
Adept ARCL Reference Guide

This is a PDF/print version of the Adept ARCL Reference Guide. A Table of Contents is provided so that you can locate the desired topics. Because the Adept ARCL Reference 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 Adept Motivity software, for a description of any recent changes.

Copyright Notice

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

Adept Technology, 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 © 2006-2013 by Adept Technology, Inc. All rights reserved.

Adept, the Adept logo, the Adept Technology logo, AdeptVision, AIM, Blox, Bloxview, FireBlox, Fireview, Meta Controls, MetaControls, Metawire, Soft Machines, and Visual Machines are registered trademarks of Adept Technology, Inc.

Brain on Board is a registered trademark of Adept Technology, Inc. in Germany.

ARAM, ARCL, Adept Enterprise Manager, Motivity, MobileEyes, MobilePlanner, SetNetGo, Adept Lynx, Adept MT400, and Adept SmartFleet EX are trademarks of Adept Technology, Inc.

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 ARCL Connection Setup Parameters 18
 - Outgoing ARCL Commands Parameters 19
 - See Also... 19

- Connect to ARCL Using a Telnet Client** **20**
 - Setting the Connection Parameters 21
 - Connecting to ARCL 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 31
 - Working With Payloads 32

- ARCL Command Reference** **33**
 - See Also... 33
 - getDateime Command 34
 - Syntax 34
 - Usage Considerations 34
 - Parameters 34
 - Examples 34
 - help Command 35

Syntax	35
Usage Considerations	35
Parameters	35
Details	35
Examples	35
payloadQuery Command	36
Syntax	36
Usage Considerations	36
Parameters	36
Responses	36
Details	36
Examples	37
Related Commands	38
payloadSlotCount Command	39
Syntax	39
Usage Considerations	39
Parameters	39
Responses	39
Details	39
Examples	39
Related Commands	40
queryFaults Command	41
Syntax	41
Usage Considerations	41
Parameter	41
Responses	41
Details	41
Example	41
Related Commands	42
queueCancel Command	43
Syntax	43
Usage Considerations	43
Parameters	43
Responses	43
Details	44
Examples	44
Related Commands	45
queueMulti Command	46
Syntax	46
Usage Considerations	46
ARAM Settings	46
Parameters	46
Responses	47
Details	47
Examples	47
Related Commands	48
queuePickup Command	49

Syntax	49
Usage Considerations	49
ARAM Settings	49
Parameters	49
Responses	49
Details	50
Examples	50
Related Commands	50
queuePickupDropoff Command	52
Syntax	52
Usage Considerations	52
Parameters	52
Responses	52
Details	53
Examples	53
Related Commands	55
queueQuery Command	56
Syntax	56
Usage Considerations	56
Parameters	56
Responses	56
Details	57
Examples	57
Related Commands	57
queueShow Command	59
Syntax	59
Usage Considerations	59
Parameters	59
Responses	59
Details	59
Examples	60
Related Commands	60
queueShowCompleted Command	61
Syntax	61
Usage Considerations	61
Parameters	61
Returns	61
Details	61
Examples	62
Related Commands	62
queueShowRobot Command	63
Syntax	63
Usage Considerations	63
Parameters	63
Responses	63
Details	63
Examples	64

Related Commands	64
------------------------	----

Introduction to ARCL

The Advanced Robotics Command Language (ARCL) is a simple, text-based, command-and-response operating language for integrating a fleet of Adept 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 SmartFleet EX appliance and Enterprise Manager software. 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 SmartFleet EX 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 Adept Motivity software, refer to the *Adept Motivity Software User's Guide*.

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 33

Version Requirements

This document pertains to ARAM version 4.4.

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 ARCL, 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 MobileEyes. 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 which are available from the Adept Document Library.

Manual	Description
<u>Adept Robot Safety Guide</u>	Describes safety information for Adept robots.
<u>Adept Motivity User's Guide</u>	Describes the Adept Motivity software, including SetNetGo, MobileEyes, and MobilePlanner.
<u>Adept Lynx Platform User's Guide</u>	Describes the installation, start-up, operation, and maintenance of the Adept mobile robot base.
<u>Adept SmartFleet EX Appliance User's Guide</u>	Describes the installation and operation of the SmartFleet 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 Adept software or hardware, you can access the following information sources on the Adept corporate website:

- For contact information: <http://www.adept.com/contact/americas>
- For product support information: <http://www.adept.com/support/service-and-support/main>
- For user discussions, support, and programming examples: <http://www.adept.com/forum/>
- For further information about Adept Technology, Inc.: <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
- Outgoing ARCL Commands Parameters on page 19
- Set ARCL Parameters in MobilePlanner on page 12

Accessing the Configuration Options

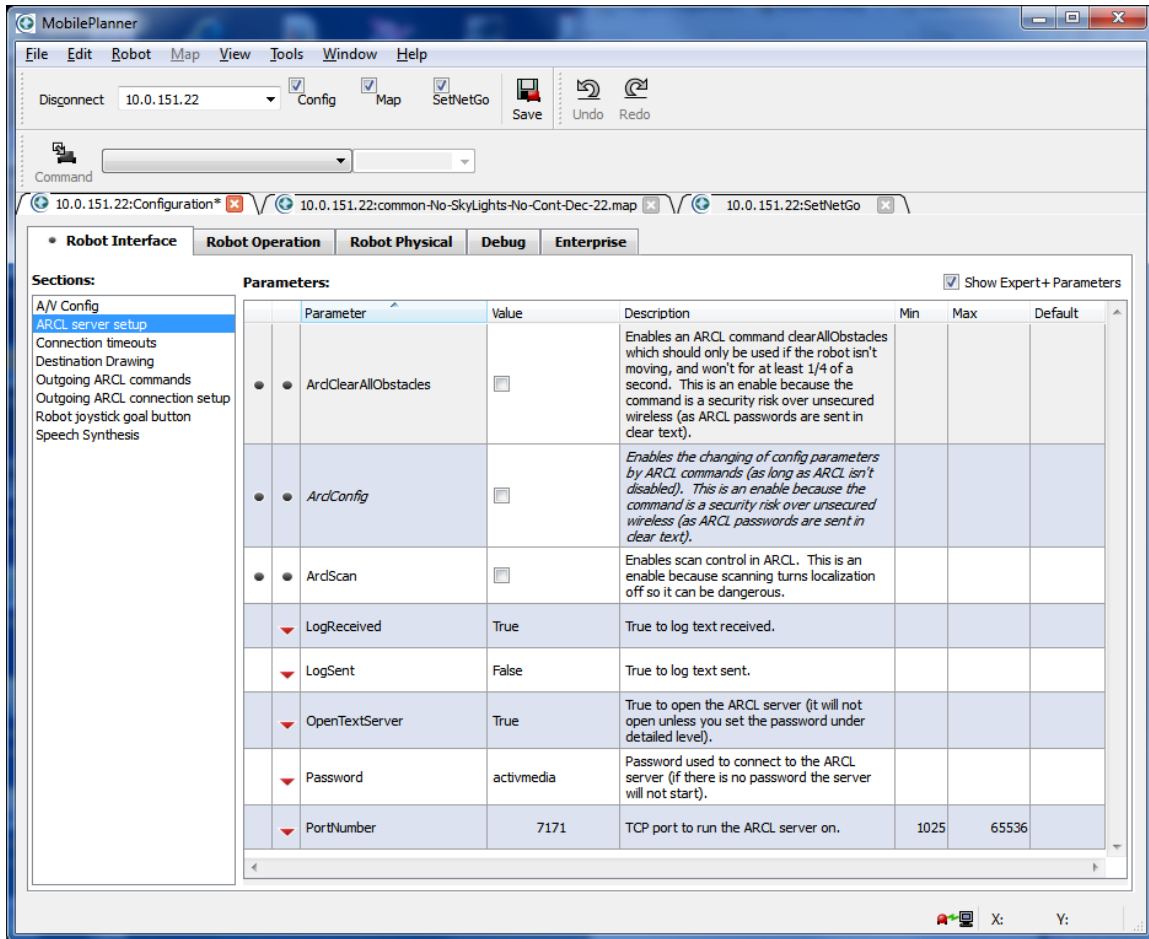
These sections allow you to access configuration parameters that control the ARCL server and its interaction with connected clients.

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 *Adept Motivity 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. These parameters allow you to control the client-server connection between an offboard client process (such as Telnet or PuTTY) and ARCL. The ARCL Server Setup parameters are shown in the following figure.

Accessing the Configuration Options

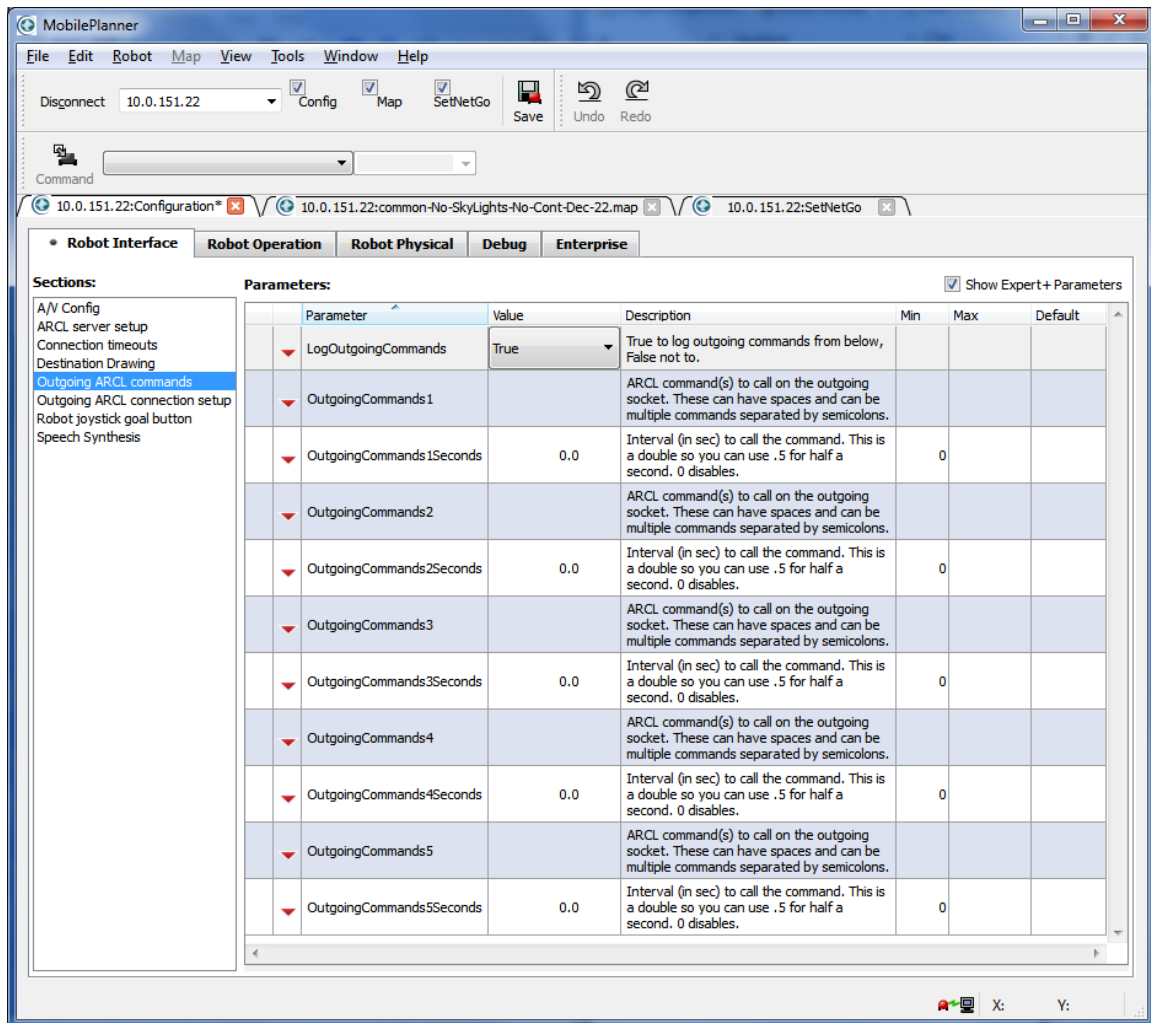


ARCL Server Setup Parameters

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 ARCL Commands** from the Sections column to display the parameters that allow you to automatically generate ARCL commands at regular intervals. The parameters are shown in the following figure. For more details, see [Outgoing ARCL Commands Parameters](#) on page 19.

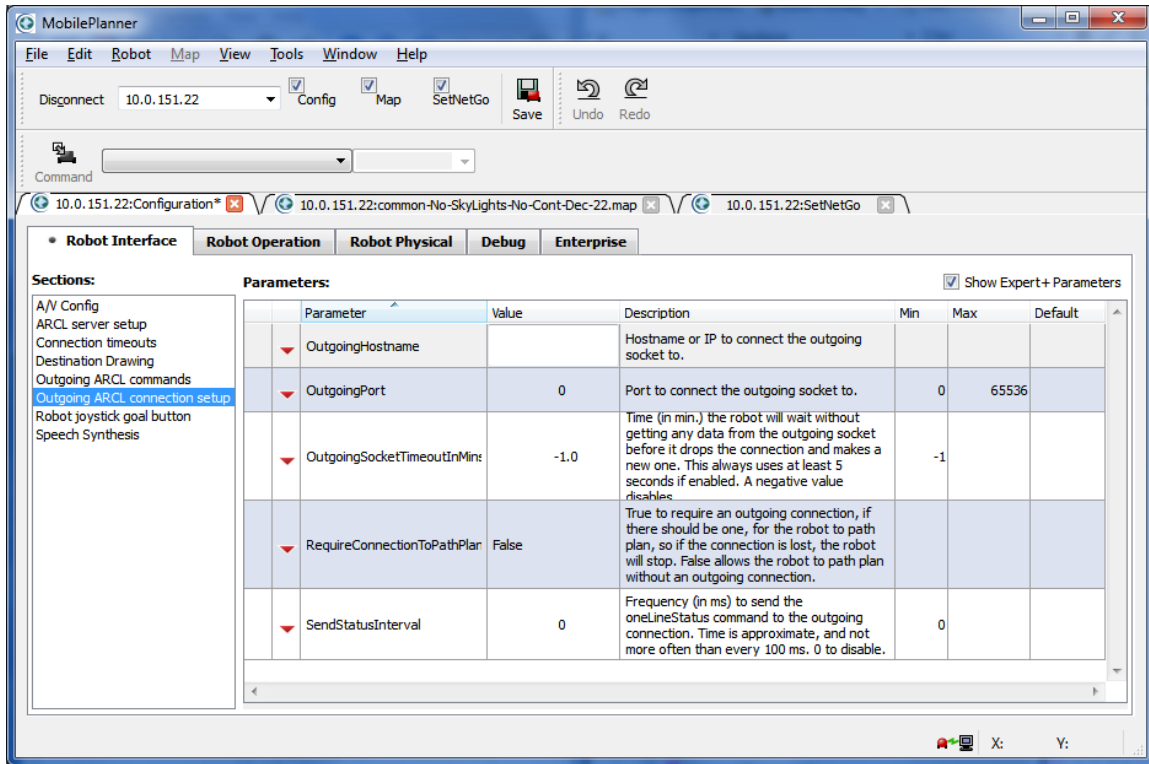
Accessing the Configuration Options



Outgoing ARCL Commands

5. Select Outgoing ARCL Connection Setup from the Sections column to display the parameters that allow you to send data to another server using ARCL commands. The parameters are shown in the following figure. For more details, refer to Set ARCL Parameters in MobilePlanner on page 12.

Accessing the Configuration Options



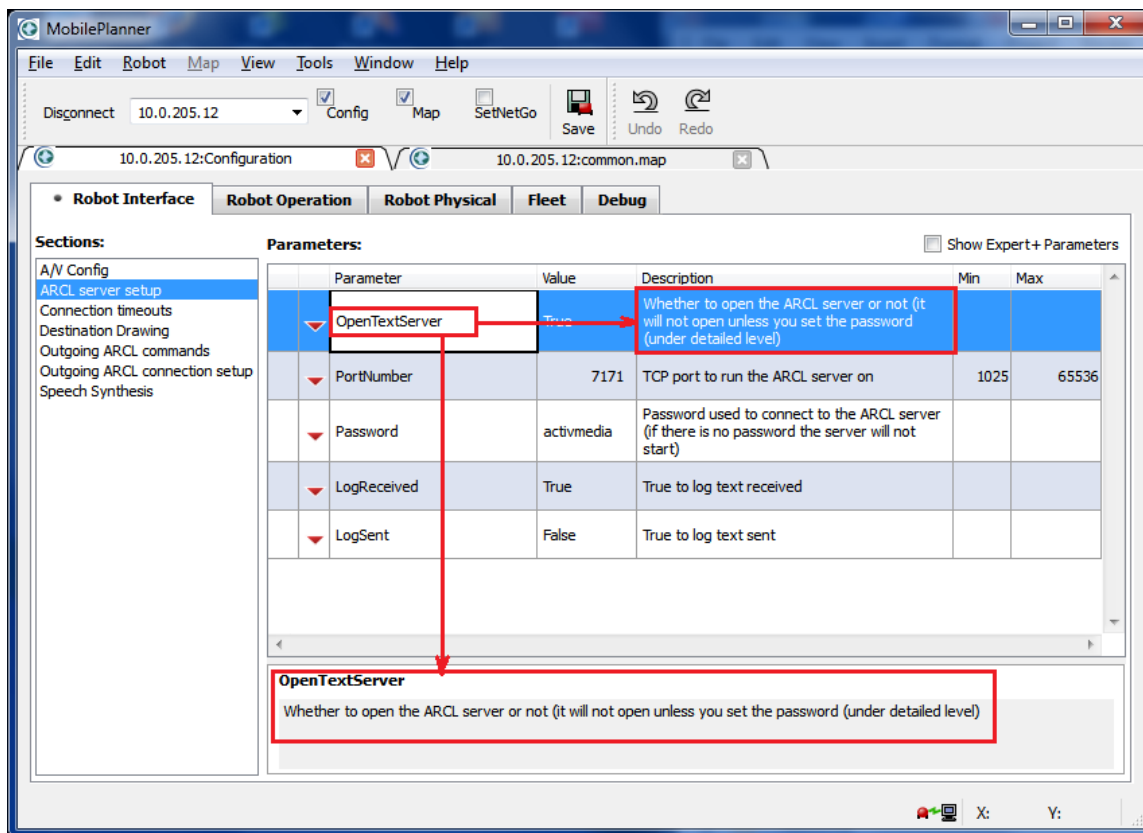
Outgoing ARCL Connection Setup

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 help 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 ARCL Connection Setup Parameters

The Outgoing ARCL Connection Setup parameters are used to instruct the AIV or 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 AIV or Enterprise Manager.

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.

Outgoing ARCL connections can serve several purposes:

- For an AIV, the outgoing ARCL connection can be used to connect to a payload on top of the AIV. The AIV can be configured so that it will not autonomously drive unless the outgoing connection is alive, by setting the Outgoing ARCL Connection setup -> `RequireConnectionToPathPlan` parameter to True.

This is useful when it would be unsafe for the AIV to move at certain times, such as when an automated load or unload is being performed. The payload is responsible for signaling when it is safe to move, so if the connection from the payload to the AIV is lost, it would be unsafe for the AIV to move without knowing the payload status.

There may be hand-shaking involved between the AIV's payload and the factory equipment, to determine when the load or unload is complete, making it safe for the AIV to move.

- For both an AIV and Enterprise Manager, 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 ARCL Commands Parameters

The Outgoing ARCL Command parameters allow you to set the mobile robot 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 semicolon (;). For example, set the OutGoingCommands1 parameter to:

```
doTaskInstant sayInstant "Enabling motors." : enableMotors
```

Then set the OutGoingCommands1Seconds parameter to:

```
60
```

Every 60 seconds, the mobile robot will announce, "Enabling motors" and then attempt to enable the motors.

The outgoing host will receive the ARCL responses:

```
Completed doing instant task: sayInstant "Enabling motors."
```

Then it will respond with, either:

```
Motors enabled
```

or

```
Estop pressed, cannot enable motors
```

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](#)

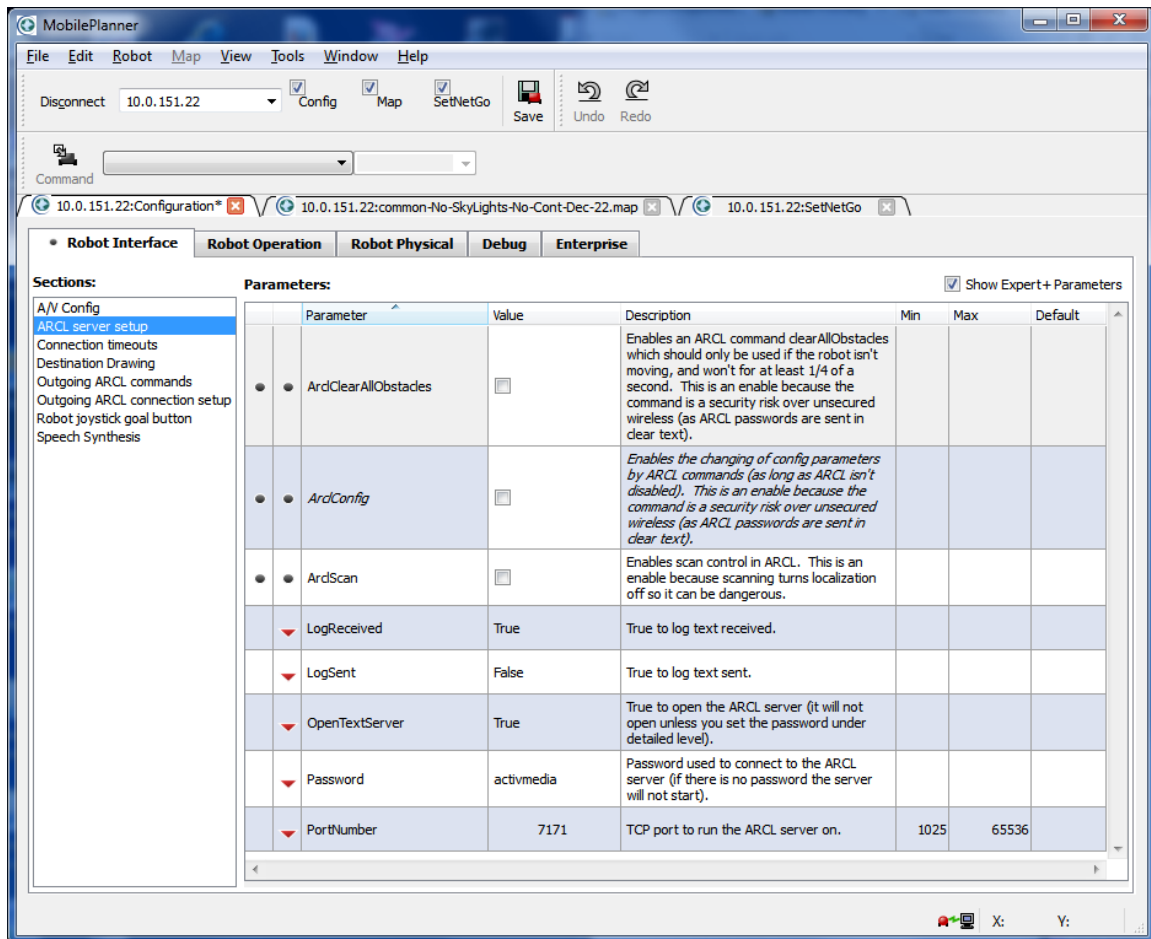
[ARCL Command Reference on page 33](#)

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 [Adept Motivity 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 ARCL

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

1. On a Windows-based PC, open the Command Prompt window.

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 ARCL 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:
  getDateime      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 33.

See Also...

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

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 33](#)

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 ARCL commands. The following topics discuss the use of these commands for certain tasks. To view an alphabetical list and description of each ARCL command, refer to ARCL Command Reference on page 33.

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	31
Working With Payloads	32

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. For more details on the help command, see help Command on page 35.

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 33

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 33.

Document Conventions

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 <goal_name> [priority] [job_id]

In this example, the <goal_name> parameter is required; the [priority] and [job_id] 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, rout-name, 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 are presented in this section.

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:

- For required parameters, 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
DROPOFFgoal_name	string	max length: 127 characters
DROPOFFpriority	integer (signed long)	range: -2147483648 to 2147483647
echo_string ²	string	max length: 127 characters
goal_name	string	max length: 127 characters
job_id 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
PICKUPgoal_name	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
robot_name 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.		

Status and Error Messages

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

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

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

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

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

Status Conditions

Status	Substatus
Pending	None
Available	Available
Available	Parking
Available	Parked
Available	DockParking
Available	DockParked
Interrupted	None
InProgress	UnAllocated
InProgress	Allocated
InProgress	BeforePickup
InProgress	BeforeDropoff
InProgress	EveryBefore
InProgress	Before
InProgress	Buffering
InProgress	Buffered
InProgress	Driving
InProgress	After
InProgress	EveryAfter
InProgress	AfterPickup
InProgress	AfterDropoff
Completed	None
Cancelling	None
Cancelled	None
Cancelling	<application_supplied_cancelReason_string>
Cancelled	<application_supplied_cancelReason_string>
UnAvailable	NotUsingEnterpriseManager

Status Conditions

Status	Substatus
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 31

Working With Payloads on page 32

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 on page 43

queuePickup Command on page 49

queuePickupDropoff Command on page 52

queueQuery Command on page 56

queueShow Command on page 59

queueShowRobot Command on page 63

See Also...

Understanding the Commands on page 25

Using the Queuing Commands on page 31

Working With Payloads on page 32

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 on page 36

payloadSlotCount Command 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 31

Working With Payloads on page 32

ARCL Command Reference

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

getDate Command	34
help Command	35
payloadQuery Command	36
payloadSlotCount Command	39
queryFaults Command	41
queueCancel Command	43
queueMulti Command	46
queuePickup Command	49
queuePickupDropoff Command	52
queueQuery Command	56
queueShow Command	59
queueShowCompleted Command	61
queueShowRobot Command	63

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 33

getDateTime Command

Returns the system date and time.

Syntax

getDateTime

Usage Considerations

This ARCL command is available on the robot and 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
```

help Command

Provides a list and brief description of available ARCL commands.

Syntax

help

Usage Considerations

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

Parameters

This command does not have any parameters.

Details

The help command provides a list and brief description of the available ARCL commands on the connected server or robot. The list shown depends on the current configuration of your server or robot; therefore, it may not show the entire library of commands.

Examples

To view the command list and descriptions, enter the following:

```
help
```

The command returns:

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

```
Commands:
  configAdd      adds to the config
  configParse    parses the added config
  configStart    starts a new config (overwrites previous)
  connectOutgoing (re)connects a socket to the given outside server
  echo          with no args gets echo, with args sets echo
  ...
  quit          closes this connection to the server
  shutdownServer shuts down the server
```

payloadQuery Command

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"] [echo_string]
```

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.
echo_string	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> "[echo_string]"
```

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

Details

This 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
- a specified slot on the "default" 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 robot_name 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
```

payloadQuery Command

```
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 on page 39](#)

payloadSlotCount Command

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

Syntax

payloadSlotCount [robotName or "default"] [echo_string]

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".
echo_string	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> "[echo_string]"
```

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
```

payloadSlotCount Command

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 on page 36](#)

queryFaults Command

Displays the faults associated with the specified robot.

Syntax

queryFaults [robotName]

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 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 display its fault information. Omitting this parameter displays faults for all robots on the Enterprise Manager.

Responses

The command returns the following for a pending item:

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

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 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
EndQueryFaults
```

queryFaults Command

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

queryfaults
RobotFaultQuery: "guiabot_2010_09_20" Fault_Driving_Application fault2 "shortd" "longd"
true false true
RobotFaultQuery: "showpatrolbot1" Fault_EncoderDegraded "Encoder degraded" "The robot's
encoders may be degraded" false true false
RobotFaultQuery: "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
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
RobotFaultQuery: "showpatrolbot1" Fault_Critical_OverTemperatureAnalog "Robot overheated
(analog)" "The robot is too hot (measured by analog) and will shut down shortly" false
true false
RobotFaultQuery: "showpatrolbot1" Fault_Critical_UnderVoltage "Robot battery critically
low" "The robot battery is critically low and will shut down shortly" false true false
RobotFaultQuery: "showpatrolbot1" Fault_Critical_Application fault1 "shortdesc" "long-
desc" false true true
RobotFaultQuery: "showpatrolbot1" Fault_Application fault3 "short" "long" false true true
EndQueryFaults
```

Related Commands

[queueCancel Command on page 43](#)
[queuePickup Command on page 49](#)
[queuePickupDropoff Command on page 52](#)
[queueQuery Command on page 56](#)
[queueShowRobot Command on page 63](#)
[queueShowCompleted Command on page 61](#)

queueCancel Command

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

Syntax

```
queueCancel <type> <value> [echo_string 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 = queries a job with an InProgress status. • pending = queries a job with a Pending status. • interrupted = queries a job with an Interrupted status.
echo_string	An optional string that is appended to each line of the results.
reason	An optional string that can be used to provide a reason for the cancellation.

Responses

The command returns the following for a pending item:

```
queuencancel cancelling <canceltype> <cancelvalue> <echostring> <reason> from queue
```

queueCancel Command

```
QueueCancel: <id> <job_id> <priority> <status = Cancelled> <subStatus = reason_or_None>
Goal <"goal_name"> <"robotname"> <queued date> <queued time> <completed date> <completed
time> <echostring>
```

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

```
queuecancel cancelling <canceltype> <cancelvalue> <echostring> from queue
QueueCancel: <id> <job_id> <priority> <status = Cancelling> <subStatus = reason_or_None>
Goal <"goal_name"> <"robotname"> <queued date> <queued time> <completed date = None>
<completed time = None> <echostring>
QueueUpdate: <id> <job_id> <priority> <status = Interrupted> <subStatus = reason_or_None>
Goal <"goal_name"> <"robotname"> <queued date> <queued time> <completed date = None>
<completed time = None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status = Cancelled> <subStatus = reason_or_None>
Goal <"goal_name"> <"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 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 job_id JOB1 successfully queued
QueueUpdate: PICKUP1 JOB1 10 Pending None Goal "x" "None" 04/15/2013 6:32:47 None None 0
queuecancel jobid job1
queuecancel cancelling "jobid" "job1" "" "" from queue
QueueCancel: PICKUP1 JOB1 10 Cancelled None Goal "x" "None" 04/15/2013 6:32:47 04/15/2013
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 12/16/2011 13:19:07 None
None
queuecancel goal w20 abc
QueueCancel: PICKUP8 JOB8 10 Cancelling None Goal "w20" None 12/16/2011 13:19:07 None
None abc
QueueUpdate: PICKUP8 JOB8 10 Interrupted None Goal "w20" None 12/16/2011 13:19:07 None
None
QueueUpdate: PICKUP8 JOB8 10 Cancelled None Goal "w20" None 12/16/2011 13:19:07
12/16/2011 13:19:13
```

In the following example, a request that is in progress is canceled. The cancel request includes a reason for the cancellation.

```
QueueUpdate: PICKUP8 JOB8 10 InProgress None Goal "w20" "MT-490" 12/16/2011 13:19:07
None None 0
queuecancel goal w20 abc cancelreason
QueueCancel: PICKUP8 JOB8 10 Cancelling cancelreason Goal "w20" None 12/16/2011 13:19:07
None None abc
QueueUpdate: PICKUP8 JOB8 10 Interrupted None Goal "w20" None 12/16/2011 13:19:07 None
None 0
QueueUpdate: PICKUP8 JOB8 10 Cancelled cancelreason Goal "w20" None 12/16/2011 13:19:07
12/16/2011 13:19:13 0
```

Related Commands

[queryFaults Command on page 41](#)

[queueMulti Command on page 46](#)

[queuePickup Command on page 49](#)

[queuePickupDropoff Command on page 52](#)

[queueQuery Command on page 56](#)

[queueShow Command on page 59](#)

[queueShowCompleted Command on page 61](#)

[queueShowRobot Command on page 63](#)

queueMulti Command

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.

Parameter	Definition
number of goals	Enter the number of goals where you want the mobile robot to go. Up to 10 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. Enter 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 10, which can be changed in MobilePlanner.
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 job_id is helpful in tracking the job. If nothing is entered, ARCL generates a random job_id.

Responses

The command returns:

```
QueueMulti: goal "x" with priority 10 id PICKUP1 and job_id JOB1 successfully queued
QueueMulti: goal <"goal1"> with priority <goal1_priority> id <PICKUPid_or_DROPOFFid>
job_id <jobid> successfully queued
QueueMulti: goal <"goal2"> with priority <goal2_priority> id <PICKUPid_or_DROPOFFid>
job_id <jobid> successfully queued and linked to <goal1_PICKUPid_or_DROPOFFid>
:
:
QueueMulti: goal <"goaln"> with priority <goaln_priority> id <PICKUPid_or_DROPOFFid>
job_id <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 deliveries.

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

queueMulti Command

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

Related Commands

[queueCancel Command on page 43](#)

[queuePickup Command on page 49](#)

[queuePickupDropoff Command on page 52](#)

[queueQuery Command on page 56](#)

[queueShow Command on page 59](#)

[queueShowRobot Command on page 63](#)

queuePickup Command

Calls any available robot for a pick up request.

Syntax

queuePickUp <goal_name> [priority or "default"] [job_id]

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
goal_name	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.
job_id	An optional identifier for the specified job. You can use a combination of string characters and integers. The job_id is helpful in tracking the job. If nothing is entered, ARCL generates a random job_id.

Responses

The command returns the following information:

```
queuepickup goal "goal_name" with priority [priority_num] id (pickup_id) and job_id [job_id] successfully queued
```

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

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

queuePickup Command

```
QueueUpdate: <id> <job_id> <priority> <status = InProgress> <substatus = None> Goal
<"goal_name"> <"robot_name"> <queued date> <queued time> <completed date = None> <com-
pleted 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 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 pick up 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 1.

```
queuepickup 1
queuepickup goal "1" with priority 10 id PICKUP2 and job_id JOB2 successfully queued
QueueUpdate: PICKUP2 JOB2 10 Pending None Goal "1" "None" 11/14/2012 11:25:34 None None 0
QueueUpdate: PICKUP2 JOB2 10 InProgress UnAllocated Goal "1" "21" 11/14/2012 11:25:34
None None 0
QueueUpdate: PICKUP2 JOB2 10 InProgress Allocated Goal "1" "21" 11/14/2012 11:25:34 None
None 0
QueueUpdate: PICKUP2 JOB2 10 InProgress Driving Goal "1" "21" 11/14/2012 11:25:34 None
None 0
QueueUpdate: PICKUP2 JOB2 10 Completed None Goal "1" "21" 11/14/2012 11:25:34 11/14/2012
11:25:34 0
```

Related Commands

[queryFaults Command on page 41](#)

[queueCancel Command on page 43](#)

[queueMulti Command on page 46](#)

[queuePickupDropoff Command on page 52](#)

[queueQuery Command on page 56](#)

[queueShow Command on page 59](#)

[queueShowCompleted Command on page 61](#)

queueShowRobot Command on page 63

queuePickupDropoff Command

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

Syntax

```
queuePickupDropoff <goal1_name> <goal2_name> [priority1 or "default"] [priority2 or "default"]
[job_id]
```

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
goal1_name	Enter the name of the goal where you want the mobile robot to go for the pickup.
goal2_name	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.
job_id	An optional identifier for the specified job. You can use a combination of string characters and integers. The job_id is helpful in tracking the job. If nothing is entered, ARCL generates a random job_id.

Responses

The command returns the following information:

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

The PICKUPid and DROPOFFid are assigned by the system.

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

```
QueueUpdate: <id> <job_id> <priority> <status=Pending> <substatus=None> Goal <"goal1">
<robot_name> <queued date> <queued time> <completed date=None> <completed time=None>
<failed count>
QueueUpdate: <id> <job_id> <priority> <status=Pending> <substatus=ID_<id>> Goal
<"goal2"> <robot_name> <queued date> <queued time> <completed date=None> <completed
time=None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=UnAllocated> Goal
<"goal1"> <robot_name> <queued date> <queued time> <completed date=None> <completed
time=None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=Allocated> Goal
<"goal1"> <robot_name> <queued date> <queued time> <completed date=None> <completed
time=None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=Driving> Goal
<"goal1"> <robot_name> <queued date> <queued time> <completed date=None> <completed
time=None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status=Completed> <substatus=None> Goal <"goal1">
<robot_name> <queued date> <queued time> <completed date> <completed time> <failed
count>
QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=UnAllocated> Goal
<"goal2"> <robot_name> <queued date> <queued time> <completed date=None> <completed
time=None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=Allocated> Goal
<"goal2"> <robot_name> <queued date> <queued time> <completed date=None> <completed
time=None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status=InProgress> <substatus=Driving> Goal
<"goal2"> <robot_name> <queued date> <queued time> <completed date=None> <completed
time=None> <failed count>
QueueUpdate: <id> <job_id> <priority> <status=Completed> <substatus=None> Goal <"goal2">
<robot_name> <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 delivery. 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 selects the first-available robot in the system.

Examples

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

```
queuepickupdropoff x y 10 11 abc
```

queuePickupDropoff Command

```
queuepickupdropoff goals "x" and "y" with priorities 10 and 11 ids PICKUP22 and DROPOFF23
job_id abc successfully queued
QueueUpdate: PICKUP22 abc 10 Pending None Goal "x" None 12/19/2011 07:24:02 None None 0
QueueUpdate: DROPOFF23 abc 11 Pending ID_PICKUP22 Goal "y" "None" 12/19/2011 07:24:29
None None 0
QueueUpdate: PICKUP22 abc 10 InProgress UnAllocated Goal "x" "Adept_Telepresence_Robot"
12/19/2011 07:24:02 None None 0
QueueUpdate: PICKUP22 abc 10 InProgress Allocated Goal "x" "Adept_Telepresence_Robot"
12/19/2011 07:24:02 None None 0
QueueUpdate: PICKUP22 abc 10 InProgress Driving Goal "x" "Adept_Telepresence_Robot"
12/19/2011 07:24:02 None None 0
QueueUpdate: PICKUP22 abc 10 Completed None Goal "x" "Adept_Telepresence_Robot"
12/19/2011 07:24:02 12/19/2011 07:24:29 0
QueueUpdate: DROPOFF23 abc 11 InProgress UnAllocated Goal "y" "Adept_Telepresence_Robot"
12/19/2011 07:24:29 None None 0
QueueUpdate: DROPOFF23 abc 11 InProgress Allocated Goal "y" "Adept_Telepresence_Robot"
12/19/2011 07:24:29 None None 0
QueueUpdate: DROPOFF23 abc 11 InProgress Driving Goal "y" "Adept_Telepresence_Robot"
12/19/2011 07:24:29 None None 0
QueueUpdate: DROPOFF23 abc 11 Completed None Goal "y" "Adept_Telepresence_Robot"
12/19/2011 07:24:29 12/19/2011 07:24:52 0
```

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 ID_PICKUP_12 Goal "y" "None" 08/16/2012 14:32:54
None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress UnAllocated Goal "x" "mt400" 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" "mt400" 08/16/2012 14:33:01
None None 0
QueueUpdate: DROPOFF15 JOB14 20 Pending ID_PICKUP14 Goal "t" "mt400" 08/16/2012 14:33:01
None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress Allocated Goal "x" "mt400" 08/16/2012 14:32:54
None None 0
QueueUpdate: PICKUP12 JOB12 10 InProgress Driving Goal "x" "mt400" 08/16/2012 14:32:54
None None 0
QueueUpdate: PICKUP12 JOB12 10 Completed None Goal "x" "mt400" 08/16/2012 14:32:54
08/16/2012 14:33:15 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress UnAllocated Goal "y" "mt400" 08/16/2012
14:32:54 None None 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress Allocated Goal "y" "mt400" 08/16/2012 14:32:54
None None 0
QueueUpdate: DROPOFF13 JOB12 20 InProgress Driving Goal "y" "mt400" 08/16/2012 14:32:54
None None 0
QueueUpdate: DROPOFF13 JOB12 20 Completed None Goal "y" "mt400" 08/16/2012 14:32:54
08/16/2012 14:33:27 0
```

queuePickupDropoff Command

```
QueueUpdate: PICKUP14 JOB14 10 Completed None Goal "y" "mt400" 08/16/2012 14:33:01
08/16/2012 14:33:27 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress UnAllocated Goal "t" "mt400" 08/16/2012
14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress Allocated Goal "t" "mt400" 08/16/2012
14:33:01 None None 0
QueueUpdate: DROPOFF15 JOB14 20 InProgress Driving Goal "t" "mt400" 08/16/2012 14:33:01
None None 0
QueueUpdate: DROPOFF15 JOB14 20 Completed None Goal "t" "mt400" 08/16/2012 14:33:01
08/16/2012 14:33:35 0
```

Related Commands

[queryFaults Command on page 41](#)
[queueCancel Command on page 43](#)
[queueMulti Command on page 46](#)
[queuePickup Command on page 49](#)
[queueQuery Command on page 56](#)
[queueShow Command on page 59](#)
[queueShowCompleted Command on page 61](#)
[queueShowRobot Command on page 63](#)

queueQuery Command

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

Syntax

queueQuery <type> <value> [echo_string]

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.
echo_string	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 <"goal_name"> <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" "Adept_Telepresence_Robot"
12/19/2011 06:54:18 12/19/2011 06:54:34 xyz 0
EndQueueQuery
```

Related Commands

[queryFaults Command on page 41](#)

[queueCancel Command on page 43](#)

[queueMulti Command on page 46](#)

[queuePickup Command on page 49](#)

[queuePickupDropoff Command on page 52](#)

[queueShow Command on page 59](#)

[queueShowCompleted Command on page 61](#)

[queueShowRobot Command on page 63](#)

queueShow Command

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 [echo_string]

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.

For details on the data types, see [Data Types](#) on page 26.

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

Responses

The command returns the following information:

```
QueueRobot: <robotname> <robotstatus> <robotsubstatus> <echo_string>
QueueShow: <id> <jobid> <priority> <status> <substatus> Goal <"goal_name"> <"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 on page 63. If you wish to look at a specific job, use the `queueQuery` command. For details, see the `queueQuery` Command on page 56.

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 on page 41](#)
[queueCancel Command on page 43](#)
[queueMulti Command on page 46](#)
[queuePickup Command on page 49](#)
[queuePickupDropoff Command on page 52](#)
[queueQuery Command on page 56](#)
[queueShowCompleted Command on page 61](#)
[queueShowRobot Command on page 63](#)

queueShowCompleted Command

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

Syntax

queueshowcompleted [echo_string]

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`.

Parameters

The command parameters are described in the following table.

For details on the data types, see [Data Types](#) on page 26.

Parameter	Definition
echo_string	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 <"goal_name"> <"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 on page 63. If you wish to look at a specific job, use the `queueQuery` command. For details, see the `queueQuery` Command on page 56.

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 on page 41](#)

[queueCancel Command on page 43](#)

[queuePickup Command on page 49](#)

[queuePickupDropoff Command on page 52](#)

[queueQuery Command on page 56](#)

[queueQuery Command on page 56](#)

[queueShow Command on page 59](#)

[queueShowRobot Command on page 63](#)

queueShowRobot Command

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

Syntax

queueShowRobot [robot_name or "default"] [echo_string]

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
robot_name	Enter the name of the robot. To view all the robots connected to the Enterprise Manager, omit this parameter or enter "default".
echo_string	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: "robot_name" robot_status robot_substatus 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 on page 59.

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 "echothis":

```
queueshowrobot default echothis
QueueRobot: "31" Available Available echothis
QueueRobot: "32" Available Available echothis
QueueRobot: "33" Available Available echothis
QueueRobot: "34" Available Available echothis
QueueRobot: "35" Available Available echothis
QueueRobot: "36" Available Available echothis
EndQueueShowRobot
```

Related Commands

[queryFaults Command on page 41](#)

[queueCancel Command on page 43](#)

[queueMulti Command on page 46](#)

[queuePickup Command on page 49](#)

[queuePickupDropoff Command on page 52](#)

[queueQuery Command on page 56](#)

[queueShow Command on page 59](#)

[queueShowCompleted Command on page 61](#)