52	6.0	
12063	MoveIDAck	51	6.0	
12064	MoveIDReq	53	6.0	
12065	NewIDErr	55	6.0	
12066	NewIDInfo	56	6.0	
12067	NewIDReq	57	6.0	
12068	SepInfo	77	6.0	
12070	UpdInfo	84	6.0	
12073	PrtLabAck	71	6.0	
12074	PrtLabReq	73	6.0	
12075	PrtLabErr	72	6.0	
12078	OrdErr	60	6.0	
12079	OrdInfo	63	6.0	
12080	OrdReq	63	6.0	
12081	IDComplReq	40	6.2	
12082	IDComplAck	38	6.2	
12083	IDComplErr	39	6.2	
12084	ProStrReq	70	6.2	
12085	ProStrInfo	68	6.2	
12086	ProStrErr	69	6.2	
12087	UpdIDReq_1	83	6.2	
12088	UpdIDErr_1	82	6.2	
12089	UpdIDAck_1	81	6.2	
12090	MsgToFile	54	6.2	The 12090 messages are handled by different processes/queues.
12090	ReactIDReq	76	6.2	The 12090 messages are handled by different processes/queues.
12091	ReactIDErr	75	6.2	
12092	ReactIDAck	74	6.2	
12093	BarUnknown	37	6.2	
12094	OrdReq_1	67	6.3	
12095	OrdErr_1	64	6.3	
12096	OrdInfo_1	65	6.3	
12097	UpdIDReq_2		6.3.1	
12098	UpdIDErr_2		6.3.1	
12099	UpdIDAck_2		6.3.1	
12100	TstReq		6.3.1	
12101	TstErr		6.3.1	
12102	TstInfo		6.3.1	
12103	FshipIdReq		6.3.1	
12104	FshipIdErr		6.3.1	
12105	FshipIdAck		6.3.1	
12106	NewIDReq_1		6.4	
12107	NewIDErr_1		6.4	
12108	NewIDInfo_1		6.4	
12109	TpcReq		6.4	

12110	TpcErr		6.4	
12111	TpcInfo		6.4	
12112	IDReq_1		6.4	
12113	IDErr_1		6.4	
12114	IDInfo_1		6.4	
12115	IDStrReq_1		6.4	
12116	IDStrErr_1		6.4	
12117	IDStrInfo_1		6.4	
12118	NewInfo_1		6.4	
12119	AsmInfo_1		6.4	
12120	ChkFlowReq		6.5	
12121	ChkFlowErr		6.5	
12122	ChkFlowInfo		6.5	
12123	FromTcyReq		6.6	
12124	FromTcyErr		6.6	
12125	FromTcyAck		6.6	
12126	GetAddReq		6.6	
12127	GetAddErr		6.6	
12128	GetAddInfo		6.6	
12129	GenAddReq		6.6	
12130	GenAddErr		6.6	
12131	GenAddAck		6.6	
12132	IDReq_2		6.6	
12133	IDErr_2		6.6	
12134	IDInfo_2		6.6	
12135	IDStrReq_2		6.6	
12136	IDStrErr_2		6.6	
12137	IDStrInfo_2		6.6	
12138	NewIDReq_2		6.6	
12139	NewIDErr_2		6.6	
12140	NewIDInfo_2		6.6	
12141	UpdIDReq_3		6.6	
12142	UpdIDErr_3		6.6	
12143	UpdIDAck_3		6.6	

List of Pictures

ERROR! NO TABLE OF FIGURES ENTRIES FOUND.

List of Tables

Error! No table of figures entries found.