

Advanced Robotics Control Language Enterprise Manager - Mobile Robots

Integration Guide

ARCL Enterprise Manager Integration Guide - Mobile Robots

This is a PDF/print version of the ARCL Enterprise Manager Integration Guide. A Table of Contents is provided so that you can locate the desired topics. Because the ARCL Enterprise Manager Integration Guide was designed for online viewing, there may be slight formatting anomalies in the PDF/print version. Additionally, links to external documents will not work in the PDF file.

NOTE: Please see the ReadMe file, which is included with your Motivity software, for a description of any recent changes.

Copyright Notice

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

Omron Adept 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@adept.com.

Copyright © 2006-2016 by Omron Adept 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

Introduction to ARCL	7
Version Requirements	8
How Do I Begin	9
Related Manuals	10
How Can I Get Help?	11
Set ARCL Parameters in MobilePlanner	12
Accessing the Configuration Options	13
Understanding the Configuration Parameters	17
Outgoing Enterprise ARCL Connection Setup Parameters	18
Outgoing Enterprise ARCL Commands Parameters	19
See Also...	19
Connect to ARCL Using a Telnet Client	20
Setting the Connection Parameters	21
Connecting to Advanced Robotics Command Language	22
See Also...	23
Using the ARCL Commands	24
See Also...	24
Understanding the Commands	25
Document Conventions	25
Command Notes	26
Data Types	26
Status and Error Messages	28
Status Conditions	28
Using the Queuing Commands	32
Working With Payloads	33
ARCL Command Reference	34
See Also...	34
getDateTime Command	35
Syntax	35
Usage Considerations	35
Parameters	35
Examples	35
payloadQuery Command (shortcut: pq)	36

Syntax	36
Usage Considerations	36
Parameters	36
Responses	36
Details	36
Examples	37
Related Commands	38
payloadSlotCount Command (shortcut: psc)	39
Syntax	39
Usage Considerations	39
Parameters	39
Responses	39
Details	39
Examples	39
Related Commands	40
queryFaults Command (shortcut: qf)	41
Syntax	41
Usage Considerations	41
Parameter	41
Responses	41
Details	41
Example	41
Related Commands	43
queueCancel Command (shortcut: qc)	44
Syntax	44
Usage Considerations	44
Parameters	44
Responses	44
Details	45
Examples	45
Related Commands	46
queueModify Command (shortcut: qmod)	47
Syntax	47
Usage Considerations	47
ARAM Settings	47
Parameters	47
Responses	48
Details	49
Examples	49
Related Commands	51
queueMulti Command (shortcut: qm)	53
Syntax	53
Usage Considerations	53
ARAM Settings	53
Parameters	53
Responses	54
Details	55

Examples	55
Related Commands	56
queuePickup Command (shortcut: qp)	57
Syntax	57
Usage Considerations	57
ARAM Settings	57
Parameters	57
Responses	57
Details	58
Examples	58
Related Commands	59
queuePickupDropoff Command (shortcut: qpd)	60
Syntax	60
Usage Considerations	60
Parameters	60
Responses	60
Details	61
Examples	61
Related Commands	64
queueQuery Command (shortcut: qq)	65
Syntax	65
Usage Considerations	65
Parameters	65
Responses	65
Details	66
Examples	66
Related Commands	66
queueShow Command (shortcut: qs)	68
Syntax	68
Usage Considerations	68
Parameters	68
Responses	68
Details	68
Examples	69
Related Commands	69
queueShowCompleted Command (shortcut: qsc)	70
Syntax	70
Usage Considerations	70
Parameters	70
Returns	70
Details	70
Examples	71
Related Commands	71
queueShowRobot Command (shortcut: qsr)	72
Syntax	72
Usage Considerations	72
Parameters	72

Responses	72
Details	72
Examples	73
Related Commands	73

Introduction to ARCL

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

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

The Enterprise Manager (EM) version of ARCL is for use with the Enterprise Manager software and appliance. This hardware and software combination has been specially designed and configured to manage a fleet of robots 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 Enterprise Manager software.

This section discusses the following topics:

Version Requirements	8
How Do I Begin	9
Related Manuals	10
How Can I Get Help?	11

For more information on using the Mobile Robots Software Suite, refer to the *Omron Adept Technologies, Inc.*

See Also...

- Introduction to ARCL on page 7
- Set ARCL Parameters in MobilePlanner on page 12
- Connect to ARCL Using a Telnet Client on page 20
- Using the ARCL Commands on page 24
- ARCL Command Reference on page 34

Version Requirements

This document pertains to ARAM version 4.6 and later.

If you need assistance, see How Can I Get Help? on page 11.

See Also...

How Do I Begin on page 9

Related Manuals on page 10

How Can I Get Help? on page 11

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 12.

2. Connect to ARCL Using a Telnet Client.

Using a Telnet client, connect to ARCL to access and run the ARCL commands on the Motivity platform. For details, see Connect to ARCL Using a Telnet Client on page 20.

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 24.

See Also...

Version Requirements on page 8

How Do I Begin on page 9

Related Manuals on page 10

How Can I Get Help? on page 11

Related Manuals

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

Manual	Description
<i>Mobile Robot 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.

See Also...

Version Requirements on page 8

How Do I Begin on page 9

Related Manuals on page 10

How Can I Get Help? on page 11

How Can I Get Help?

For details on getting assistance with your Omron Adept Technologies, Inc. software or hardware, you can access the following Omron corporate and Omron Adept Technologies, Inc. websites:

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

See Also...

Version Requirements on page 8

How Do I Begin on page 9

Related Manuals on page 10

How Can I Get Help? on page 11

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 13
- Understanding the Configuration Parameters on page 17
- Set ARCL Parameters in MobilePlanner on page 12
- Set ARCL Parameters in MobilePlanner on page 12

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.

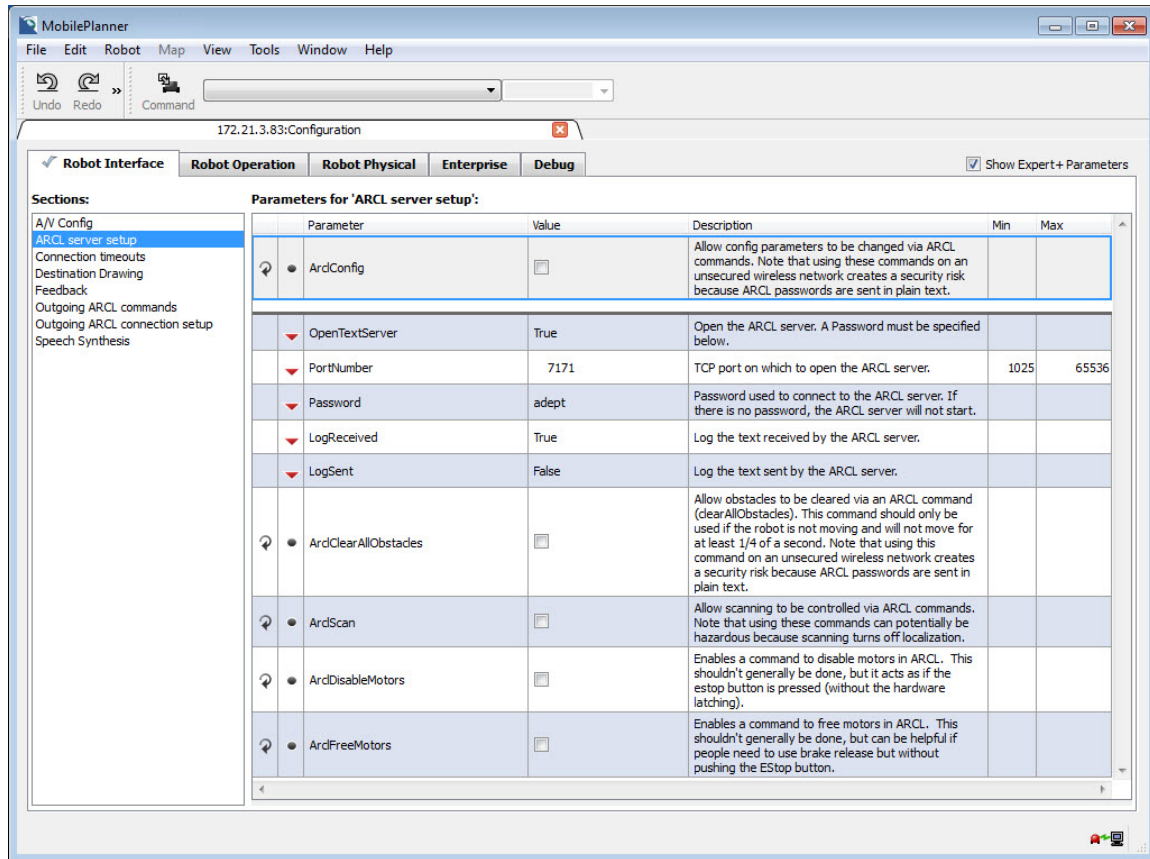
To access ARCL configuration options from MobilePlanner:

1. Open the MobilePlanner software, version 4.0 or later, and connect to the mobile robot. 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. These parameters allow you to control the client-server connection between an offboard client process (such as Telnet or PuTTY) and ARCL. The Advanced Robotics Command Language server setup parameters are shown in the following figure.

Incoming connections refer to a client initiating the connection to the Enterprise Manager. 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.

Accessing the Configuration Options

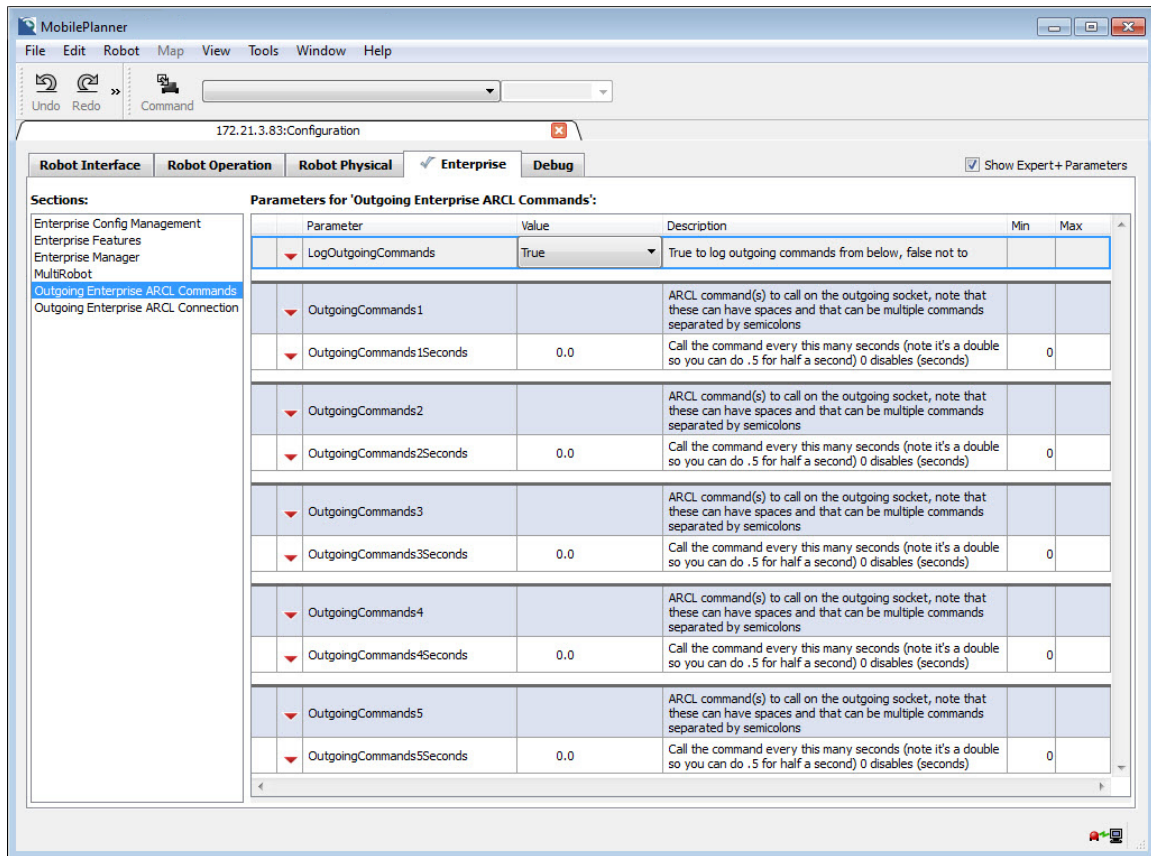


ARCL Server Setup Parameters

For more information on using a client (like Telnet or PuTTY), see [Connect to ARCL Using a Telnet Client](#) on page 20.

4. Select Outgoing Enterprise Advanced Robotics Command Language 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 19.

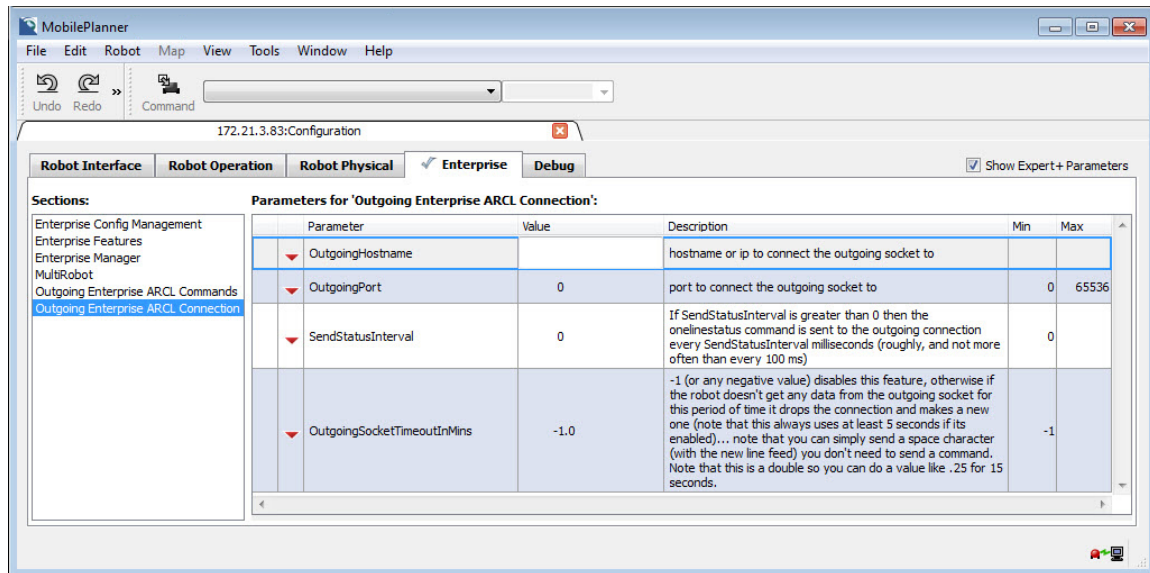
Accessing the Configuration Options



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 Enterprise Manager using Advanced Robotics Command Language commands, intended to connect to the facility WMS/MES. For more details, refer to Outgoing Enterprise ARCL Connection Setup Parameters on page 18.

Accessing the Configuration Options



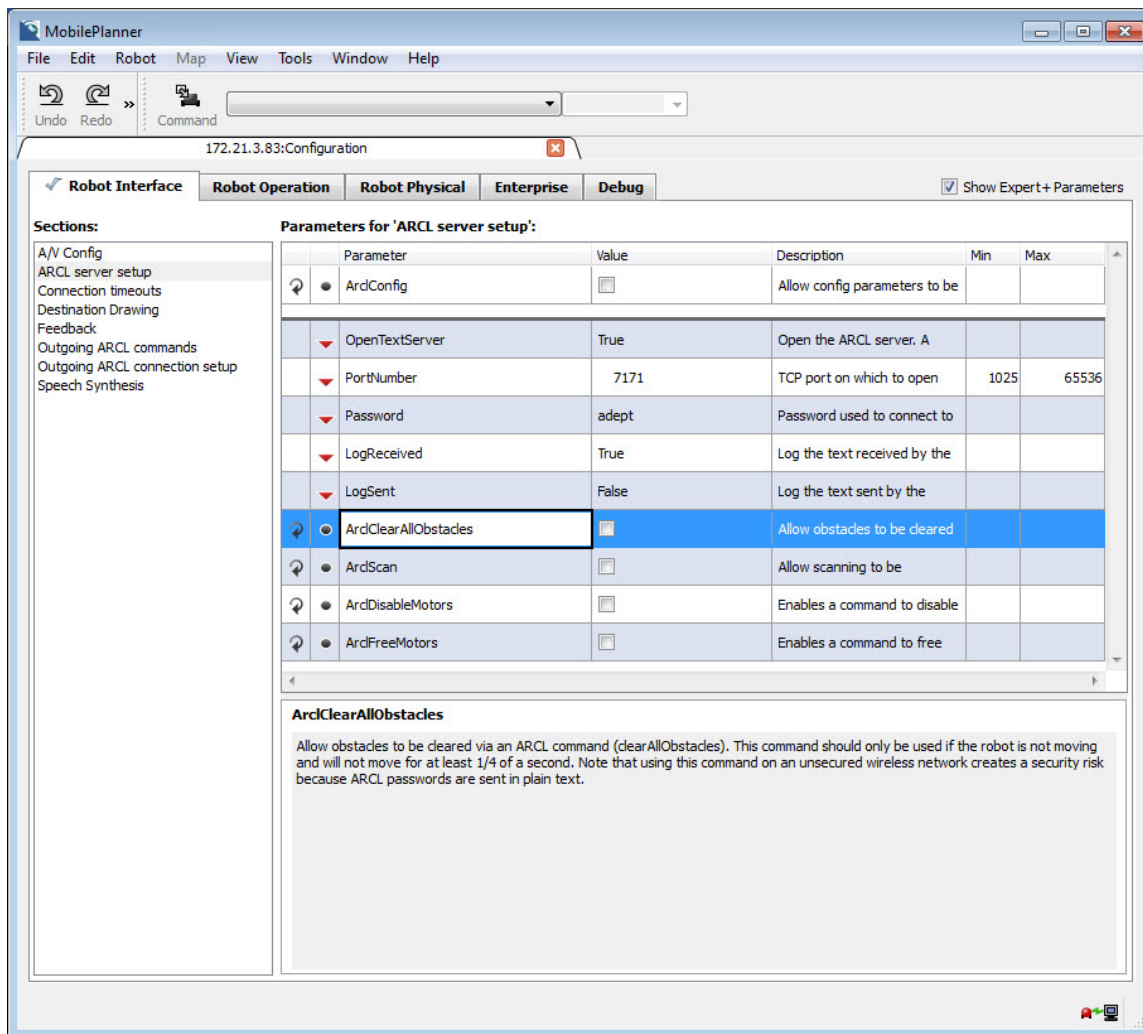
Outgoing Enterprise ARCL Connection Setup

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 robot is idle and stationary; the Configuration changes are saved.

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 the function of the parameter. The selected parameter's help description is located in the Description column and, optionally, 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.



Parameter Help

Outgoing Enterprise ARCL Connection Setup Parameters

The Outgoing Enterprise ARCL connection setup parameters are used to instruct the Enterprise Manager 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 Enterprise Manager.

There may be hand-shaking involved between the Enterprise 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:

- 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 Enterprise Manager 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 robots connected vs. how many robots should be connected, and generate an alarm if there is a mismatch.

See Also...

[Introduction to ARCL on page 7](#)

[Set ARCL Parameters in MobilePlanner on page 12](#)

[Connect to ARCL Using a Telnet Client on page 20](#)

[Using the ARCL Commands on page 24](#)

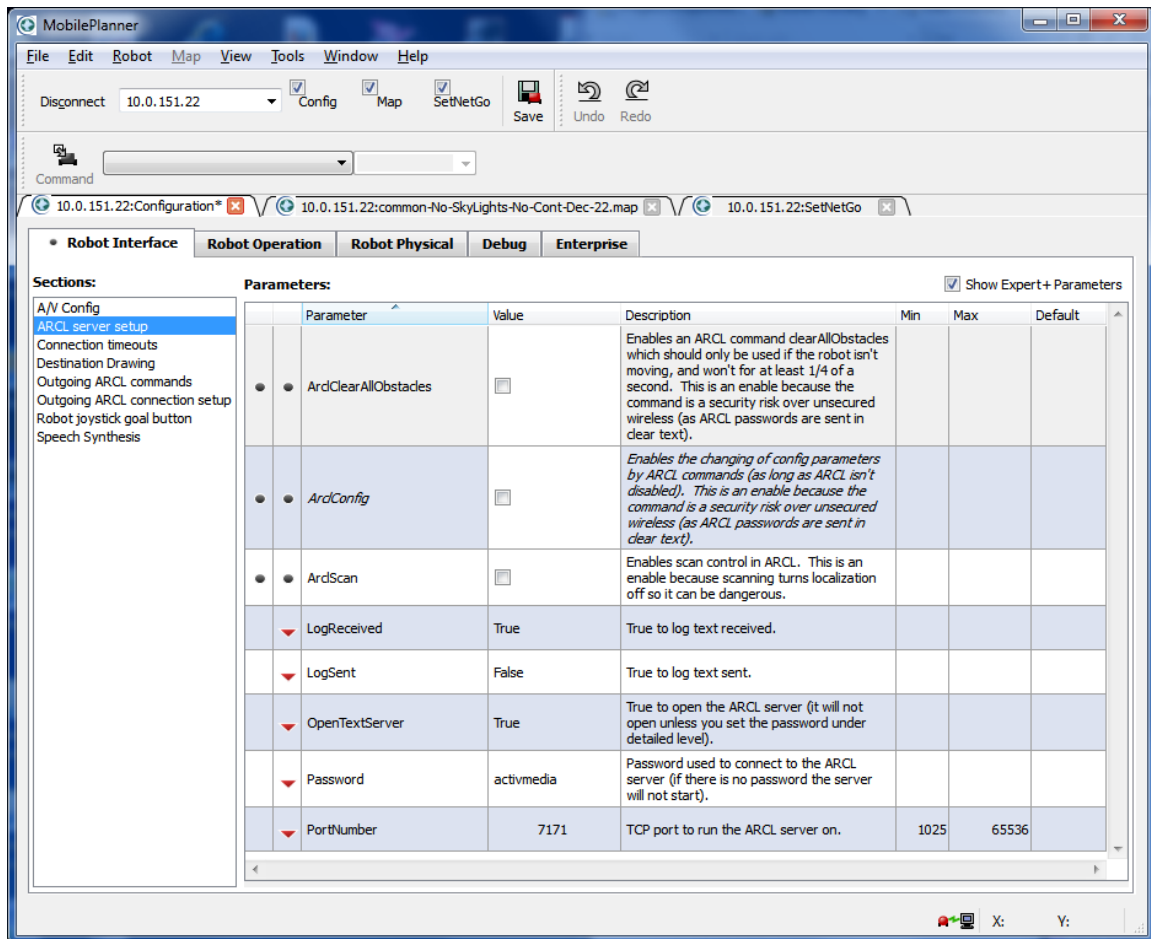
[ARCL Command Reference on page 34](#)

Connect to ARCL Using a Telnet Client

This section tells you how to connect to your mobile robot 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 mobile robot. 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.



ARCL Server Setup Parameters

These parameters allow you to control the client-server connection, see Understanding the Configuration Parameters on page 17 for details.

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

Connecting to Advanced Robotics Command Language

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.

In Windows, hold down the "Window" key and 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.

2. 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
```

3. Enter the password that you set in Step 5, above. If you mis-type the password, you will have to restart the Telnet client.

After you have successfully 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
```

Example Command List after Login

4. If needed, you can enter the **echo off** command to prevent your input from echoing (typing double characters).
5. When you are finished, use the **quit** command to properly 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 34.

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 **oneLineStatus** response actually receives the status message.

See Also...

Introduction to ARCL on page 7

Set ARCL Parameters in MobilePlanner on page 12

Connect to ARCL Using a Telnet Client on page 20

Using the ARCL Commands on page 24

ARCL Command Reference on page 34

Using the ARCL Commands

After you have established a connection to the ARCL server, you are ready to operate and monitor the mobile robot using the Advanced Robotics Command Language 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 34.

This section discusses the following topics:

See Also...	24
Understanding the Commands	25
Document Conventions	25
Command Notes	26
Data Types	26
Status and Error Messages	28
Status Conditions	28
Using the Queuing Commands	32
Working With Payloads	33

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.

See Also...

Introduction to ARCL on page 7
Set ARCL Parameters in MobilePlanner on page 12
Connect to ARCL Using a Telnet Client on page 20
Using the ARCL Commands on page 24
ARCL Command Reference on page 34

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 34.

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 only available on the robot.
- This ARCL command is available only on the Enterprise Manager.
- This ARCL command is available on the robot and Enterprise Manager.

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 goalname, routename, echo, etc.).

Responses

This section shows the information returned by the command.

Details

This section provides more details about the functions of the command.

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

Data Types

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 file-name, or because a feature was not set up properly, a `SetupError` is displayed. For example, if you attempt to execute `listAdd` or `listExecute` before entering the command `listStart`, the following error is displayed:

```
SetupError: You need to start a list before you can add to it.
```

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 robot's or a job's state.

Status Conditions

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

Status Conditions

Status	Substatus
Pending	None
Pending	AssignedRobotOffLine
Pending	NoMatchingRobotForLinkedJob
Pending	NoMatch- ingRobotForOtherSegment
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
InProgress	Before
InProgress	Buffering
InProgress	Buffered
InProgress	Driving
InProgress	After
InProgress	AfterEvery

Status	Substatus
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

See Also...

Understanding the Commands on page 25

Using the Queuing Commands on page 32

Working With Payloads on page 33

Using the Queuing Commands

The ARCL queuing commands are used with the Enterprise Manager. They allow you to request a mobile robot 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 44

queueModify Command (shortcut: qmod) on page 47

queueMulti Command (shortcut: qm) on page 53

queuePickup Command (shortcut: qp) on page 57

queuePickupDropoff Command (shortcut: qpd) on page 60

queueQuery Command (shortcut: qq) on page 65

queueShow Command (shortcut: qs) on page 68

queueShowCompleted Command (shortcut: qsc) on page 70

queueShowRobot Command (shortcut: qsr) on page 72

See Also...

Understanding the Commands on page 25

Using the Queuing Commands on page 32

Working With Payloads on page 33

Working With Payloads

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

The following commands are supported:

payloadQuery Command (shortcut: pq) on page 36

payloadSlotCount Command (shortcut: psc) on page 39

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

The default number of slots is 4. Note that slot numbering starts at 1. There is no slot 0; that would indicate there is no payload.

See Also...

Understanding the Commands on page 25

Using the Queuing Commands on page 32

Working With Payloads on page 33

ARCL Command Reference

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

getDateTime Command	35
payloadQuery Command (shortcut: pq)	36
payloadSlotCount Command (shortcut: psc)	39
queryFaults Command (shortcut: qf)	41
queueCancel Command (shortcut: qc)	44
queueModify Command (shortcut: qmod)	47
queueMulti Command (shortcut: qm)	53
queuePickup Command (shortcut: qp)	57
queuePickupDropoff Command (shortcut: qpd)	60
queueQuery Command (shortcut: qq)	65
queueShow Command (shortcut: qs)	68
queueShowCompleted Command (shortcut: qsc)	70
queueShowRobot Command (shortcut: qsr)	72

See Also...

Introduction to ARCL on page 7
Set ARCL Parameters in MobilePlanner on page 12
Connect to ARCL Using a Telnet Client on page 20
Using the ARCL Commands on page 24
ARCL Command Reference on page 34

getDateTime Command

Returns the system date and time.

Syntax

getDateTime

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

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
```

payloadQuery Command (shortcut: pq)

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

Syntax

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

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

Parameters

The command parameters are described in the following table.

For details on the data types, see Data Types on page 26.

Parameter	Definition
robotName	Enter the name of the robot 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 robot.

Details

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

- all slots on all robots
- a specified slot on a robot
- all slots on a specified robot

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, robot 21 is carrying books and glasses. To view what robot 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 robots connected to the Enterprise 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 "morestuff" 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 robots and echoes the string "hello":

```
payloadquery default default hello
PayloadQuery: "31" 1 "slotjunk" 05/07/2012 21:11:33 hello
PayloadQuery: "31" 2 "abc" 05/07/2012 21:10:53 hello
PayloadQuery: "31" 3 "def" 09/10/2012 12:14:14 hello
PayloadQuery: "31" 4 "ghi" 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
```

Related Commands

```
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 39

payloadSlotCount Command (shortcut: psc)

Displays the slot count on a specific robot or on all robots.

Syntax

payloadSlotCount [robotName or "default"] [echoString]

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

Parameters

The command parameters are described in the following table.

For details on the data types, see Data Types on page 26.

Parameter	Definition
robotName	Enter the name of the robot to display its slot count. To view the slot counts for all connected robots, 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.

Details

The payloadSlotCount command is used to display the slot count on a specific robot or on all robots. To limit the query to a specific robot, enter the robot name; to view the slot count on all robots, omit the robot name.

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

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

```
payloadslotcount 21
```


The command returns:

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

The following example displays the slot counts on all robots connected to the Enterprise 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 36

queryFaults Command (shortcut: qf)

Displays the faults associated with the specified robot.

Syntax

queryFaults [robotName or "default"] [echoString]

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

Displays all faults on the specified robot. Displays faults on all robots if the robotName parameter is omitted.

Parameter

The command parameters are described in the following table.

For details on the data types, see Data Types on page 26.

Parameter	Definition
robotName	Enter the name of the robot. To view all the robots connected to the Enterprise 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> <faultLongDescription>
<bool:drivingFault> <bool:criticalFault><bool:applicationFault><bool:clearedOnGo><bool:
clearedOnAcknowledgement> <echoString>
EndQueryFaults
```

Details

The queryFaults command provides a listing of all faults for the specified robot, or all faults for all robots connected to the Enterprise Manager if no robot is specified.

Example

```
queryfaults robot1
RobotFaultQuery: "robot1" Fault_Critical_Application fault1 "shortdesc" "longdesc" false
true true false false ""
EndQueryFaults

queryfaults robot1 echoit
RobotFaultQuery: "robot1" Fault_Critical_Application fault1 "shortdesc" "longdesc" false
true true false false echoit
EndQueryFaults
```

Example

```
queryfaults
RobotFaultQuery: "robot2" Fault_Driving_Application fault2 "shortd" "longd" true false
true false false ""
RobotFaultQuery: "robot1" Fault_Critical_Application fault1 "shortdesc" "longdesc" 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 provider 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 continue 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" "long-
desc" 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 robots 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 overheated
(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 critically 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 main-
tenance" 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 false
true false true
```

Related Commands

```
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 provider 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 continue 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 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 "shortdesc" "long-desc" false true true false false
RobotFaultCleared: "Sim2" Fault_Application_ClearedOnAcknowledgement f1 "s" "l" false false true false true
EndQueryFaults
```

Related Commands

queueCancel Command (shortcut: qc) on page 44

queuePickup Command (shortcut: qp) on page 57

queuePickupDropoff Command (shortcut: qpd) on page 60

queueModify Command (shortcut: qmod) on page 47

queueMulti Command (shortcut: qm) on page 53

queueQuery Command (shortcut: qq) on page 65

queueShowRobot Command (shortcut: qsr) on page 72

queueShowCompleted Command (shortcut: qsc) on page 70

queueCancel Command (shortcut: qc)

Cancels a queued request for a robot by type or value.

Syntax

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

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

Parameters

The command parameters are described in the following table.

For details on the data types, see Data Types on page 26.

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 robot 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 robot 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:

```
queuecancel cancelling <cancelType> <cancelValue> <echoString> <reason> from queue
```

```
QueueUpdate: <id> <jobId> <priority> <status = Cancelled> <subStatus = reason_or_None>
Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <completedDate> <com-
pletedTime> <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> <completedDate = None> <com-
pletedTime = None> <echoString>
QueueUpdate: <id> <jobId> <priority> <status = Interrupted> <subStatus = reason_or_None>
Goal <"goalName"> <"robotName"> <queuedDate> <queuedTime> <completedDate = None> <com-
pletedTime = None> <failedCount>
QueueUpdate: <id> <jobId> <priority> <status = Cancelled> <subStatus = reason_or_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 Enterprise Manager software.

All failed counts are based on the `jobId`.

For details on the status conditions, see [Status Conditions](#) on page 28.

Details

The `queueCancel` command is used to cancel a queued robot request. The request can be canceled by type (such as the robot 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 41

queueMulti Command (shortcut: qm) on page 53

queuePickup Command (shortcut: qp) on page 57

queuePickupDropoff Command (shortcut: qpd) on page 60

queueQuery Command (shortcut: qq) on page 65

queueShow Command (shortcut: qs) on page 68

queueShowCompleted Command (shortcut: qsc) on page 70

queueShowRobot Command (shortcut: qsr) on page 72

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>

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

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 Data Types on page 26.

Responses

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 identificationpriority = 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
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 Enterprise Manager software.

All failed counts are based on the jobId.

Details

The queueModify command allows modification of goal or priority values for job segments in these job types:

- Pickup-dropoff
- Pickups
- QueueMulti

It allows modification of segments in these states:

- Pending job segments
- InProgress jobs up to and including "InProgress Driving", 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.

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
```

Examples

```
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
```

Related Commands

```
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
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 44

queuePickup Command (shortcut: qp) on page 57

queuePickupDropoff Command (shortcut: qpd) on page 60

queueMulti Command (shortcut: qm) on page 53

queueQuery Command (shortcut: qq) on page 65

queueShow Command (shortcut: qs) on page 68

queueShowRobot Command (shortcut: qsr) on page 72

queryFaults Command (shortcut: qf) on page 41

queueMulti Command (shortcut: qm)

Queues the robot 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]

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

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 Data Types on page 26.

Responses

Parameter	Definition
number of goals	Enter the number of goals where you want the mobile robot 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 Enterprise 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 a robot.
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 Enterprise Manager software.

All failed counts are based on the jobId.

For details on the status conditions, see Status Conditions on page 28.

Details

The queueMulti command tells the mobile robot 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
```

Related Commands

```
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 44](#)

[queuePickup Command \(shortcut: qp\) on page 57](#)

[queuePickupDropoff Command \(shortcut: qpd\) on page 60](#)

[queueQuery Command \(shortcut: qq\) on page 65](#)

[queueShow Command \(shortcut: qs\) on page 68](#)

[queueShowRobot Command \(shortcut: qsr\) on page 72](#)

queuePickup Command (shortcut: qp)

Calls any available robot for a pick up request.

Syntax

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

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

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 Data Types on page 26.

Parameter	Definition
goalName	Enter the name of the goal where you want the mobile robot to go for the pickup.
priority	An optional integer value that represents the priority of the pickup request. The higher the number, the sooner Enterprise 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> <completedDate =  
None> <completedTime = None> <failedCount>
```

Details

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

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

All failed counts are based on the jobId.

For details on the status conditions, see Status Conditions on page 28.

Details

The queuePickup command calls any available robot for a pick up request. When the job is at the top of the queue, the mobile robot drives to the specified goal.

If multiple robots are available for the pickup request, the Enterprise Manager determines which robot answers the request based on such factors as which robot 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" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress Allocated Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress BeforePickup Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress BeforeEvery Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress Before Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress Driving Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress After Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress AfterEvery Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 InProgress AfterPickup Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 None None 0
QueueUpdate: PICKUP13 xyz 11 Completed None Goal "z" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 12/19/2011 06:54:34 0
```

Related Commands

queryFaults Command (shortcut: qf) on page 41

queueCancel Command (shortcut: qc) on page 44

queueMulti Command (shortcut: qm) on page 53

queuePickupDropoff Command (shortcut: qpd) on page 60

queueQuery Command (shortcut: qq) on page 65

queueShow Command (shortcut: qs) on page 68

queueShowCompleted Command (shortcut: qsc) on page 70

queueShowRobot Command (shortcut: qsr) on page 72

queuePickupDropoff Command (shortcut: qpd)

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

Syntax

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

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

Parameters

The command parameters are described in the following table.

For details on the data types, see Data Types on page 26.

Parameter	Definition
goal1Name	Enter the name of the goal where you want the mobile robot to go for the pickup.
goal2Name	Enter the name of the goal where you want the mobile robot to go for the dropoff.
priority1	An optional integer value that represents the priority of the pickup request. The higher the number, the sooner Enterprise 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 Enterprise 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=
e=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Allocated> Goal
<"goal1"> <robotName> <queued date> <queued time> <completed date=None> <completed time=
e=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Driving> Goal
<"goal1"> <robotName> <queued date> <queued time> <completed date=None> <completed time=
e=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=
e=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Allocated> Goal
<"goal2"> <robotName> <queued date> <queued time> <completed date=None> <completed time=
e=None> <failed count>
QueueUpdate: <id> <jobId> <priority> <status=InProgress> <substatus=Driving> Goal
<"goal2"> <robotName> <queued date> <queued time> <completed date=None> <completed time=
e=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 Enterprise Manager software.

All failed counts are based on the jobId.

For details on the status conditions, see Status Conditions on page 28.

Details

The queuePickupDropoff command calls any available robot 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 the job identifier. However, note that there is no robot specification parameter in this command—it automatically chooses the most appropriate robot 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.

Examples

```
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> <completed 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> <completed time> <failed count>
```

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

```
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
```

Examples

```
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

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
```


Related Commands

```
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 41](#)

[queueCancel Command \(shortcut: qc\) on page 44](#)

[queueMulti Command \(shortcut: qm\) on page 53](#)

[queuePickup Command \(shortcut: qp\) on page 57](#)

[queueQuery Command \(shortcut: qq\) on page 65](#)

[queueShow Command \(shortcut: qs\) on page 68](#)

[queueShowCompleted Command \(shortcut: qsc\) on page 70](#)

[queueShowRobot Command \(shortcut: qsr\) on page 72](#)

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]

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

Parameters

The command parameters are described in the following table.

For details on the data types, see Data Types on page 26.

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 robot 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 robot 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 robot 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 Enterprise 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 28.

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 41

queueCancel Command (shortcut: qc) on page 44

queueMulti Command (shortcut: qm) on page 53

queuePickup Command (shortcut: qp) on page 57

queuePickupDropoff Command (shortcut: qpd) on page 60

queueShow Command (shortcut: qs) on page 68

queueShowCompleted Command (shortcut: qsc) on page 70

queueShowRobot Command (shortcut: qsr) on page 72

queueShow Command (shortcut: qs)

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

Syntax

queueShow [echoString]

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

Shows all jobs and all robots. To look at a specific job, use `queueQuery`. To look at a specific robot, 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"> <"robotName">
<queued date> <queued time> <completed date> <completed time> <echoString> <failed
count>
EndQueueShow
```

Details

The `queueShow` command provides a listing of all robots connected to the Enterprise Manager, and all jobs in the queue including those that are pending, interrupted, or are currently assigned to the robots. You do not specify a robot with this command. Instead, it lists the information for all robots. If you wish to look at a specific robot, use the `queueShowRobot` command. For details, see the `queueShowRobot` Command (shortcut: `qsr`) on page 72. If you wish to look at a specific job, use the `queueQuery` command. For details, see the `queueQuery` Command (shortcut: `qq`) on page 65.

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

All failed counts are based on the `jobId`.

For details on the status conditions, see `Status Conditions` on page 28.

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 41

queueCancel Command (shortcut: qc) on page 44

queueMulti Command (shortcut: qm) on page 53

queuePickup Command (shortcut: qp) on page 57

queuePickupDropoff Command (shortcut: qpd) on page 60

queueQuery Command (shortcut: qq) on page 65

queueShowCompleted Command (shortcut: qsc) on page 70

queueShowRobot Command (shortcut: qsr) on page 72

queueShowCompleted Command (shortcut: qsc)

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

Syntax

queueshowcompleted [echoString]

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

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

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

The 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 [Data Types](#) on page 26.

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"> <"robotName">
  <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 a robot with this command. Instead, it lists the information for all robots. If you wish to look at a specific robot, use the `queueShowRobot` command. For details, see the `queueShowRobot` Command (shortcut: qsr) on page 72. If you wish to look at a specific job, use the `queueQuery` command. For details, see the `queueQuery` Command (shortcut: qq) on page 65.

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

All failed counts are based on the jobId.

For details on the status conditions, see Status Conditions on page 28.

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 41

queueCancel Command (shortcut: qc) on page 44

queuePickup Command (shortcut: qp) on page 57

queuePickupDropoff Command (shortcut: qpd) on page 60

queueQuery Command (shortcut: qq) on page 65

queueQuery Command (shortcut: qq) on page 65

queueShow Command (shortcut: qs) on page 68

queueShowRobot Command (shortcut: qsr) on page 72

queueShowRobot Command (shortcut: qsr)

Shows the status and substatus of all robots (or, optionally, a specific robot) connected to the Enterprise Manager.

Syntax

queueShowRobot [robotName or "default"] [echoString]

Usage Considerations

This ARCL command is available only on the Enterprise Manager.

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

Parameters

The command parameters are described in the following table.

For details on the data types, see Data Types on page 26.

Parameter	Definition
robotName	Enter the name of the robot. To view all the robots connected to the Enterprise 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 28.

Details

The queueShowRobot command displays the status of the robots currently connected to the Enterprise Manager. Optionally, this command allows you to query a specific robot name, versus the queueShow command, which returns the queue status for all robots along with queue information.

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 68.

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 robot 31:

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

The following example shows the status and substatus of all robots 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 41
- queueCancel Command (shortcut: qc) on page 44
- queueMulti Command (shortcut: qm) on page 53
- queuePickup Command (shortcut: qp) on page 57
- queuePickupDropoff Command (shortcut: qpd) on page 60
- queueQuery Command (shortcut: qq) on page 65
- queueShow Command (shortcut: qs) on page 68
- queueShowCompleted Command (shortcut: qsc) on page 70

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 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. I618-E-01

Printed in USA
0117

18449-000 A