

Advanced Robotics Control Language Fleet Manager - Mobile Robots

Integration Guide

Copyright Notice

The information contained herein is the property of Omron Robotics and Safety Technologies, Inc., and shall not be reproduced in whole or in part without prior written approval of Omron Robotics and Safety Technologies, Inc. The information herein is subject to change without notice and should not be construed as a commitment by Omron Robotics and Safety Technologies, Inc. The documentation is periodically reviewed and revised.

Omron Robotics and Safety Technologies, Inc., assumes no responsibility for any errors or omissions in the documentation. Critical evaluation of the documentation by the user is welcomed. Your comments assist us in preparation of future documentation. Please submit your comments to:

techpubs@omron.com.

Copyright © 2017-2019 by Omron Robotics and Safety Technologies, Inc. All rights reserved.

Any trademarks from other companies used in this publication
are the property of those respective companies.

Created in the United States of America

Table Of Contents

Chapter 1: Introduction to ARCL	7
1.1 Version Requirements	7
1.2 How Do I Begin?	7
1.3 How Can I Get Help?	8
Related Manuals	8
Chapter 2: Set ARCL Parameters in MobilePlanner	9
2.1 Definitions	9
2.2 Accessing the Configuration Options	9
To access ARCL configuration options from MobilePlanner:	10
Understanding the Configuration Parameters	12
Outgoing Enterprise ARCL Connection Setup Parameters	13
Outgoing Enterprise ARCL Commands Parameters	14
Chapter 3: Connecting with a Telnet Client	15
Setting the Connection Parameters	15
Connecting to ARCL	16
Chapter 4: Using the ARCL Commands	19
4.1 Understanding the Commands	20
Document Conventions	20
Command Notes	21
Data Types	22
Status and Error Messages	22
Status Conditions	23
4.2 Using the Queuing Commands	25
4.3 Working With Payloads	25
Chapter 5: ARCL Command Reference	27
5.1 getDateime Command	30
Syntax	30
Parameters	30
Examples	30
5.2 payloadQuery Command (shortcut: pq)	31
Syntax	31
Parameters	31
Responses	31
Details	31

Examples	31
Related Commands	33
5.3 payloadSlotCount Command (shortcut: psc)	34
Syntax	34
Parameters	34
Responses	34
Examples	34
Related Commands	35
5.4 queryFaults Command (shortcut: qf)	36
Syntax	36
Usage Considerations	36
Parameter	36
Responses	36
Details	36
Example	36
Related Commands	38
5.5 queueCancel Command (shortcut: qc)	39
Syntax	39
Parameters	39
Responses	39
Details	40
Examples	40
Related Commands	41
5.6 queueModify Command (shortcut: qmod)	42
Syntax	42
ARAM Settings	42
Parameters	42
Responses	43
Examples	44
Related Commands	46
5.7 queueMulti Command (shortcut: qm)	47
Syntax	47
ARAM Settings	47
Parameters	47
Responses	48
Details	48
Examples	48
Related Commands	49
5.8 queuePickup Command (shortcut: qp)	50
Syntax	50
ARAM Settings	50
Parameters	50
Responses	50
Details	51
Examples	51
Related Commands	51

5.9	queuePickupDropoff Command (shortcut: qpd)	53
	Syntax	53
	Parameters	53
	Responses	53
	Details	54
	Examples	54
	Related Commands	56
5.10	queueQuery Command (shortcut: qq)	58
	Syntax	58
	Parameters	58
	Responses	58
	Details	59
	Examples	59
	Related Commands	59
5.11	queueShow Command (shortcut: qs)	61
	Syntax	61
	Usage Considerations	61
	Parameters	61
	Responses	61
	Details	61
	Examples	62
	Related Commands	62
5.12	queueShowCompleted Command (shortcut: qsc)	63
	Syntax	63
	Usage Considerations	63
	Parameters	63
	Returns	63
	Details	63
	Examples	64
	Related Commands	64
5.13	queueShowRobot Command (shortcut: qsr)	65
	Syntax	65
	Usage Considerations	65
	Parameters	65
	Responses	65
	Details	65
	Examples	65
	Related Commands	66

Revision History

Revision code	Date	Revised Content
01	January, 2017	Original release
02	August, 2019	<ul style="list-style-type: none"> • Replaced all "Omron Adept Technologies, Inc." with "Omron Robotics and Safety Technologies, Inc." in Copyright page. • Updated copyright dates to include 2019. • Updated techpub's email address in copyright page. Previously it was "techpubs@adept.com". It is now "techpubs@omron.com". • Removed references to www.adept.com. • Corrected Omron's website address to ia.omron.com. • Title of the manual changed from "ARCL Enterprise Manager Integration Guide" to "ARCL Fleet Manager Integration Guide". • Added Revision History. • Updated the manual style. • Replaced "mobile robot" and "robot" with "AMR". • Where applicable, "Omron Robotics and Safety Technologies, Inc." got replaced with "Omron". • Added Enterprise Manager 2100 User's Guide as well as ARCL Reference Guide to Related Manuals • Moved Related Manuals to section 1.3 How Can I Get Help? • Removed listAdd, listExecute, and listStart from SetUpError example and replaced with Play command. • Removed "See Also" section from all chapters. • Made minor updates to the content. • Removed "This ARCL command is only available on the AMR" from section 4.1, as it does not occur anywhere else in this manual. • Removed "This ARCL command is available on the AMR and Fleet Manager" from section 4.1, as it does not occur anywhere else in this manual. • Added a note to section 4.1, explaining that the only Usage Consideration applicable to commands in this manual, is the one noted. • Revised description of the Details section in chapter 4.1. This section is optional and only used if more information has to be covered. • Removed Usage Considerations section from each command, except for queryFaults Command, queueShow Command, queueShowCompleted, and queueShowRobot Command. For these commands only removed "This ARCL command is available on the Fleet Manager". • Removed Details section from section 5.3 and 5.6.

Chapter 1: Introduction to ARCL

The Advanced Robotics Command Language (ARCL) is a simple, text-based, command-and-response operating language for integrating a fleet of Autonomous Mobile Robots (AMR) with an external automation system.

ARCL allows you to submit jobs to the Fleet Manager, and monitor the job status from start to finish. It also allows you to monitor payload information, if reported by the AMR, in the fleet.

The EM2100 Version of ARCL is for use with the Fleet Manager software and EM2100 appliance. This hardware and software combination has been specially designed and configured to manage a fleet of AMRs operating in a facility. Therefore, it uses a minimal ARCL command set, because all of the critical work is being handled directly by the appliance and Fleet Manager software.

This section discusses the following topics:

1.1 Version Requirements	7
1.2 How Do I Begin?	7
1.3 How Can I Get Help?	8
Related Manuals	8

For more information on using the Mobile Robots Software Suite, refer to Mobile Robots Software Suite User's Guide.

1.1 Version Requirements

This document pertains to ARAM version 4.6 and later.

If you need assistance, see [How Can I Get Help?](#) on page 8

1.2 How Do I Begin?

Before you can access Advanced Robotics Command Language, you must complete the following steps:

1. Set ARCL Parameters in MobilePlanner.

Define the ARCL server address, port number and password parameters in MobilePlanner, and configure other ARCL parameters. The server port will not open without a password, therefore you must configure a password before you can connect to ARCL. For details, see [Set ARCL Parameters in MobilePlanner](#) on page 9.

2. Connect to ARCL Using a Telnet Client.

Using a Telnet client, connect to ARCL to access and run the ARCL commands on the Mobile Robots Software Suite. For details, see [Connecting with a Telnet Client](#) on page 15.

After you've set up and established a connection to the ARCL server, you can start using the ARCL commands to submit and monitor jobs that will be performed by the fleet. You can do

all of this with or without MobilePlanner. For more details, see Using the ARCL Commands on page 19.

1.3 How Can I Get Help?

For details on getting assistance with your Omron software or hardware, you can access the corporate website:

<http://www.ia.omron.com>.

Related Manuals

In addition to this manual, you may want to refer to the following manuals:

Manual	Description
<i>Mobile Robot LD Safety Guide</i>	Describes safety information for our robots.
<i>Mobile Robots Software Suite User's Guide</i>	Describes the Mobile Robots Software Suite software, including SetNetGo and MobilePlanner.
<i>Mobile Robots OEM - LD Platform User's Guide</i>	Describes the installation, start-up, operation, and maintenance of the mobile robot base.
<i>Enterprise Manager 1100 User's Guide</i>	Describes the installation and operation of the Enterprise Manager 1100 appliance and the Enterprise Manager software.
<i>Enterprise Manager 2100 User's Guide</i>	Describes the installation and operation of the Enterprise Manager 2100 appliance and the Enterprise Manager software.
<i>ARCL Reference Guide</i>	Describes operation of Advanced Robotics Command Language commands.

Chapter 2: Set ARCL Parameters in MobilePlanner

This section describes how to access the configuration items in the MobilePlanner software. It describes the following:

- Accessing the Configuration Options on page 9
- Understanding the Configuration Parameters on page 12

2.1 Definitions

Platform: The basic part of the AMR, including the chassis, drive assemblies, suspension, wheels, battery, lasers, sonar, a core with gyroscope, navigation software, connectors for interfacing with and powering the payload structure, and platform covers.

Payload Structure: Anything you attach to the platform. This could be as simple as a box for holding parts or documents that you want transported, or as complicated as a robotic arm that will be used to pick up parts to transport.

AMR (Autonomous Mobile Robot): A platform with a payload structure attached to it. This is your complete mobile robot, which will transport your products, parts, or data.

Fleet: Two or more AMRs operating in the same area, and managed by the same Fleet Manager.

EM2100: A network appliance, with built-in processor, that runs the FLOW Core software.

FLOW Core: All of the software used by Fleet Operations Workspace. The software runs on a combination of the EM appliance(s), the AMRs, and the user's PC.

Fleet Operations Workspace (FLOW): One or more EM2100 appliance(s) and the Flow Core.

Fleet Manager: The set of capabilities within the FLOW Core software that executes all fleet management activities. These include the management of maps, AMR configuration, job queue management, and traffic coordination. Fleet Manager runs on an EM2100 appliance.

2.2 Accessing the Configuration Options

These sections allow you to access configuration parameters that control the ARCL server and its interaction with connected clients.



CAUTION: The server port will not open without a password. Therefore, you must configure a password before you can connect to ARCL.

To access ARCL configuration options from MobilePlanner:

1. Open the MobilePlanner software, version 4.0 or later, and connect to the AMR. Refer to the *Mobile Robots Software Suite User's Guide* for details on installing and starting MobilePlanner.
2. From the MobilePlanner > Config, select the Robot Interface tab.
3. Select ARCL server setup from the Sections: column. See the following figure.

These parameters allow you to control the client-server connection between an offboard client process (such as Telnet or PuTTY) and ARCL.

Incoming connections refer to a client initiating the connection to the EM2100. Multiple simultaneous connections are allowed and supported.

NOTE: ARCL server setup lets you configure the port for incoming connections. This does not affect outgoing connections.

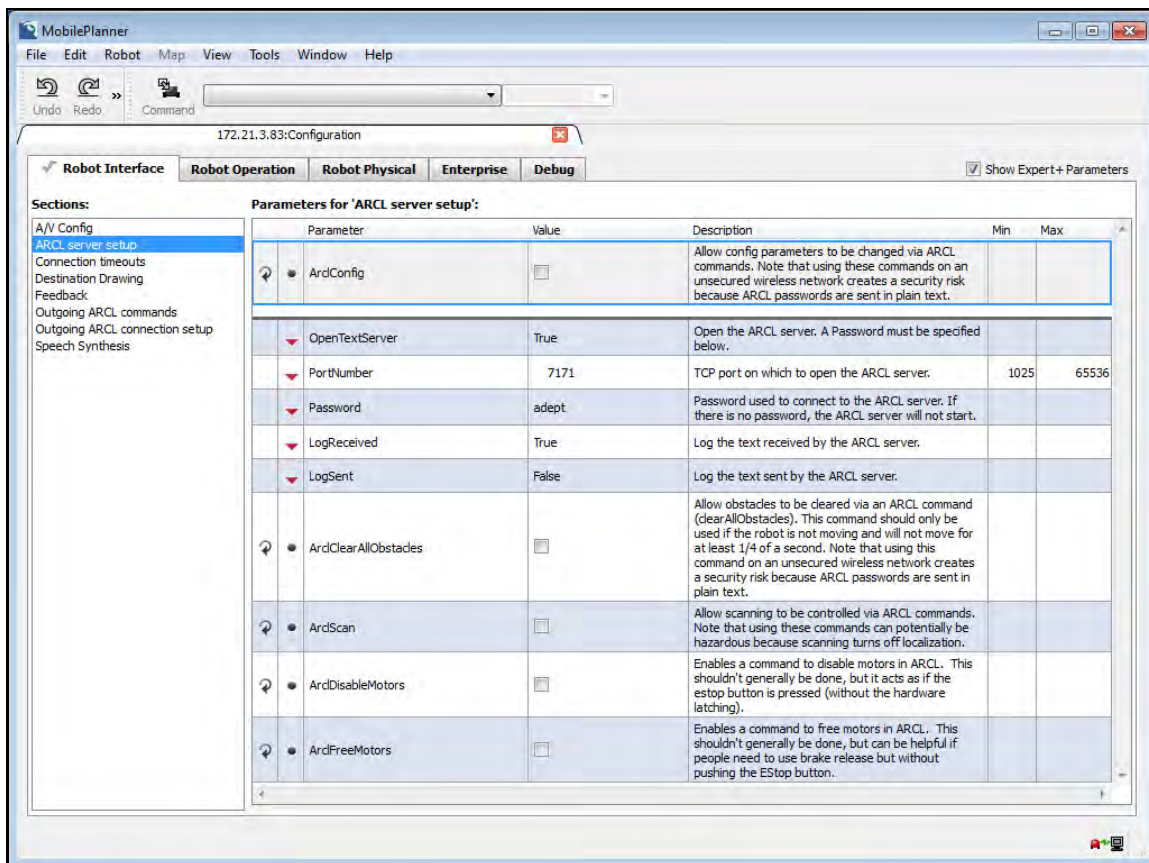


Figure 2-1. ARCL Server Setup Parameters

For more information on using a client (like Telnet or PuTTY), see Connecting with a Telnet Client on page 15.

4. Select Outgoing Enterprise ARCL Commands from the Sections: column to display the

parameters that allow you to configure commands that are automatically executed on the connection indicated in the Outgoing Enterprise ARCL connection setup. For more details, see Outgoing Enterprise ARCL Commands Parameters on page 14.

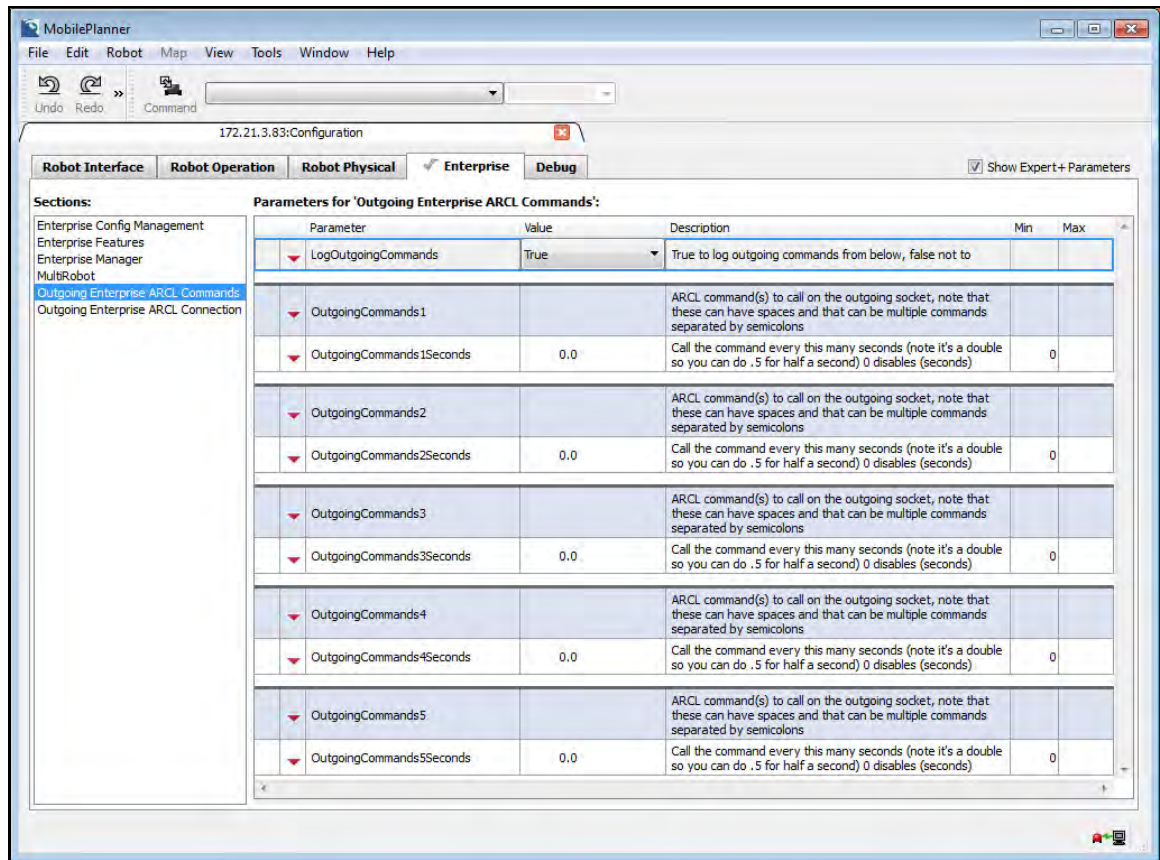


Figure 2-2. Outgoing Enterprise ARCL Commands

5. Select Outgoing Enterprise ARCL connection setup from the Sections: column to display the parameters that allow you to send data from the EM2100 using ARCL commands, intended to connect to the facility Warehouse Management System (WMS)/Management Execution System (MES). For more details, refer to Outgoing Enterprise ARCL Connection Setup Parameters on page 13.

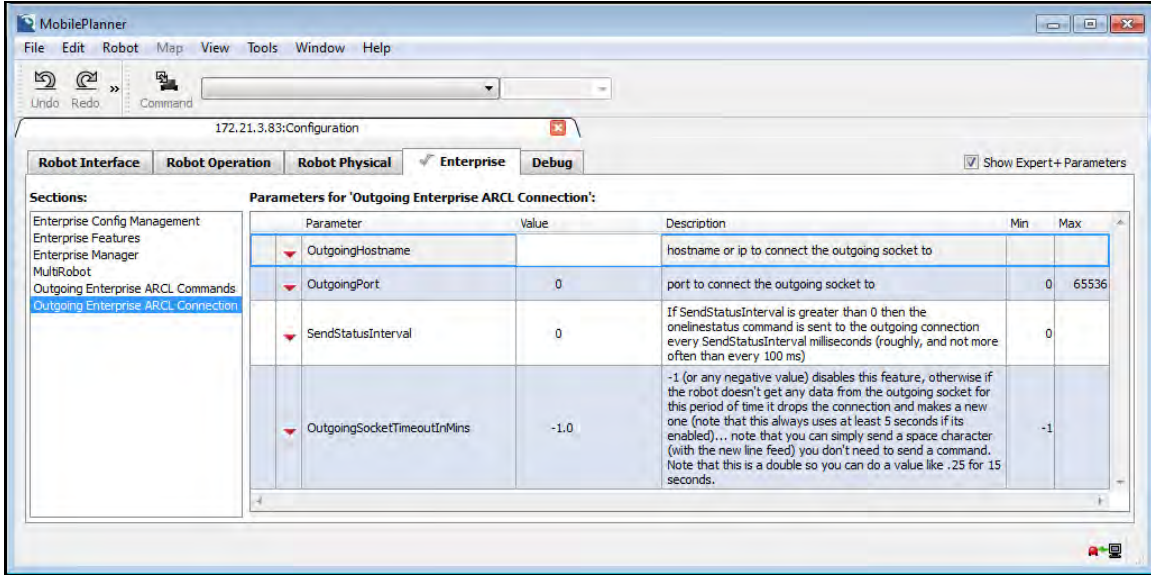


Figure 2-3. Outgoing Enterprise ARCL Connection Setup

6. After the configuration options are set, click the **Save** button on the toolbar to save the changes to the Configuration file. Changes do not take effect until the AMR is idle and stationary.

Understanding the Configuration Parameters

The configuration parameters are grouped by function - each functional group is accessed from the alphabetical list in the left pane. The corresponding configuration parameters are listed in a tabular format on the configuration pages, as shown in the previous figures. The parameters are organized alphabetically. You can sort the list in ascending or descending order by name, value, min, or max.

Each parameter has a description that briefly describes its function. The selected parameter's help description is located in the Description column. The description shows up at the bottom of the window when the entire contents can't be shown in the Description column. For an example, see the following figure.

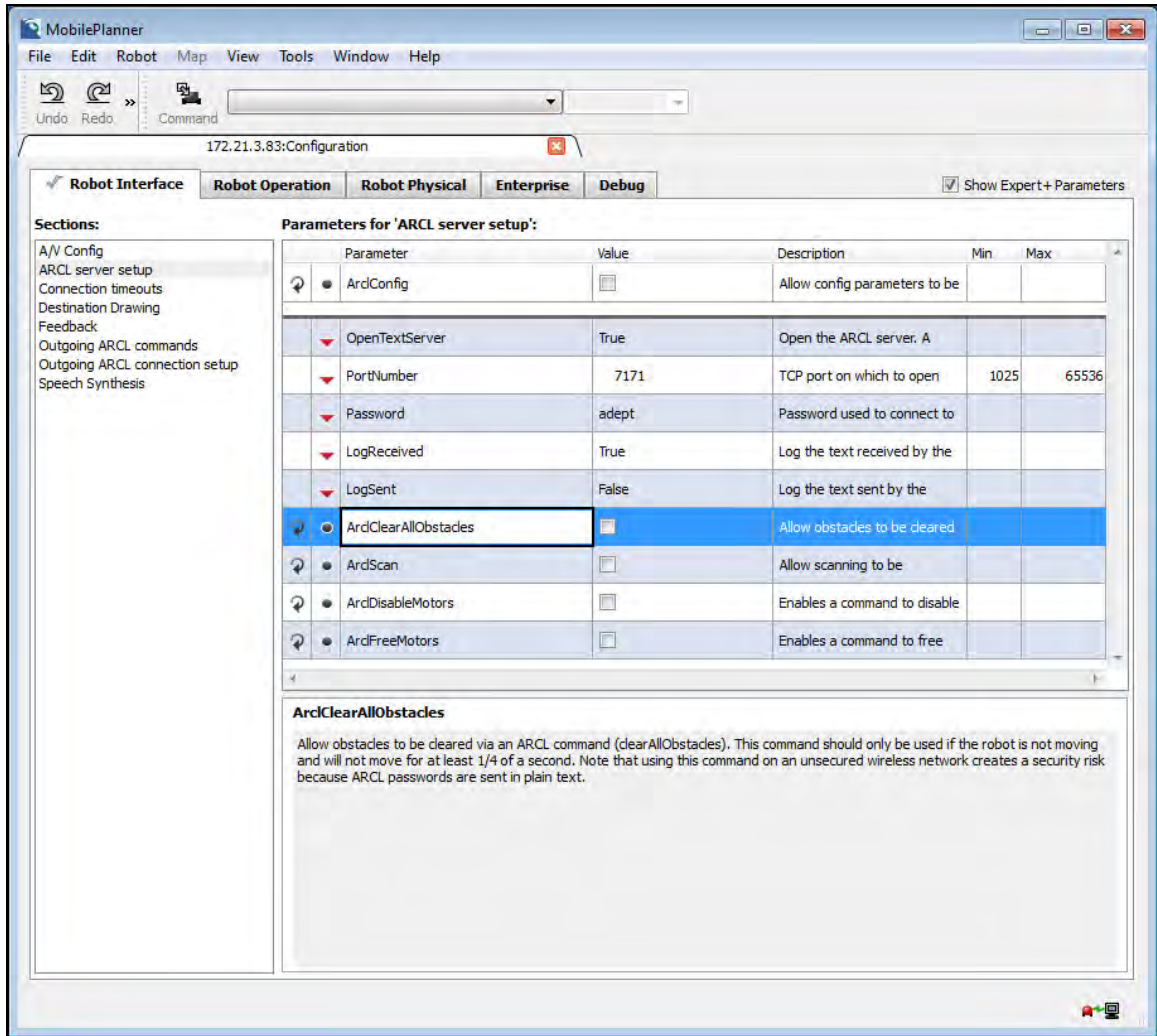


Figure 2-4. Parameter Help

Outgoing Enterprise ARCL Connection Setup Parameters

The Outgoing Enterprise ARCL connection setup parameters are used to instruct the EM2100 to initiate an outgoing ARCL TCP connection to another device on the network. This approach can be used in lieu of requiring that the other device initiate an incoming connection to the EM2100.

There may be hand-shaking involved between the Fleet Manager and the factory equipment, to determine when the command should be executed.

In order to use this feature, the OutgoingHostname needs to be set to a string and the OutgoingPort needs to be a non-zero number.

Use of the outgoing ARCL connections:

NOTE: The outgoing connection can be used to automatically execute certain ARCL commands at specified intervals. This can be useful for gathering certain

information without requiring that the application, running on the connected device, continuously request the data.

Outgoing Enterprise ARCL Commands Parameters

The Outgoing Enterprise ARCL command parameters allow you to set the EM2100 up to automatically generate ARCL commands at regular intervals. You can send one or more ARCL commands. To send multiple commands, separate each command with a pipe character (|). For example, set the OutGoingCommands1 parameter to:

```
Queueshowrobot default echoit

QueueRobot: "Robot1" UnAvailable EStopPressed echoit
QueueRobot: "Robot2" UnAvailable Interrupted echoit
QueueRobot: "Robot3" UnAvailable InterruptedButNotYetIdle echoit
QueueRobot: "Robot4" Available Available echoit
QueueRobot: "Robot5" InProgress Driving echoit

QueueRobot: "Robot6" UnAvailable NotUsingEnterpriseManager echoit
QueueRobot: "Robot7" UnAvailable UnknownBatteryType echoit
QueueRobot: "Robot8" UnAvailable ForcedDocked echoit
QueueRobot: "Robot9" UnAvailable NotLocalized echoit
QueueRobot: "patrolbot" UnAvailable Fault_Driving_Application_
faultName echoit

EndQueueShowRobot
```

Then you could parse the output to compare the number of AMRs connected to the number of AMRs that should be connected, and generate an alarm if there is a mismatch.

Chapter 3: Connecting with a Telnet Client

This section tells you how to connect your AMR to ARCL using a client, such as Telnet or PuTTY.

Setting the Connection Parameters

1. Open the MobilePlanner software, version 4.0 or later, and connect to the AMR. Refer to the *Mobile Robots Software Suite User's Guide* for details on installing and starting MobilePlanner.
2. From the Configuration tab, select the Robot Interface tab.
3. Select ARCL Server Setup from the Sections column. The ARCL Server Setup parameters are shown in the following figure.

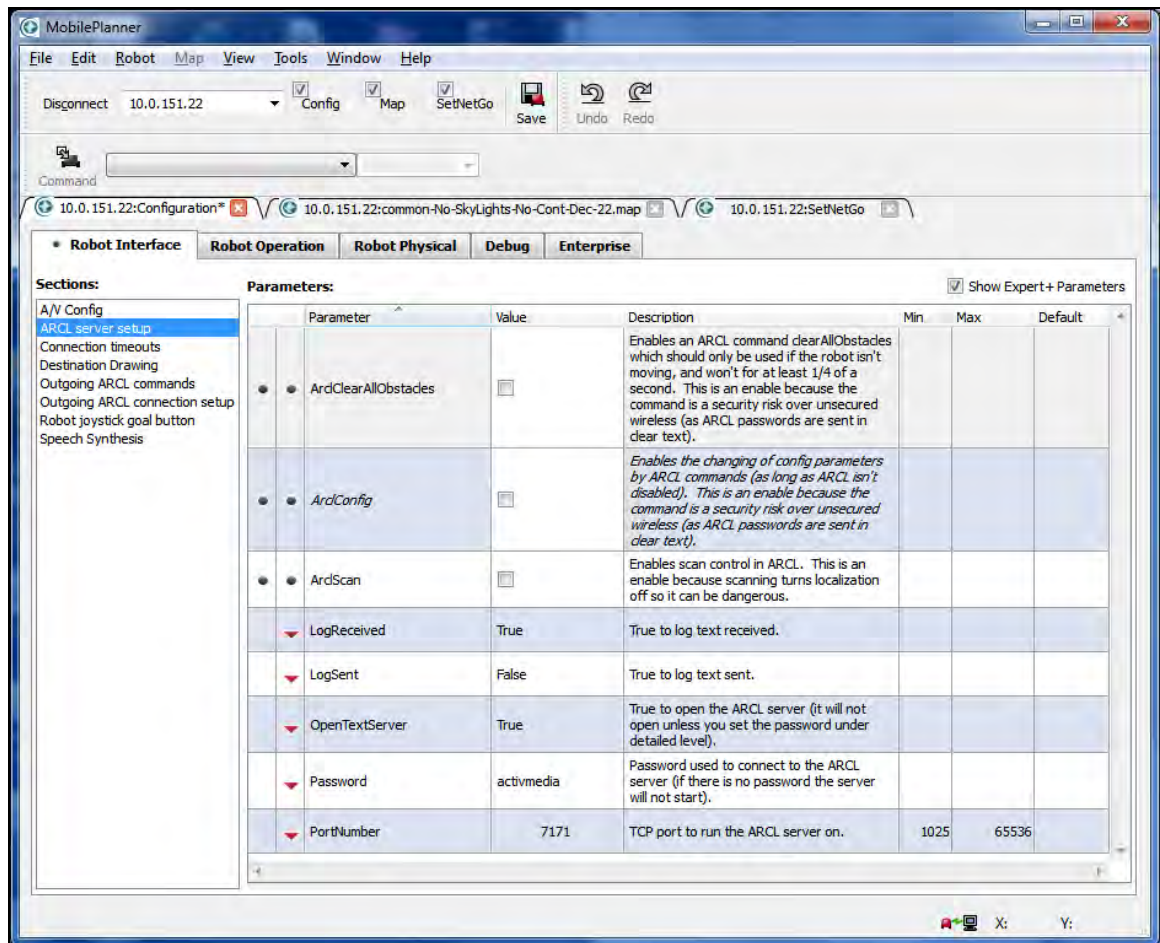


Figure 3-1. ARCL Server Setup Parameters

These parameters allow you to control the client-server connection. See Understanding the Configuration Parameters on page 12 for details.

4. Enter a password for the Telnet client. If a password already exists, make a note of it so that you can open the ARCL server from the Telnet connection.

Connecting to ARCL

The following instructions describe how to connect to ARCL using the Command Prompt window in the Microsoft Windows operating system. You can also use a terminal-emulation utility, such as PuTTY. For details on PuTTY, see the PuTTY website: <http://www.putty.org>.

1. On a Windows-based PC, open the Command Prompt window.
2. In Windows, hold down the "Window" key and press the "R" key to open the Run dialog box. Type `cmd` to display the command terminal.

NOTE: On some Windows installations, you may need to enable Telnet using:

Control Panel > Programs and Features > Turn Windows feature On or Off.

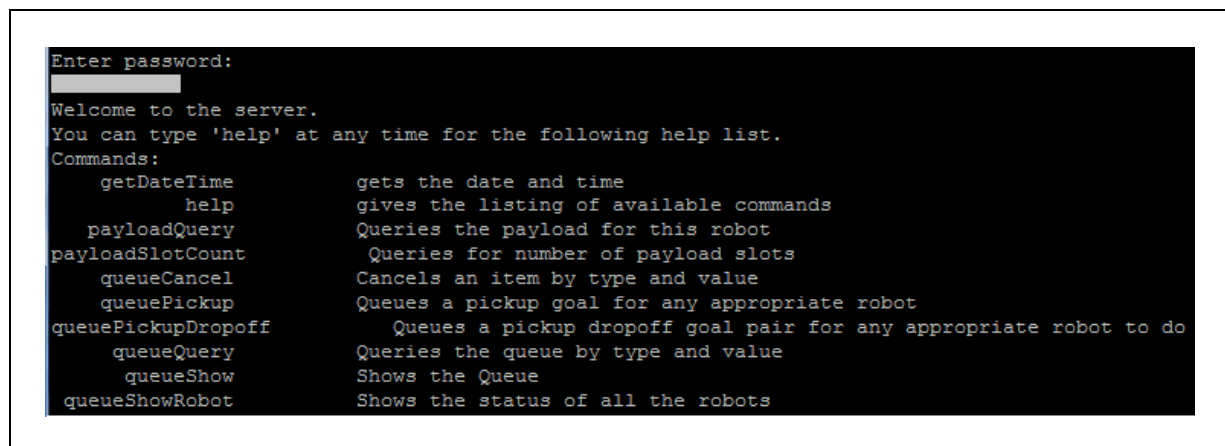
3. Start Telnet using the ARCL server address and the port number specified in the Advanced Robotics Command Language Server Setup Parameters. For example:

```
Telnet 192.168.0.44 7171
```

4. Enter the password that you set in Setting the Connection Parameters on page 15. If you mis-type the password, you will have to restart the Telnet client.

After you have logged-in, the server responds with a list of supported commands and a brief description of each. See the example in the following figure.

NOTE: The list of available commands depends on your system configuration.



```
Enter password:
Welcome to the server.
You can type 'help' at any time for the following help list.
Commands:
  getDateTime      gets the date and time
  help             gives the listing of available commands
  payloadQuery     Queries the payload for this robot
  payloadSlotCount Queries for number of payload slots
  queueCancel      Cancels an item by type and value
  queuePickup      Queues a pickup goal for any appropriate robot
  queuePickupDropoff Queues a pickup dropoff goal pair for any appropriate robot to do
  queueQuery       Queries the queue by type and value
  queueShow        Shows the Queue
  queueShowRobot   Shows the status of all the robots
```

Figure 3-2. Example Command List after Login

5. If needed, you can enter the `echo off` command to prevent your input from echoing

(typing double characters).

6. When you are finished, use the **quit** command to close the connection.

After you connect to ARCL, you can execute any of the ARCL commands available. For a complete list of the different ARCL commands and their arguments, refer to ARCL Command Reference on page 27.

ARCL supports multiple client/server connections through the TCP/IP socket. However, commands and query responses are connection-specific. For example, you can have two Telnet clients connected. However, only the one that requested a **oneLineStyle** response, actually receives the status message.

Chapter 4: Using the ARCL Commands

After you have established a connection to the ARCL server, you are ready to operate and monitor the AMR using the ARCL commands. The following topics discuss the use of these commands for certain tasks. To view an alphabetical list and description of each ARCL command, refer to ARCL Command Reference on page 27.

This section discusses the following topics:

4.1 Understanding the Commands	20
Document Conventions	20
Command Notes	21
Data Types	22
Status and Error Messages	22
Status Conditions	23
4.2 Using the Queuing Commands	25
4.3 Working With Payloads	25

The ARCL command set is evolutionary and backward compatible. To see added commands, consult the ARCL help list when connecting with a new ARAM version.

4.1 Understanding the Commands

This section describes the document conventions, command notes, and status and error messages.

The commands are discussed by task in this chapter. To view commands presented in alphabetical order, see the ARCL Command Reference on page 27.

Document Conventions

Command name (shortcut: *cn*)

The command can be invoked with its full name or, in some cases, with a shortcut. When there is a shortcut, it will be listed in parentheses after the command name in the title of the command description. The syntax, usage, and parameters are the same, whether the full command name or the shortcut is used.

Syntax

The ARCL commands are not case sensitive. In this guide, commands are shown in mixed case and bold type. Required parameters are shown in angled brackets and regular type; whereas, optional parameters are shown in square brackets [] and regular type. For example:

queuePickup <goalName> [priority] [jobId]

In this example, the <goalName> parameter is required; the [priority] and [jobId] parameters are optional.

Usage Considerations

This section describes any special considerations that must be followed when using the command. It also describes where the command can be used, as follows:

- This ARCL command is available only on the Fleet Manager.

NOTE: This is the only usage consideration for commands noted in this manual.

ARAM Settings

This section lists any ARAM settings that must be enabled to use the command.

Parameters

This section describes each of the required and optional command parameters (such as goal-name, routename, echo, etc.).

Responses

This section shows the information returned by the command.

Details

This section covers additional information about the functions of the commands. If there is no additional information, this section is not present.

Examples

This section provides examples of correctly-formatted command lines.

Related Commands

This section lists additional commands that are similar or often used with this command.

Command Notes

Below are some helpful notes to remember when using ARCL commands:

- ARCL responds with the command's syntax if you omit any or all required parameters.
- Extraneous parameters are ignored.
- ARCL limits commands to a maximum of 5,000 ASCII characters.
- As a general rule, use double quotes for string parameters, especially if there are spaces in the string.
- Mistyped Telnet commands and parameters cannot be edited on the command line. You have to completely re-type the command.
- Mistyped or non-existent commands are rejected with the response, "Unknown command".
- Although commands are not case-sensitive, some parameters are case-sensitive.

Data Types

The following table shows all of the available ARCL data types (not all of these may apply to a particular command):

Parameter	Data Type	Max Length/Range
cancelType	string	max length: 127 characters
cancelValue	string	max length: 127 characters
DROPOFFgoalName	string	max length: 127 characters
DROPOFFpriority	integer (signed long)	range: -2147483648 to 2147483647
echoString ²	string	max length: 127 characters
goalName	string	max length: 127 characters
jobId ²	string	max length: 127 characters
payload slot number	integer (signed long)	range: 1 to 2147483647
payload slot string ¹	string	max length: 127 characters
PICKUPgoalName	string	max length: 127 characters
PICKUPpriority	integer (signed long)	range: -2147483648 to 2147483647
priority	integer (signed long)	range: -2147483648 to 2147483647
queryType	string	max length: 127 characters
queryValue	string	max length: 127 characters
reason ²	string	max length: 127 characters
robotName ¹	string	max length: 127 characters
¹ These parameters support spaces, and need to be enclosed in quotes if they include spaces.		
² These parameters do not support spaces or double quotes.		

Status and Error Messages

ARCL sends important status updates to the connected client for certain commands, such as **queuePickup** goalName. For example, when the job is first received, then the following is sent to the client:

```
queuepickup goal "<goalName>" with priority 10, id PICKUP138 and
jobId JOB138 successfully queued
```

When the job has been completed, this update message is sent:

```
QueueUpdate: PICKUP138 JOB138 10 Completed None Goal "<goalName>"
"robotName" 04/08/2013 13:46:34 0
```

If ARCL is unable to execute the command because of a command sequence error, a non-existent filename, or because a feature was not set up properly, a `SetUpError` is displayed. For example, if you attempt to execute `Play` and the filename specified does not exist, the following error is displayed:

```
SetUpError: There was nothing given to replay.
```

All other argument errors result in a two-line ARCL response, with two distinct error messages, such as the following:

```
CommandError: queuePickup goal6
CommandErrorDescription: queuePickup no such goal "goal6"
```

ARCL sends status update messages without prompting, for example, when there are changes in a AMR's or a job's state.

Status Conditions

The following table shows the possible AMR and job status conditions:

Status	Substatus
Pending	None
Pending	AssignedRobotOffLine
Pending	NoMatchingRobotForLinkedJob
Pending	NoMatchingRobotForOtherSegment
Pending	NoMatchingRobot
Pending	ID_PICKUPxx <where PICKUPxx is the jobSegment ID for which this Job Segment is waiting>
Pending	ID_DROPOFFxx <where DROPOFFxx is the jobSegment ID for which this Job Segment is waiting>
Available	Available
Available	Parking
Available	Parked
Available	DockParking
Available	DockParked
Interrupted	None
InProgress	UnAllocated
InProgress	Allocated
InProgress	BeforePickup
InProgress	BeforeDropoff
InProgress	BeforeEvery

Status	Substatus
InProgress	Before
InProgress	Buffering
InProgress	Buffered
InProgress	Driving
InProgress	After
InProgress	AfterEvery
InProgress	AfterPickup
InProgress	AfterDropoff
Completed	None
Cancelling	None
Cancelled	None
Cancelling	<application_supplied_cancelReason_string>
Cancelled	<application_supplied_cancelReason_string>
BeforeModify	None
InterruptedByModify	None
AfterModify	None
UnAvailable	NotUsingEnterpriseManager
UnAvailable	UnknownBatteryType
UnAvailable	ForcedDocked
UnAvailable	Lost
UnAvailable	EStopPressed
UnAvailable	Interrupted
UnAvailable	InterruptedButNotYetIdle
UnAvailable	Fault_Driving_Application_<application_supplied_string>
UnAvailable	OutgoingARCLConnLost
UnAvailable	Parking
UnAvailable	DockParking
UnAvailable	ModeIsLocked

4.2 Using the Queuing Commands

The ARCL queuing commands are used with the Fleet Manager. They allow you to request an AMR to drive to a goal (for example, for a pickup) and then drive to another goal (for example, for a dropoff).

queueCancel Command (shortcut: qc) on page 39

queueModify Command (shortcut: qmod) on page 42

queueMulti Command (shortcut: qm) on page 47

queuePickup Command (shortcut: qp) on page 50

queuePickupDropoff Command (shortcut: qpd) on page 53

queueQuery Command (shortcut: qq) on page 58

queueShow Command (shortcut: qs) on page 61

queueShowCompleted Command (shortcut: qsc) on page 63

queueShowRobot Command (shortcut: qsr) on page 65

4.3 Working With Payloads

Using the ARCL payload commands, you can view the number of slots on an AMR and see what objects the AMR is carrying.

The following commands are supported:

- payloadQuery Command (shortcut: pq) on page 31
- payloadSlotCount Command (shortcut: psc) on page 34

Slots represent containers where the objects (payload) are carried on top of the AMR.

The default number of slots is 4. Note that slot numbering starts at 1. A slot count of 0 indicates there is no payload.

Chapter 5: ARCL Command Reference

This section provides a description of each command in the ARCL command set. The command descriptions are provided in alphabetical order.

5.1 getDateTime Command	30
Syntax	30
Parameters	30
Examples	30
5.2 payloadQuery Command (shortcut: pq)	31
Syntax	31
Parameters	31
Responses	31
Details	31
Examples	31
Related Commands	33
5.3 payloadSlotCount Command (shortcut: psc)	34
Syntax	34
Parameters	34
Responses	34
Examples	34
Related Commands	35
5.4 queryFaults Command (shortcut: qf)	36
Syntax	36
Usage Considerations	36
Parameter	36
Responses	36
Details	36
Example	36
Related Commands	38
5.5 queueCancel Command (shortcut: qc)	39
Syntax	39
Parameters	39
Responses	39
Details	40
Examples	40

Related Commands	41
5.6 queueModify Command (shortcut: qmod)	42
Syntax	42
ARAM Settings	42
Parameters	42
Responses	43
Examples	44
Related Commands	46
5.7 queueMulti Command (shortcut: qm)	47
Syntax	47
ARAM Settings	47
Parameters	47
Responses	48
Details	48
Examples	48
Related Commands	49
5.8 queuePickup Command (shortcut: qp)	50
Syntax	50
ARAM Settings	50
Parameters	50
Responses	50
Details	51
Examples	51
Related Commands	51
5.9 queuePickupDropoff Command (shortcut: qpd)	53
Syntax	53
Parameters	53
Responses	53
Details	54
Examples	54
Related Commands	56
5.10 queueQuery Command (shortcut: qq)	58
Syntax	58
Parameters	58
Responses	58
Details	59
Examples	59

Related Commands	59
5.11 queueShow Command (shortcut: qs)	61
Syntax	61
Usage Considerations	61
Parameters	61
Responses	61
Details	61
Examples	62
Related Commands	62
5.12 queueShowCompleted Command (shortcut: qsc)	63
Syntax	63
Usage Considerations	63
Parameters	63
Returns	63
Details	63
Examples	64
Related Commands	64
5.13 queueShowRobot Command (shortcut: qsr)	65
Syntax	65
Usage Considerations	65
Parameters	65
Responses	65
Details	65
Examples	65
Related Commands	66

5.1 getDateTime Command

Returns the system date and time.

Syntax

`getDateTime`

Parameters

This command does not have any parameters.

Examples

To view the current system date and time, enter:

```
getdatetime
```

The command returns:

```
DateTime: 05/03/2012 04:48:55
```

5.2 payloadQuery Command (shortcut: pq)

Queries the payload for a specified AMR, a specified AMR and slot, or all connected AMRs that have a payload configured.

Syntax

```
payloadQuery [robotName or "default"] [slotNumber or "default"] [echoString]
```

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
robotName	Enter the name of the AMR to display its slot information.
slotNumber	Enter the slot number to display its information. Requires a value in the previous parameter.
echoString	An optional string that is appended to each line of the results. Requires a value in the previous parameter.

Responses

The command returns the payload query in the following format:

```
PayloadQuery: "<robotName>" <slotNumber> "<description>" <date> <time>
"[echoString]"
```

The date and time are assigned by the system when the slot payload is set on the AMR.

Details

The payloadQuery command can be used to view the payload information for:

- all slots on all AMRs
- a specified slot on a AMR
- all slots on a specified AMR

Slot numbering starts at 1 (there is no slot 0).

An optional string can be specified, which will be appended to each line of the results.

Examples

In the example below, AMR 21 is carrying books and glasses. To view what AMR 21 is carrying, enter the following command:

```
payloadQuery 21
```

The command returns:

```
PayloadQuery: "21" 1 "Books" 05/07/2012 21:11:33 ""
PayloadQuery: "21" 2 "Glasses" 05/07/2012 21:15:11 ""
PayloadQuery: "21" 3 "Empty" None None ""
PayloadQuery: "21" 4 "Empty" None None ""
EndPayloadQuery
```

The following example displays all of the defined slots on all AMRs connected to the Fleet Manager. The command is entered without the robotName argument.

```
payloadQuery
PayloadQuery: "21" 1 "Books" 05/07/2012 21:11:33 ""
PayloadQuery: "21" 2 "Glasses" 05/07/2012 21:14:51 ""
PayloadQuery: "21" 3 "Empty" None None ""
PayloadQuery: "21" 4 "Empty" None None ""
PayloadQuery: "22" 1 "Empty" None None ""
PayloadQuery: "22" 2 "Empty" None None ""
PayloadQuery: "22" 3 "stuff" 09/10/2012 12:14:14 ""
PayloadQuery: "22" 4 "Empty" None None ""
PayloadQuery: "23" 1 "Paper" 09/10/2012 12:17:23 ""
PayloadQuery: "23" 2 "Empty" None None ""
PayloadQuery: "23" 3 "Bread" 09/10/2012 12:23:39 ""
PayloadQuery: "23" 4 "Empty" None None ""
EndPayloadQuery
```

The following example displays all of the defined slots on all AMRs and echoes the string "hello":

```
payloadquery default default hello
PayloadQuery: "31" 1 "boxes" 05/07/2012 21:11:33 hello
PayloadQuery: "31" 2 "Bags" 05/07/2012 21:10:53 hello
PayloadQuery: "31" 3 "Hats" 09/10/2012 12:14:14 hello
PayloadQuery: "31" 4 "Gloves" 09/10/2012 12:23:39 hello
PayloadQuery: "32" 1 "Empty" None None hello
PayloadQuery: "32" 2 "Empty" None None hello
```



```
PayloadQuery: "32" 3 "Empty" None None hello
PayloadQuery: "32" 4 "Empty" None None hello
PayloadQuery: "33" 1 "Empty" None None hello
PayloadQuery: "33" 2 "Empty" None None hello
PayloadQuery: "33" 3 "Empty" None None hello
PayloadQuery: "33" 4 "Empty" None None hello
PayloadQuery: "34" 1 "Empty" None None hello
PayloadQuery: "34" 2 "Empty" None None hello
PayloadQuery: "34" 3 "Empty" None None hello
PayloadQuery: "34" 4 "Empty" None None hello
PayloadQuery: "35" 1 "Empty" None None hello
PayloadQuery: "35" 2 "Empty" None None hello
PayloadQuery: "35" 3 "Empty" None None hello
PayloadQuery: "35" 4 "Empty" None None hello
PayloadQuery: "36" 1 "Empty" None None hello
PayloadQuery: "36" 2 "Empty" None None hello
PayloadQuery: "36" 3 "Empty" None None hello
PayloadQuery: "36" 4 "Empty" None None hello
EndPayloadQuery
```

Related Commands

[payloadSlotCount Command \(shortcut: psc\) on page 34](#)

5.3 payloadSlotCount Command (shortcut: psc)

Displays the slot count on a specific AMR or on all AMRs.

Syntax

```
payloadSlotCount [robotName or "default"] [echoString]
```

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
robotName	Enter the name of the AMR to display its slot count. To view the slot counts for all connected AMRs, enter the command with no parameter or enter "default".
echoString	An optional string that is appended to each line of the results. Requires a value in the previous parameter.

Responses

The command returns the slot count in the following format:

```
PayloadSlotCount: "<robotName>" <slotCount> <date> <time> "[echoString]"
```

The date and time are assigned by the system.

Examples

To view the slot count for AMR 21, enter the following command:

```
payloadslotcount 21
```

The command returns:

```
PayloadSlotCount: "21" 4 ""
```

```
EndPayloadSlotCount
```

The following example displays the slot counts on all AMRs connected to the Fleet Manager. The command is entered without the robotName argument.

```
payloadSlotCount
```

```
PayloadSlotCount: "21" 4 04/27/2012 06:37:33 ""
```

```
PayloadSlotCount: "22" 5 04/27/2012 08:37:33 ""
```

```
PayloadSlotCount: "23" 4 04/27/2012 07:37:33 ""  
EndPayloadSlotCount
```

Related Commands

[payloadQuery Command \(shortcut: pq\) on page 31](#)

5.4 queryFaults Command (shortcut: qf)

Displays the faults associated with the specified AMR.

Syntax

```
queryFaults [robotName or "default"] [echoString]
```

Usage Considerations

Displays all faults on the specified AMR. Displays faults on all AMRs if the robotName parameter is omitted.

Parameter

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
robotName	Enter the name of the AMR. To view all the AMRs connected to the Fleet Manager, omit this parameter or enter "default".
echoString	An optional string that is appended to each line of the results.

Responses

The command returns the following for a pending item:

```
RobotFaultQuery: <robotName> <faultName> <faultShortDescription> <faultLongDe-
scription> <bool:drivingFault> <bool:crit-
icalFault><bool:applicationFault><bool:clearedOnGo><bool:
clearedOnAcknowledgement> <echoString>
EndQueryFaults
```

Details

The queryFaults command provides a listing of all faults for the specified AMR, or all faults for all AMRs connected to the Fleet Manager if no AMR is specified.

Example

```
queryfaults robot1
```

```
RobotFaultQuery: "robot1" Fault_Critical_Application fault1 "shortdesc" "long-
desc" false true true false false ""
EndQueryFaults
```

```
queryfaults robot1 echoit
```

```
RobotFaultQuery: "robot1" Fault_Critical_Application fault1 "shortdesc" "long-
```

```
desc" false true true false false echoit
EndQueryFaults
```

```
queryfaults
RobotFaultQuery: "robot2" Fault_Driving_Application fault2 "shortd" "longd" true
false true false false ""
RobotFaultQuery: "robot1" Fault_Critical_Application fault1 "shortdesc" "long-
desc" false true true false false ""
EndQueryFaults
```

```
queryfaults
RobotFaultQuery: "guiabot_2010_09_20" Fault_Driving_Application fault2 "shortd"
"longd" true false true false false ""
RobotFaultQuery: "showpatrolbot1" Fault EncoderDegraded "Encoder degraded" "The
robot's encoders may be degraded" false true false false false ""
RobotFaultQuery: "showpatrolbot1" Fault_Driving EncoderFailed "Encoder failed"
"The robot's encoders have failed, turn off the robot and contact your robot pro-
vider for maintenance" true true false false false ""
RobotFaultQuery: "showpatrolbot1" Fault_Critical GyroFault "Gyro fault" "The
robot's gyro has had a critical fault, you may power cycle the robot and con-
tinue using it, but you should also contact your robot provider for maintenance"
true true false false false ""
RobotFaultQuery: "showpatrolbot1" Fault_Critical OverTemperatureAnalog "Robot
overheated (analog)" "The robot is too hot (measured by analog) and will shut
down shortly" false true false false false ""
RobotFaultQuery: "showpatrolbot1" Fault_Critical UnderVoltage "Robot battery
critically low" "The robot battery is critically low and will shut down shortly"
false true false false false ""
RobotFaultQuery: "showpatrolbot1" Fault_Critical_Application fault1 "shortdesc"
"longdesc" false true true false false ""
RobotFaultQuery: "showpatrolbot1" Fault_Application fault3 "short" "long" false
true true false false ""
EndQueryFaults
```

The broadcast messages to EM ARCL when AMRs set/clear faults will have the following formats:

```
RobotFault: " showpatrolbot1" Fault_Application fault3 "short" "long" false true
true false false
RobotFault: " showpatrolbot1" Fault_Driving_Application fault2 "shortd" "longd"
true false true false false
RobotFault: " showpatrolbot1" Fault_Critical OverTemperatureAnalog "Robot over-
heated (analog)" "The robot is too hot (measured by analog) and will shut down
shortly" false true false false false
RobotFault: " showpatrolbot1" Fault_Critical UnderVoltage "Robot battery crit-
ically low" "The robot battery is critically low and will shut down shortly"
false true false false false
RobotFault: " showpatrolbot1" Fault EncoderDegraded "Encoder degraded" "The
robot's encoders may be degraded" false true false false false
RobotFault: " showpatrolbot1" Fault_Driving EncoderFailed "Encoder failed" "The
robot's encoders have failed, turn off the robot and contact your robot provider
for maintenance" true true false false false
RobotFault: " showpatrolbot1" Fault_Critical GyroFault "Gyro fault" "The robot's
gyro has had a critical fault, you may power cycle the robot and continue using
it, but you should also contact your robot provider for maintenance" true true
false false false
RobotFault: "Sim2" Fault_Application_ClearedOnAcknowledgement f1 "s" "l" false
```

5.4 queryFaults Command (shortcut: qf)

```
false true false true
RobotFaultCleared: "showpatrolbot1" Fault_EncoderDegraded "Encoder degraded"
"The robot's encoders may be degraded" false true false false false
RobotFaultCleared: "showpatrolbot1" Fault_Driving EncoderFailed "Encoder failed"
"The robot's encoders have failed, turn off the robot and contact your robot pro-
vider for maintenance" true true false false false
RobotFaultCleared: "showpatrolbot1" Fault_Critical GyroFault "Gyro fault" "The
robot's gyro has had a critical fault, you may power cycle the robot and con-
tinue using it, but you should also contact your robot provider for maintenance"
true true false false false
RobotFaultCleared: "showpatrolbot1" Fault_Critical OverTemperatureAnalog "Robot
overheated (analog)" "The robot is too hot (measured by analog) and will shut
down shortly" false true false false
RobotFaultCleared: "showpatrolbot1" Fault_Critical UnderVoltage "Robot battery
critically low" "The robot battery is critically low and will shut down shortly"
false true false false false
RobotFaultCleared: "showpatrolbot1" Fault_Critical_Application fault1 "short-
desc" "longdesc" false true true false false
RobotFaultCleared: "Sim2" Fault_Application_ClearedOnAcknowledgement fl "s" "1"
false false true false true
EndQueryFaults
```

Related Commands

[queueCancel Command \(shortcut: qc\) on page 39](#)

[queuePickup Command \(shortcut: qp\) on page 50](#)

[queuePickupDropoff Command \(shortcut: qpd\) on page 53](#)

[queueModify Command \(shortcut: qmod\) on page 42](#)

[queueMulti Command \(shortcut: qm\) on page 47](#)

[queueQuery Command \(shortcut: qq\) on page 58](#)

[queueShowRobot Command \(shortcut: qsr\) on page 65](#)

[queueShowCompleted Command \(shortcut: qsc\) on page 63](#)

5.5 queueCancel Command (shortcut: qc)

Cancels a queued request for an AMR by type or value.

Syntax

```
queueCancel <type> <value> [echoString or "default"] [reason]
```

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
type	Enter the type of job. Valid types are: <ul style="list-style-type: none"> • id = the pickup or dropoff identification • jobId = the job identification • robotName = the AMR name • status = the item status.
value	Enter the value that corresponds with the type used: For id, enter the pickup or dropoff identification, for example: PICKUP2 For jobId, enter the job identification, for example: JOB2 For robotName, enter the AMR name, for example: robot_34 For status, enter one of the following values: <ul style="list-style-type: none"> • inprogress = cancels a job with an InProgress status. • pending = cancels a job with a Pending status. • interrupted = cancels a job with an Interrupted status.
echoString	An optional string that is appended to each line of the results. Use "default" when you don't want an echoString, but you do want to show a reason.
reason	An optional string that can be used to provide a reason for the cancellation.

Responses

The command returns the following for a pending item:

```
queuencancel cancelling <cancelType> <cancelValue> <echoString> <reason> from
queue
QueueUpdate: <id> <jobId> <priority> <status = Cancelled> <subStatus = reason_
or_None> Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <com-
pletedDate> <completedTime> <echoString>
```

The command returns the following for an in-progress item:

```
queuecancel cancelling <cancelType> <cancelValue> <echoString> from queue
QueueUpdate: <id> <jobId> <priority> <status = Cancelling> <subStatus = reason_
or_None> Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <com-
pletedDate = None> <completedTime = None> <echoString>
QueueUpdate: <id> <jobId> <priority> <status = Interrupted> <subStatus = reason_
or_None> Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <com-
pletedDate = None> <completedTime = None> <failedCount>
QueueUpdate: <id> <jobId> <priority> <status = Cancelled> <subStatus = reason_
or_None> Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <com-
pletedDate> <completedTime> <failedCount>
```

The reported jobId was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the jobId.

For details on the status conditions, see Status Conditions on page 23.

Details

The queueCancel command is used to cancel a queued AMR request. The request can be canceled by type (such as the AMR name or job identification) or by the request status.

An optional string can be specified, which will be appended to each line of the results.

Examples

In the following example, a pending item in the queue is canceled.

```
queuepickup x
queuepickup goal "x" with priority 10, id PICKUP1 and jobId JOB1 successfully
queued
QueueUpdate: PICKUP1 JOB1 10 Pending None Goal "x" "None" 04/15/2015 6:32:47
None None 0
queuecancel jobId job1
QueueUpdate cancelling "jobid" "job1" "" "None" from queue
QueueUpdate: PICKUP1 JOB1 10 Cancelled None Goal "x" "None" 04/15/2015 6:32:47
04/15/2015 6:32:53 ""
```

In the following example, a request that is in progress is canceled.

```
QueueUpdate: PICKUP8 JOB8 10 InProgress None Goal "w20" MT-490 12/16/2014
13:19:07 None None
queuecancel goal w20 abc
QueueUpdate: PICKUP8 JOB8 10 Cancelling None Goal "w20" None 12/16/2014 13:19:07
None None abc
QueueUpdate: PICKUP8 JOB8 10 Interrupted None Goal "w20" None 12/16/2014
13:19:07 None None
QueueUpdate: PICKUP8 JOB8 10 Cancelled None Goal "w20" None 12/16/2014 13:19:07
12/16/2014 13:19:13
```


In the following example, a request that is in progress is canceled. The cancel request includes a reason for the cancellation but no echo.

```
QueueUpdate: PICKUP2 JOB2 10 InProgress After Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:04:59 None None 0

queuecancel id pickup2 default reason

queuecancel cancelling "id" "pickup2" "" "reason" from queue
QueueUpdate: PICKUP2 JOB2 10 Cancelling reason Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:04:59 None None ""
QueueUpdate: PICKUP2 JOB2 10 Interrupted None Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:04:59 None None 0
QueueUpdate: PICKUP2 JOB2 10 Cancelled reason Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:04:59 01/21/2014 15:05:40 0
```

In the following example, a request that is in progress is canceled. The cancel request includes no reason for the cancellation and no echo.

```
QueueUpdate: PICKUP3 JOB3 10 InProgress After Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:07:58 None None 0

queuecancel jobid job3

QueueUpdate cancelling "jobid" "job3" "" "None" from queue
QueueCancel: PICKUP3 JOB3 10 Cancelling None Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:07:58 None None ""
QueueUpdate: PICKUP3 JOB3 10 Interrupted None Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:07:58 None None 0
QueueUpdate: PICKUP3 JOB3 10 Cancelled None Goal "w20" "guiabot_2010_09_20"
01/21/2014 15:07:58 01/21/2014 15:08:32 0
```

Related Commands

- queryFaults Command (shortcut: qf) on page 36
- queueMulti Command (shortcut: qm) on page 47
- queuePickup Command (shortcut: qp) on page 50
- queuePickupDropoff Command (shortcut: qpd) on page 53
- queueQuery Command (shortcut: qq) on page 58
- queueShow Command (shortcut: qs) on page 61
- queueShowCompleted Command (shortcut: qsc) on page 63
- queueShowRobot Command (shortcut: qsr) on page 65

5.6 queueModify Command (shortcut: qmod)

Allows modification of goal and priority for job segments in these job types:

- PickupDropoff
- Pickups
- Dropoffs
- Swaps
- QueueMulti

Allows modification of segments in these states:

- Pending job segments
- InProgress jobs up to and including “InProgressDriving”, but not after

Changing the priority for the first segment in a job may change the order in which it gets assigned. Changing the priority of other segments in the job will never change the order in which the job is assigned.

The queue time for a job will never be changed as a result of a queueModify command.

Changing the shared goal in a swap will break the link between the two jobs. Changing the other goals in the swap will not break the link.

Modified jobs will be candidates for swaps. The linking would occur immediately following the modify.

Syntax

```
queueModify <id> <type> <value>
```

ARAM Settings

In order to use this feature, you have to explicitly enable it in the MobilePlanner software, by setting the EnterpriseQueuing argument in the Enterprise Features section of the **Configuration > Enterprise** tab.

Parameters

The queueModify arguments are described in the table below.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
<id>	Enter the string id for the job segment you wish to modify (either PICKUPxx or DROPOFFxx)
<type>	Enter the type of modification. Valid types are: <ul style="list-style-type: none"> goal = the goal identification priority = the priority level
<value>	Enter the value that corresponds with the type used: For goal, enter the goal identification, for example: goal_1 For priority, enter the priority level, for example: 10

Responses

Returns (for goal modify of a pending item)

```

queuemodify modifying id <id> goal <"modifiedGoal">
QueueUpdate: <id> <jobId> <priority> BeforeModify None Goal <goal> "None"
<queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <priority> AfterModify None Goal <modifiedGoal> "None"
<queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <priority> Pending None Goal <modifiedGoal> "None"
<queuedDate> <queuedTime> None None 0

```

Returns (for priority modify of a pending item)

```

queuemodify modifying id <id> priority <modifiedpriority>
QueueUpdate: <id> <jobId> <priority> BeforeModify None Goal <goal> "None"
<queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <modifiedPriority> AfterModify None Goal <goal> "None"
<queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <modifiedPriority> Pending None Goal <modifiedGoal>
"None" <queuedDate> <queuedTime> None None 0

```

Returns (for goal modify of an in-progress item)

```

queuemodify modifying id <id> goal <modifiedGoal>
QueueUpdate: <id> <jobId> <priority> BeforeModify Driving Goal <goal> <robot>
<queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <priority> InterruptedByModify None Goal <goal>
<robot> <queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <priority> AfterModify None Goal <modifiedGoal>
<robot> <queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <priority> Pending None Goal <modifiedGoal> "None"
<queuedDate> <queuedTime> None None 0

```

Returns (for priority modify of an in-progress item)

```

queuemodify modifying id <id> priority <modifiedPriority>
QueueUpdate: <id> <jobId> <priority> BeforeModify Driving Goal <goal> <robot>
<queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <priority> InterruptedByModify None Goal <goal>
<robot> <queuedDate> <queuedTime> None None 0
QueueUpdate: <id> <jobId> <modifiedPriority> AfterModify None Goal <goal>
<robot> <queuedDate> <queuedTime> None None 0

```

5.6 queueModify Command (shortcut: qmod)

```
QueueUpdate: <id> <jobId> <modifiedPriority> Pending None Goal <goal> "None"  
<queuedDate> <queuedTime> None None 0
```

The reported jobId was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the jobId.

Examples

Example #1 – goal modify of a pending item:

```
queuePickup t  
queuePickup goal "t" with priority 10 id PICKUP5 and jobId JOB5 successfully  
queued  
QueueUpdate: PICKUP5 JOB5 10 Pending None Goal "t" "None" 03/25/2015 07:36:58  
None None 0  
queuemodify pickup5 goal w20  
queuemodify modifying id pickup5 goal "w20"  
QueueUpdate: PICKUP5 JOB5 10 BeforeModify None Goal "t" "None" 03/25/2015  
07:36:58 None None 0  
QueueUpdate: PICKUP5 JOB5 10 AfterModify None Goal "w20" "None" 03/25/2015  
07:36:58 None None 0  
QueueUpdate: PICKUP5 JOB5 10 Pending None Goal "w20" "None" 03/25/2015 07:36:58  
None None 0
```

```
queueDropoff y  
queueDropoff attempting to queue goal "y" using default priority  
queueDropoff goal "y" with priority 20 id DROPOFF6 and jobId JOB6 successfully  
queued  
QueueUpdate: DROPOFF6 JOB6 20 Pending None Goal "y" "robotOne" 03/25/2015  
07:38:09 None None 0  
queuemodifylocal dropoff6 goal x  
queuemodifylocal modifying id dropoff6 goal "x"  
QueueUpdate: DROPOFF6 JOB6 20 BeforeModify None Goal "y" "robotOne" 03/25/2015  
07:38:09 None None 0  
QueueUpdate: DROPOFF6 JOB6 20 AfterModify None Goal "x" "robotOne" 03/25/2015  
07:38:09 None None 0  
QueueUpdate: DROPOFF6 JOB6 20 Pending None Goal "x" "robotOne" 03/25/2015  
07:38:09 None None 0
```

Example #2 – priority modify of a pending item:

```
queueDropoff w20  
queueDropoff attempting to queue goal "w20" using default priority  
queueDropoff goal "w20" with priority 20 id DROPOFF7 and jobId JOB7 successfully  
queued  
QueueUpdate: DROPOFF7 JOB7 20 Pending None Goal "w20" "robotOne" 03/25/2015  
07:39:01 None None 0  
queuemodifylocal dropoff7 priority 22  
queuemodifylocal modifying id dropoff7 priority 22  
QueueUpdate: DROPOFF7 JOB7 20 BeforeModify None Goal "w20" "robotOne" 03/25/2015  
07:39:01 None None 0  
QueueUpdate: DROPOFF7 JOB7 22 AfterModify None Goal "w20" "robotOne" 03/25/2015  
07:39:01 None None 0  
QueueUpdate: DROPOFF7 JOB7 22 Pending None Goal "w20" "robotOne" 03/25/2015  
07:39:01 None None 0
```

```
queuePickup v  
queuePickup goal "v" with priority 10 id PICKUP8 and jobId JOB8 successfully  
queued
```

```

QueueUpdate: PICKUP8 JOB8 10 Pending None Goal "v" "None" 03/25/2015 07:40:24
None None 0
queuemodify pickup8 priority 6
queuemodify modifying id pickup8 priority 6
QueueUpdate: PICKUP8 JOB8 10 BeforeModify None Goal "v" "None" 03/25/2015
07:40:24 None None 0
QueueUpdate: PICKUP8 JOB8 6 AfterModify None Goal "v" "None" 03/25/2015 07:40:24
None None 0
QueueUpdate: PICKUP8 JOB8 6 Pending None Goal "v" "None" 03/25/2015 07:40:24
None None 0

```

Example #3 – goal modify of an inProgress item:

```

queuePickup x
queuePickup goal "x" with priority 10 id PICKUP9 and jobId JOB9 successfully
queued
QueueUpdate: PICKUP9 JOB9 10 Pending None Goal "x" "None" 03/25/2015 07:47:21
None None 0
QueueUpdate: PICKUP9 JOB9 10 InProgress UnAllocated Goal "x" "robotTwo"
03/25/2015 07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 InProgress Allocated Goal "x" "robotTwo" 03/25/2015
07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 InProgress Driving Goal "x" "robotTwo" 03/25/2015
07:47:21 None None 0
queuemodify pickup9 goal y
queuemodify modifying id pickup9 goal "y"
QueueUpdate: PICKUP9 JOB9 10 BeforeModify Driving Goal "x" "robotTwo" 03/25/2015
07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 InterruptedByModify None Goal "x" "robotTwo"
03/25/2015 07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 AfterModify None Goal "y" "robotTwo" 03/25/2015
07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 Pending None Goal "y" "None" 03/25/2015 07:47:21
None None 0
QueueUpdate: PICKUP9 JOB9 10 InProgress UnAllocated Goal "y" "robotTwo"
03/25/2015 07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 InProgress Allocated Goal "y" "robotTwo" 03/25/2015
07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 InProgress Driving Goal "y" "robotTwo" 03/25/2015
07:47:21 None None 0
QueueUpdate: PICKUP9 JOB9 10 Completed None Goal "y" "robotTwo" 03/25/2015
07:47:21 03/25/2015 07:48:00 0

```

Example #4 – priority modify of an inProgress item:

```

queuePickup t
queuePickup goal "t" with priority 10 id PICKUP10 and jobId JOB10 successfully
queued
QueueUpdate: PICKUP10 JOB10 10 Pending None Goal "t" "None" 03/25/2015 07:49:34
None None 0
QueueUpdate: PICKUP10 JOB10 10 InProgress UnAllocated Goal "t" "robotTwo"
03/25/2015 07:49:34 None None 0
QueueUpdate: PICKUP10 JOB10 10 InProgress Allocated Goal "t" "robotTwo"
03/25/2015 07:49:34 None None 0
QueueUpdate: PICKUP10 JOB10 10 InProgress Driving Goal "t" "robotTwo" 03/25/2015
07:49:34 None None 0
queuemodify pickup10 priority 13
queuemodify modifying id pickup10 priority 13
QueueUpdate: PICKUP10 JOB10 10 BeforeModify Driving Goal "t" "robotTwo"
03/25/2015 07:49:34 None None 0

```

5.6 queueModify Command (shortcut: qmod)

```
QueueUpdate: PICKUP10 JOB10 10 InterruptedByModify None Goal "t" "robotTwo"
03/25/2015 07:49:34 None None 0
QueueUpdate: PICKUP10 JOB10 13 AfterModify None Goal "t" "robotTwo" 03/25/2015
07:49:34 None None 0
QueueUpdate: PICKUP10 JOB10 13 Pending None Goal "t" "None" 03/25/2015 07:49:34
None None 0
QueueUpdate: PICKUP10 JOB10 13 InProgress UnAllocated Goal "t" "robotTwo"
03/25/2015 07:49:34 None None 0
QueueUpdate: PICKUP10 JOB10 13 InProgress Allocated Goal "t" "robotTwo"
03/25/2015 07:49:34 None None 0
QueueUpdate: PICKUP10 JOB10 13 InProgress Driving Goal "t" "robotTwo" 03/25/2015
07:49:34 None None 0
QueueUpdate: PICKUP10 JOB10 13 Completed None Goal "t" "robotTwo" 03/25/2015
07:49:34 03/25/2015 07:49:46 0
```

Related Commands

[queueCancel Command \(shortcut: qc\) on page 39](#)

[queuePickup Command \(shortcut: qp\) on page 50](#)

[queuePickupDropoff Command \(shortcut: qpd\) on page 53](#)

[queueMulti Command \(shortcut: qm\) on page 47](#)

[queueQuery Command \(shortcut: qq\) on page 58](#)

[queueShow Command \(shortcut: qs\) on page 61](#)

[queueShowRobot Command \(shortcut: qsr\) on page 65](#)

[queryFaults Command \(shortcut: qf\) on page 36](#)

5.7 queueMulti Command (shortcut: qm)

Queues the AMR for multiple pickups and dropoffs at multiple goals.

Syntax

```
queueMulti <number of goals> <number of fields per goal> <goal1> <goal1 args> <goal2>
<goal2 args> ... <goalN> <goalN args> [jobid]
```

ARAM Settings

In order to use this feature, you have to explicitly enable it in the MobilePlanner software, by setting the EnterpriseQueuing argument in the Enterprise Features section of the **Configuration** > **Enterprise** tab.

Parameters

The queueMulti arguments are described in the table below.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
number of goals	Enter the number of goals where you want the AMR to go. Up to 50 goals are supported.
number of fields per goal	Enter the number of fields to be used for all goals. Two fields are supported, in this order: <pickup dropoff> <priority>.
goal1	Enter the name of the first goal.
goal1 args	Enter the arguments associated with the first goal in the form: <pickup dropoff> <priority or "default"> The first goal MUST be a pickup. All subsequent goals can be either pickups or dropoffs. The priority is an integer value that represents the priority of the job segment. The higher the number, the sooner the Fleet Manager is going to service the item. The default priority is 10, which can be changed in MobilePlanner. Only the priority of the first segment in the queueMulti command will have an impact on how soon the job is assigned to an AMR.
goalN	Enter the name of the Nth goal.
goalN args	Enter the arguments associated with the Nth goal.
jobId	Enter an optional identifier for the specified job. You can use a combination of string characters and integers. The jobId is helpful in tracking the job. If nothing is entered, ARCL generates a random jobId.

Responses

The command returns:

```
QueueMulti: goal "x" with priority 10 id PICKUP1 and jobid JOB1 successfully
queued
QueueMulti: goal <"goal1"> with priority <goal1_priority> id <PICKUPid_or_
DROPOFFid> jobid <jobId> successfully queued
QueueMulti: goal <"goal2"> with priority <goal2_priority> id <PICKUPid_or_
DROPOFFid> jobid <jobId> successfully queued and linked to <goal1_PICKUPid_or_
DROPOFFid>
:
:
QueueMulti: goal <"goaln"> with priority <goaln_priority> id <PICKUPid_or_
DROPOFFid> jobid <jobId> successfully queued and linked to <goal(n-1)_PICKUPid_
or_DROPOFFid>
EndQueueMulti
```

The reported jobId was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the jobId.

For details on the status conditions, see Status Conditions on page 23.

Details

The queueMulti command tells the AMR to go to multiple goals, to make pickups and dropoffs.

Examples

The following example shows a queuedropoff at goal 1.

Example #1 - Using Default job id

```
queuemulti 4 2 x pickup 10 y pickup 19 z dropoff 20 t dropoff 20
```

```
QueueMulti: goal "x" with priority 10 id PICKUP1 and jobid JOB1 successfully
queued
QueueMulti: goal "y" with priority 19 id PICKUP2 and jobid JOB1 successfully
queued and linked to PICKUP1
QueueMulti: goal "z" with priority 20 id DROPOFF3 and jobid JOB1 successfully
queued and linked to PICKUP2
QueueMulti: goal "t" with priority 20 id DROPOFF4 and jobid JOB1 successfully
queued and linked to DROPOFF3
EndQueueMulti

QueueUpdate: PICKUP1 JOB1 10 Pending None Goal "x" "None" 08/15/2013 06:02:59
None None 0
QueueUpdate: PICKUP2 JOB1 19 Pending ID_PICKUP1 Goal "y" "None" 08/15/2013
06:02:59 None None 0
QueueUpdate: DROPOFF3 JOB1 20 Pending ID_PICKUP2 Goal "z" "None" 08/15/2013
06:02:59 None None 0
QueueUpdate: DROPOFF4 JOB1 20 Pending ID_DROPOFF3 Goal "t" "None" 08/15/2013
06:02:59 None None 0
```



```

QueueUpdate: PICKUP1 JOB1 10 InProgress UnAllocated Goal "x" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: PICKUP1 JOB1 10 InProgress Allocated Goal "x" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: PICKUP1 JOB1 10 InProgress Driving Goal "x" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: PICKUP1 JOB1 10 Completed None Goal "x" "Bullwinkle (.53)"
08/15/2013 06:02:59 08/15/2013 06:03:20 0
QueueUpdate: PICKUP2 JOB1 19 InProgress UnAllocated Goal "y" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: PICKUP2 JOB1 19 InProgress Allocated Goal "y" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: PICKUP2 JOB1 19 InProgress Driving Goal "y" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: PICKUP2 JOB1 19 Completed None Goal "y" "Bullwinkle (.53)"
08/15/2013 06:02:59 08/15/2013 06:03:33 0
QueueUpdate: DROPOFF3 JOB1 20 InProgress UnAllocated Goal "z" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF3 JOB1 20 InProgress Allocated Goal "z" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF3 JOB1 20 InProgress Before Goal "z" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF3 JOB1 20 InProgress Driving Goal "z" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF3 JOB1 20 InProgress After Goal "z" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF3 JOB1 20 Completed None Goal "z" "Bullwinkle (.53)"
08/15/2013 06:02:59 08/15/2013 06:03:47 0
QueueUpdate: DROPOFF4 JOB1 20 InProgress UnAllocated Goal "t" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF4 JOB1 20 InProgress Allocated Goal "t" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF4 JOB1 20 InProgress Driving Goal "t" "Bullwinkle (.53)"
08/15/2013 06:02:59 None None 0
QueueUpdate: DROPOFF4 JOB1 20 Completed None Goal "t" "Bullwinkle (.53)"
08/15/2013 06:02:59 08/15/2013 06:04:03 0

```

Related Commands

- queueCancel Command (shortcut: qc) on page 39
- queuePickup Command (shortcut: qp) on page 50
- queuePickupDropoff Command (shortcut: qpd) on page 53
- queueQuery Command (shortcut: qq) on page 58
- queueShow Command (shortcut: qs) on page 61
- queueShowRobot Command (shortcut: qsr) on page 65

5.8 queuePickup Command (shortcut: qp)

Calls any available AMR for a pick up request.

Syntax

```
queuePickup <goalName> [priority or "default"] [jobId]
```

ARAM Settings

In order to use this feature, you have to explicitly enable it in the MobilePlanner software, by setting the EnterpriseQueuing argument in the Enterprise Features section of the **Configuration > Enterprise** tab.

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
goalName	Enter the name of the goal where you want the AMR to go for the pickup.
priority	An optional integer value that represents the priority of the pickup request. The higher the number, the sooner Fleet Manager is going to service the item. The default priority is 10, which can be changed in MobilePlanner.
jobId	An optional identifier for the specified job. You can use a combination of string characters and integers. The jobId is helpful in tracking the job. If nothing is entered, ARCL generates a random jobId.

Responses

The command returns the following information:

```
queuepickup goal "goalName" with priority [priority] id (id) and jobId [jobid]
successfully queued
```

Assuming the command was successful, the status of the robot is displayed:

```
QueueUpdate: <id> <jobId> <priority> <status = Pending> <substatus = None> Goal
<"goalName"> <assigned robotName = None> <queuedDate> <queuedTime> <com-
pletedDate = None> <completedTime = None> <failedCount>
QueueUpdate: <id> <jobId> <priority> <status = InProgress> <substatus = None>
Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <completedDate =
None> <completedTime = None> <failedCount>
QueueUpdate: <id> <jobId> <priority> <status = Completed> <substatus = None>
Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <completedDate> <com-
pletedTime> <failedCount>
```

The reported `jobId` was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the `jobId`.

For details on the status conditions, see [Status Conditions](#) on page 23.

Details

The `queuePickup` command calls any available AMR for a pick up request. When the job is at the top of the queue, the Fleet Manager sends the AMR to the specified goal.

If multiple AMRs are available for the pickup request, the Fleet Manager determines which AMR answers the request based on such factors as which AMR is closest to the goal, how long it has been idle, and its charge state. You can also enter a priority value: the higher the number, the higher the priority.

Examples

The following example shows a `queuePickup` at goal `z` with priority 11 and `job_id xyz`.

```

queuepickup z 11 xyz
queuepickup goal "z" with priority 11, id PICKUP13 and job_id xyz successfully
queued
QueueUpdate: PICKUP13 xyz 11 Pending None Goal "z" none 12/19/2011 06:54:18 None
None 0
QueueUpdate: PICKUP13 xyz 11 InProgress UnAllocated Goal "z" "Omron_Tele-
presence_Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress Allocated Goal "z" "Omron_Telepresence_
Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress BeforePickup Goal "z" "Omron_Tele-
presence_Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress BeforeEvery Goal "z" "Omron_Tele-
presence_Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress Before Goal "z" "Omron_Telepresence_
Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress Driving Goal "z" "Omron_Telepresence_
Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress After Goal "z" "Omron_Telepresence_
Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress AfterEvery Goal "z" "Omron_Telepresence_
Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress AfterPickup Goal "z" "Omron_Tele-
presence_Robot" 12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 Completed None Goal "z" "Omron_Telepresence_Robot"
12/19/2011 06:54:18 12/19/2011 06:54:34 0

```

Related Commands

[queryFaults Command](#) (shortcut: `qf`) on page 36

[queueCancel Command](#) (shortcut: `qc`) on page 39

[queueMulti Command](#) (shortcut: `qm`) on page 47

[queuePickupDropoff Command](#) (shortcut: `qpd`) on page 53

[queueQuery Command](#) (shortcut: `qq`) on page 58

queueShow Command (shortcut: qs) on page 61

queueShowCompleted Command (shortcut: qsc) on page 63

queueShowRobot Command (shortcut: qsr) on page 65

5.9 queuePickupDropoff Command (shortcut: qpd)

Queues a pick-up and drop-off request for any available AMR.

Syntax

```
queuePickupDropoff <goal1Name> <goal2Name> [priority1 or "default"] [priority2 or "default"] [jobId]
```

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
goal1Name	Enter the name of the goal where you want the AMR to go for the pickup.
goal2Name	Enter the name of the goal where you want the AMR to go for the dropoff.
priority1	An optional integer value that represents the priority of the pickup request. The higher the number, the sooner Fleet Manager is going to service the item. The default priority is 10, which can be changed in MobilePlanner.
priority2	An optional integer value that represents the priority of the dropoff request. The higher the number, the sooner Fleet Manager is going to service the item. The default priority is 20, which can be changed in MobilePlanner.
jobId	An optional identifier for the specified job. You can use a combination of string characters and integers. The jobId is helpful in tracking the job. If nothing is entered, ARCL generates a random jobId.

Responses

The command returns the following information:

```
queuepickupdropoff goals <"goal1"> and <"goal2"> with priorities <priority1> and <priority2> ids <PICKUPid> and <DROPOFFid> jobId <jobId> successfully queued and linked to jobId <jobid>
```

The PICKUPid and DROPOFFid are assigned by the system.

Assuming the command was successful, the status is displayed as follows:

```
QueueUpdate: <id> <jobId> <priority> <status=Pending> <substatus=None> Goal <"goal1"> <robotName> <queued date> <queued time> <completed date=None> <completed time=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=Pending> <substatus=ID_<id>> Goal <"goal2"> <robotName> <queued date> <queued time> <completed date=None> <completed time=None> <failed count>
```

```
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=UnAllocated>
Goal <"goal1"> <robotName> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Allocated>
Goal <"goal1"> <robotName> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Driving>
Goal <"goal1"> <robotName> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=Completed> <substatus=None> Goal
<"goal1"> <robotName> <queued date> <queued time> <completed date> <completed
time> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=UnAllocated>
Goal <"goal2"> <robotName> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Allocated>
Goal <"goal2"> <robotName> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Driving>
Goal <"goal2"> <robotName> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=Completed> <substatus=None> Goal
<"goal2"> <robotName> <queued date> <queued time> <completed date> <completed
time> <failed count>
```

The reported `jobId` was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the `jobId`.

For details on the status conditions, see [Status Conditions](#) on page 23.

Details

The `queuePickupDropoff` command calls any available AMR for a pick-up request and then tells it to go to a specific goal for a dropoff. You must specify the goal names. You can optionally specify the priorities for each goal, and specify a job identifier for each job. However, note that there is no AMR specification parameter in this command—it automatically chooses the most appropriate AMR in the fleet, as determined by the selection criteria and task requirements.

Examples

The following example shows the `queuepickupdropoff` command with `priority1` and `priority2` values and a job identifier.

```
queuepickupdropoff <PICKUPgoal_name> <DROPOFFgoal_name> [PICKUPpriority]
[DROPOFFpriority] [job_id]
```

Returns:

```
queuepickupdropoff goals <"PICKUPgoal"> and <"DROPOFFgoal"> with priorities
<PICKUPpriority> and <DROPOFFpriority> ids <PICKUPid> and <DROPOFFid> job_id
<jobid> successfully queued
```

```
QueueUpdate: <id> <job_id> <priority> <status=Pending> <substatus=None> Goal
```

```

<"goal_name"> <robot_name> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>

QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=None> Goal
<"goal_name"> <robot_name> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>

QueueUpdate: <id> <job_id> <priority> <status=Completed> <substatus=None> Goal
<"goal_name"> <robot_name> <queued date> <queued time> <completed date> <com-
pleted time> <failed count>

QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=None> Goal
<"goal_name"> <robot_name> <queued date> <queued time> <completed date=None>
<completed time=None> <failed count>

QueueUpdate: <id> <job_id> <priority> <status=Completed> <substatus=None> Goal
<"goal_name"> <robot_name> <queued date> <queued time> <completed date> <com-
pleted time> <failed count>

```

The following example shows the `queuepickupdropoff` command being used to swap the payload on the AMR:

```

queuepickupdropoff x y
queuepickupdropoff goals "x" and "y" with priorities 10 and 20 ids PICKUP12 and
DROPOFF13 job_id JOB12 successfully queued
QueueUpdate: PICKUP12 JOB12 10 Pending None Goal "x" "None" 08/16/2012 14:32:54
None None 0
QueueUpdate: DROPOFF13 JOB12 20 Pending None Goal "y" "None" 08/16/2012 14:32:54
None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress UnAllocated Goal "x" "Lynx1"
08/16/2012 14:32:54 None None 0
queuepickupdropoff y t
queuepickupdropoff goals "y" and "t" with priorities 10 and 20 ids PICKUP14 and
DROPOFF15 job_id JOB14 successfully queued and linked to job_id JOB12
QueueUpdate: PICKUP14 JOB14 10 Pending None Goal "y" "Lynx1" 08/16/2012 14:33:01
None None 0
QueueUpdate: DROPOFF15 JOB14 20 Pending None Goal "t" "Lynx1" 08/16/2012
14:33:01 None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress Allocated Goal "x" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress Driving Goal "x" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: PICKUP12 JOB12 10 Completed None Goal "x" "Lynx1" 08/16/2012
14:32:54 08/16/2012 14:33:15 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress UnAllocated Goal "y" "Lynx1"
08/16/2012 14:32:54 None None 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress Allocated Goal "y" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress Driving Goal "y" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: DROPOFF13 JOB12 20 Completed None Goal "y" "Lynx1" 08/16/2012
14:32:54 08/16/2012 14:33:27 0
QueueUpdate: PICKUP14 JOB14 10 Completed None Goal "y" "Lynx1" 08/16/2012
14:33:01 08/16/2012 14:33:27 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress UnAllocated Goal "t" "Lynx1"
08/16/2012 14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress Allocated Goal "t" "Lynx1" 08/16/2012

```

5.9 queuePickupDropoff Command (shortcut: qpd)

```
14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress Driving Goal "t" "Lynx1" 08/16/2012
14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 Completed None Goal "t" "Lynx1" 08/16/2012
14:33:01 08/16/2012 14:33:35 0

queuepickupdropoff x y
queuepickupdropoff goals "x" and "y" with priorities 10 and 20 ids PICKUP12 and
DROPOFF13 job_id JOB12 successfully queued
QueueUpdate: PICKUP12 JOB12 10 Pending None Goal "x" "None" 08/16/2012 14:32:54
None None 0
QueueUpdate: DROPOFF13 JOB12 20 Pending ID_PICKUP12 Goal "y" "None" 08/16/2012
14:32:54 None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress UnAllocated Goal "x" "Lynx1"
08/16/2012 14:32:54 None None 0
queuepickupdropoff y t
queuepickupdropoff goals "y" and "t" with priorities 10 and 20 ids PICKUP14 and
DROPOFF15 job_id JOB14 successfully queued and linked to job_id JOB12
QueueUpdate: PICKUP14 JOB14 10 Pending ID_DROPOFF13 Goal "y" "Lynx1" 08/16/2012
14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 Pending ID_PICKUP14 Goal "t" "Lynx1" 08/16/2012
14:33:01 None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress Allocated Goal "x" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress Driving Goal "x" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: PICKUP12 JOB12 10 Completed None Goal "x" "Lynx1" 08/16/2012
14:32:54 08/16/2012 14:33:15 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress UnAllocated Goal "y" "Lynx1"
08/16/2012 14:32:54 None None 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress Allocated Goal "y" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress Driving Goal "y" "Lynx1" 08/16/2012
14:32:54 None None 0
QueueUpdate: DROPOFF13 JOB12 20 Completed None Goal "y" "Lynx1" 08/16/2012
14:32:54 08/16/2012 14:33:27 0
QueueUpdate: PICKUP14 JOB14 10 Completed None Goal "y" "Lynx1" 08/16/2012
14:33:01 08/16/2012 14:33:27 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress UnAllocated Goal "t" "Lynx1"
08/16/2012 14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress Allocated Goal "t" "Lynx1" 08/16/2012
14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress Driving Goal "t" "Lynx1" 08/16/2012
14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 Completed None Goal "t" "Lynx1" 08/16/2012
14:33:01 08/16/2012 14:33:35 0
```

Related Commands

queryFaults Command (shortcut: qf) on page 36

queueCancel Command (shortcut: qc) on page 39

queueMulti Command (shortcut: qm) on page 47

queuePickup Command (shortcut: qp) on page 50

queueQuery Command (shortcut: qq) on page 58

queueShow Command (shortcut: qs) on page 61

queueShowCompleted Command (shortcut: qsc) on page 63

queueShowRobot Command (shortcut: qsr) on page 65

5.10 queueQuery Command (shortcut: qq)

Shows the job status of the queue by type or value.

Items will be displayed by priority. If, for example, dropoff priority is 20 and pickup priority is 10, then dropoff items will be displayed first, followed by pickup items.

Syntax

```
queueQuery <type> <value> [echoString]
```

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
type	Enter the type of job. Valid types are: <ul style="list-style-type: none"> id = the pickup or dropoff identification jobId = the job identification robotName = the AMR name status = the item status.
value	Enter the value that corresponds with the type used: For id, enter the pickup or dropoff identification, for example: PICKUP2 For jobId, enter the job identification, for example: JOB2 For robotname, enter the AMR name, for example: robot_34 For status, enter one of the following values: <ul style="list-style-type: none"> inprogress = queries a job with an InProgress status. pending = queries a job with a Pending status. interrupted = queries a job with an Interrupted status. completed = queries a job with a Completed status. cancelled = queries a job with a Cancelled status. failed = queries a job with a Failed status.
echoString	An optional string that is appended to each line of the results.

Responses

The command returns the following for a pending item:

```
QueueQuery: <id> <jobId> <priority> <status> <substatus> Goal <"goalName">
<robotName> <queued date> <queued time> <completed date> <completed time>
<echostring> <failed count>
```

EndQueueQuery

The returned items will be displayed by priority, as shown in the Examples. If, for example, dropoff priority is 20 and pickup priority is 10, the dropoff items will be displayed before the pickup items.

Details

The queueQuery command is used to view the status of the job queue. The queue can be queried by type (such as the AMR name or job identification) or by the job status.

The reported jobId was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the jobId.

An optional string can be specified, which will be appended to each line of the results.

For details on the status conditions, see Status Conditions on page 23.

Examples

The following example shows the status of the completed jobs in the queue.

```
queuequery status completed xyz

QueueQuery: DROPOFF18 y4rt 22 Completed None Goal "x" "MT-490" 12/19/2011
07:07:53 12/19/2011 07:08:07 xyz 0
QueueQuery: DROPOFF16 abc 20 Completed None Goal "x" "MT-490" 12/19/2011
07:06:00 12/19/2011 07:06:16 xyz 0
QueueQuery: DROPOFF17 JOB17 20 Completed None Goal "z" "MT-490" 12/19/2011
07:06:21 12/19/2011 07:06:35 xyz 0
QueueQuery: DROPOFF19 yyy 20 Completed None Goal "x" "MT-490" 12/19/2011
07:08:49 12/19/2011 07:08:49 xyz 0
QueueQuery: DROPOFF20 yyy 20 Completed None Goal "x" "MT-490" 12/19/2011
07:09:08 12/19/2011 07:09:09 xyz 1
QueueQuery: DROPOFF21 JOB21 20 Completed None Goal "x" "MT-490" 12/19/2011
07:09:33 12/19/2011 07:09:34 xyz 0
QueueQuery: PICKUP12 xyz 11 Completed None Goal "t" "MT-490" 12/19/2011
06:53:51 12/19/2011 06:54:02 xyz 5
QueueQuery: PICKUP13 xyz 11 Completed None Goal "z" "OAT_Telepresence_Robot"
12/19/2011 06:54:18 12/19/2011 06:54:34 xyz 0
EndQueueQuery
```

Related Commands

queryFaults Command (shortcut: qf) on page 36

queueCancel Command (shortcut: qc) on page 39

queueMulti Command (shortcut: qm) on page 47

queuePickup Command (shortcut: qp) on page 50

queuePickupDropoff Command (shortcut: qpd) on page 53

queueShow Command (shortcut: qs) on page 61

queueShowCompleted Command (shortcut: qsc) on page 63

queueShowRobot Command (shortcut: qsr) on page 65

5.11 queueShow Command (shortcut: qs)

Shows the status of the last 11 jobs in the queue, including any jobs assigned to the AMRs and the status of each job. Oldest jobs are displayed first.

Syntax

```
queueShow [echoString]
```

Usage Considerations

Shows all jobs and all AMRs. To look at a specific job, use `queueQuery`. To look at a specific AMR, use `queueShowRobot`.

Parameters

The command parameters are described in the following table.

Parameter	Definition
echoString	An optional string that is appended to each line of the results.

Responses

The command returns the following information:

```
QueueRobot: <robotName> <robotStatus> <robotSubstatus> <echoString>
QueueShow: <id> <jobId> <priority> <status> <substatus> Goal <"goalName"> <"ro-
botName"> <queued date> <queued time> <completed date> <completed time>
<echoString> <failed count>
EndQueueShow
```

Details

The `queueShow` command provides a listing of all AMRs connected to the Fleet Manager, and all jobs in the queue including those that are pending, interrupted, or are currently assigned to the AMRs. You do not specify a AMR with this command. Instead, it lists the information for all AMRs. If you wish to look at a specific AMR, use the `queueShowRobot` command. For details, see the `queueShowRobot` Command (shortcut: `qsr`) on page 65. If you wish to look at a specific job, use the `queueQuery` command. For details, see the `queueQuery` Command (shortcut: `qq`) on page 58.

The reported `jobId` was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the `jobId`.

For details on the status conditions, see `Status Conditions` on page 23.

An optional string can be specified, which will be appended to each line of the results.

Examples

```
queueshow
QueueRobot: "21" InProgress Driving ""
QueueRobot: "22" Available Available ""
QueueRobot: "23" Available Available ""
QueueRobot: "24" Available Available ""
QueueRobot: "25" Available Available ""
QueueRobot: "26" Available Available ""

QueueShow: PICKUP3 JOB3 10 Completed None Goal "1" "21" 11/14/2012 11:49:23
11/14/2012 11:49:23 "" 0
QueueShow: PICKUP4 JOB4 10 InProgress Driving Goal "7" "21" 11/14/2012 11:49:34
None None "" 0
EndQueueShow
```

Related Commands

[queryFaults Command \(shortcut: qf\) on page 36](#)

[queueCancel Command \(shortcut: qc\) on page 39](#)

[queueMulti Command \(shortcut: qm\) on page 47](#)

[queuePickup Command \(shortcut: qp\) on page 50](#)

[queuePickupDropoff Command \(shortcut: qpd\) on page 53](#)

[queueQuery Command \(shortcut: qq\) on page 58](#)

[queueShowCompleted Command \(shortcut: qsc\) on page 63](#)

[queueShowRobot Command \(shortcut: qsr\) on page 65](#)

5.12 queueShowCompleted Command (shortcut: qsc)

Shows the jobs in the queue with a status of Completed, oldest first.

Syntax

```
queueShowCompleted [echoString]
```

Usage Considerations

Shows only jobs with a status of Completed. To look at a specific job, use queueQuery. To look at a specific AMR, use queueShowRobot.

The MobilePlanner configuration parameter `maxNumberOfCompletedItems`, which has a default of 100, limits the number of completed jobs that will be kept in the queue.

The MobilePlanner configuration parameter `DeleteCompletedItemsMinutes`, which has a default of 60, determines how long completed jobs will be kept in the queue. Jobs older than this will be deleted from the queue, and cannot be viewed.

Either of these two parameters can limit the number of jobs in the queue that are available for viewing with the queueShowCompleted command.

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
echoString	An optional string that is appended to each line of the results.

Returns

The command returns the following information:

```
QueueShow: <id> <jobId> <priority> <status> <substatus> Goal <"goalName"> <"ro-
botName"> <queued date> <queued time> <completed date> <completed time>
<echoString> <failed count>
EndQueueShowCompleted
```

Details

The queueShowCompleted command provides a listing of the jobs in the queue that are Completed, oldest first. You do not specify an AMR with this command. Instead, it lists the information for all AMRs. If you wish to look at a specific AMR, use the queueShowRobot command. For details, see the queueShowRobot Command (shortcut: qsr) on page 65. If you wish to look at a specific job, use the queueQuery command. For details, see the queueQuery Command (shortcut: qq) on page 58.

The reported jobId was either provided as part of the request, or was autogenerated by the Fleet Manager software.

All failed counts are based on the jobId.

For details on the status conditions, see Status Conditions on page 23.

An optional string can be specified, which will be appended to each line of the results.

Examples

```
queueshowcompleted
```

```
QueueShow: PICKUP19 JOB19 10 Completed None Goal "t" "Bullwinkle (.53)"
05/06/2013 05:55:33 05/06/2013 05:56:02 "" 0
QueueShow: PICKUP21 JOB21 10 Completed None Goal "t" "guiabot_2010_09_20"
05/06/2013 06:00:21 05/06/2013 06:00:42 "" 0
QueueShow: PICKUP22 JOB22 10 Completed None Goal "t" "Bullwinkle (.53)"
05/06/2013 06:00:32 05/06/2013 06:01:05 "" 0
QueueShow: PICKUP23 JOB23 10 Completed None Goal "t" "guiabot_2010_09_20"
05/06/2013 06:01:03 05/06/2013 06:01:23 "" 0
EndQueueShowCompleted
```

Related Commands

[queryFaults Command \(shortcut: qf\) on page 36](#)

[queueCancel Command \(shortcut: qc\) on page 39](#)

[queuePickup Command \(shortcut: qp\) on page 50](#)

[queuePickupDropoff Command \(shortcut: qpd\) on page 53](#)

[queueQuery Command \(shortcut: qq\) on page 58](#)

[queueShow Command \(shortcut: qs\) on page 61](#)

[queueShowRobot Command \(shortcut: qsr\) on page 65](#)

5.13 queueShowRobot Command (shortcut: qsr)

Shows the status and substatus of all AMRs (or, optionally, a specific AMR) connected to the Fleet Manager.

Syntax

```
queueShowRobot [robotName or "default"] [echoString]
```

Usage Considerations

This command does not return any job information; to view the queue and job information, use the queueShow command from ARCL on the Fleet Manager.

Parameters

The command parameters are described in the following table.

For details on the data types, see Understanding the Commands on page 20.

Parameter	Definition
robotName	Enter the name of the AMR. To view all the AMRs connected to the Fleet Manager, omit this parameter or enter "default".
echoString	An optional string that is appended to each line of the results. Requires a value in the previous parameter.

Responses

The command returns the following:

```
QueueRobot: "robotName" robotStatus robotSubstatus echoString
EndQueueShowRobot
```

For details on the status conditions, see Status Conditions on page 23.

Details

The queueShowRobot command displays the status of the AMRs currently connected to the Fleet Manager. Optionally, this command allows you to query a specific AMR name.

This command does not return the job status for jobs currently in progress. To view that information, use the queueShow command. For details, see queueShow Command (shortcut: qs) on page 61.

An optional string can be specified, which will be appended to each line of the results.

Examples

The following example shows the status and substatus of AMR 31:

5.13 queueShowRobot Command (shortcut: qsr)

```
queueshowrobot 31
```

```
QueueRobot: "31" Available Available ""
```

The following example shows the status and substatus of all AMRs and includes an optional message "echoit":

```
Queueshowrobot default echoit
```

```
QueueRobot: "Robot1" UnAvailable EStopPressed echoit
```

```
QueueRobot: "Robot2" UnAvailable Interrupted echoit
```

```
QueueRobot: "Robot3" UnAvailable InterruptedButNotYetIdle echoit
```

```
QueueRobot: "Robot4" Available Available echoit
```

```
QueueRobot: "Robot5" InProgress Driving echoit
```

```
QueueRobot: "Robot6" UnAvailable NotUsingEnterpriseManager echoit
```

```
QueueRobot: "Robot7" UnAvailable UnknownBatteryType echoit
```

```
QueueRobot: "Robot8" UnAvailable ForcedDocked echoit
```

```
QueueRobot: "Robot9" UnAvailable NotLocalized echoit
```

```
QueueRobot: "patrolbot" UnAvailable Fault_Driving_Application_faultName echoit
```

```
EndQueueShowRobot
```

Related Commands

[queryFaults Command \(shortcut: qf\) on page 36](#)

[queueCancel Command \(shortcut: qc\) on page 39](#)

[queueMulti Command \(shortcut: qm\) on page 47](#)

[queuePickup Command \(shortcut: qp\) on page 50](#)

[queuePickupDropoff Command \(shortcut: qpd\) on page 53](#)

[queueQuery Command \(shortcut: qq\) on page 58](#)

[queueShow Command \(shortcut: qs\) on page 61](#)

[queueShowCompleted Command \(shortcut: qsc\) on page 63](#)

OMRON Corporation Industrial Automation Company
Kyoto, JAPAN

Contact: www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.

Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ASIA PACIFIC PTE. LTD.

No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON ELECTRONICS LLC

2895 Greenspoint Parkway, Suite 200 Hoffman Estates,
IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ADEPT TECHNOLOGIES, INC.

4550 Norris Canyon Road, Suite 150, San Ramon, CA 94583 U.S.A.
Tel: (1) 925-245-3400/Fax: (1) 925-960-0590

OMRON (CHINA) CO., LTD.

Room 2211, Bank of China Tower, 200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

© OMRON Corporation 2017-2019 All Rights Reserved.
In the interest of product improvement, specifications are
subject to change without notice.

Cat. No. I618-E-02

Printed in USA
0819 (0117)

18449-000 B