VWorks Automation Control Software

User Guide

Original Instructions

Agilent Technologies

Notices

User Guide Part Number

G5415-90063

Edition

Revision I, October 2014

Contact Information

Agilent Technologies Inc.
Automation Solutions
5301 Stevens Creek Blvd.
Santa Clara, CA 95051
USA

Technical Support: 1.800.979.4811 (US only) or +1.408.345.8011 (rest of the world)
service.automation@agilent.com

Customer Service: 1.866.428.9811
or +1.408.345.8356
orders.automation@agilent.com

European Service: +44 (0) 845 712 5292
euroservice.automation@agilent.com

Documentation feedback:
documentation.automation@agilent.com

Web:
www.agilent.com/lifesciences/automation

Acknowledgements

Microsoft® and Windows® are either registered trademarks or trademarks of the Microsoft Corporation in the United States and other countries.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

Contents

Preface

This preface contains the following topics:

About this guide

Who should read this guide

This user guide is for people with the following job roles:

Job role	Responsibilities
Integrator	Someone who writes software and configures hardware controlled by VWorks software.
Lab manager, administrator, or technician	Someone who is responsible for: • Developing the applications that are run using VWorks software • Developing training materials and standard operating procedures for operators
Operator	Someone who performs the daily production work using VWorks software and solves routine problems. Your organization may choose to create its own procedures for operators including the procedures in this guide.

What this guide covers

This guide explains how to use the VWorks software. This guide does not provide procedures for setting up, operating, or troubleshooting devices using the device diagnostic software. For information on devices and how to use the diagnostic software, see the device user guide.

What is new in this revision

Feature and description	See...
New method is added to the plateDB object.	"Creating a protocol: advanced topics" on page 73

Software version

This guide describes VWorks Automation Control 12.2 and later.

Related guides

This guide should be used in conjunction with the following user documents:

- *VWorks Automation Control Setup Guide*. Explains how to install the VWorks software, define labware, specify pipetting speed and accuracy, track and manage labware in storage, manage user accounts, and use VWorks ActiveX control.

- *Automation Control Unit User Guide*. Describes the functions of the Automation Control Unit and explains how to operate an automation system that uses the Automation Control Unit.
- *Agilent lab automation system user documentation*. Presents safety information, and explains how to set up, operate, maintain, and troubleshoot the lab automation system, such as the BioCel System, BenchBot Workstation, or the BenchCel Workstation.
- *Agilent device user documentation*. Explains how to set up and use Agilent devices.
- *Third-party device user documentation*. Explains how to set up and use third-party devices.

Related information

For information about...	See...
Accessing related user guides	"Accessing Automation Solutions user guides" on page xi
Reporting problems with the software	"Reporting problems" on page 661

Accessing Automation Solutions user guides

About this topic

This topic describes the different formats of Automation Solutions user information and explains how to access the user information.

Where to find user information

The Automation Solutions user information is available in the following locations:

- *Knowledge base*. The help system that contains information about all of the Automation Solutions products is available from the Help menu within the VWorks software.
- *PDF files*. The PDF files of the user guides are installed with the VWorks software and are on the software CD that is supplied with the product. A PDF viewer is required to open a user guide in PDF format. You can download a free PDF viewer from the internet. For information about using PDF documents, see the user documentation for the PDF viewer.
- *Agilent Technologies website*. You can search the online knowledge base or download the latest version of any PDF file from the Agilent Technologies website at www.agilent.com/lifesciences/automation.

Accessing safety information

Safety information for the Agilent Technologies devices appears in the corresponding device safety guide or user guide.

You can also search the knowledge base or the PDF files for safety information.

Using the knowledge base

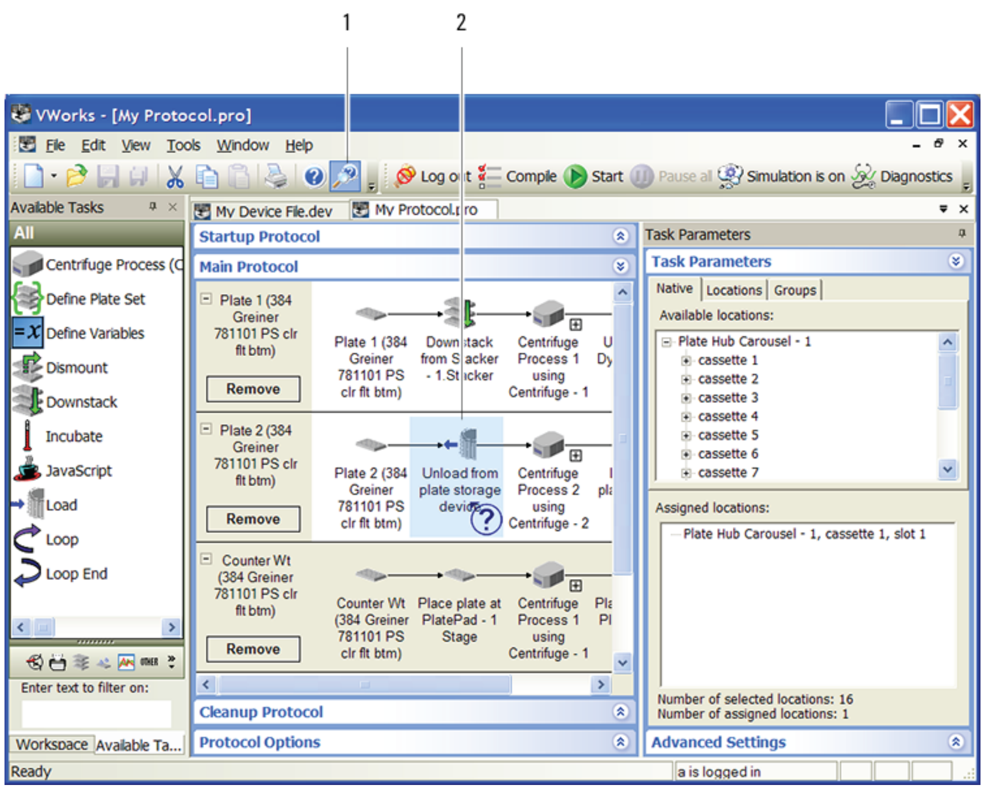
Knowledge base topics are displayed using web browser software such as Microsoft Internet Explorer and Mozilla Firefox.

Note: If you want to use Internet Explorer to display the topics, you might have to allow local files to run active content (scripts and ActiveX controls). To do this, in Internet Explorer, open the **Internet Options** dialog box. Click the **Advanced** tab, locate the **Security** section, and select **Allow active content to run in files on my computer.**

To open the knowledge base, do one of the following:

- From within VWorks software, select **Help** > **Knowledge Base** or press F1.
- From the Windows desktop, select **Start** > **All Programs > Agilent Technologies > VWorks** > **User Guides** > **Knowledge Base.**

Opening the help topic for an area in the VWorks window



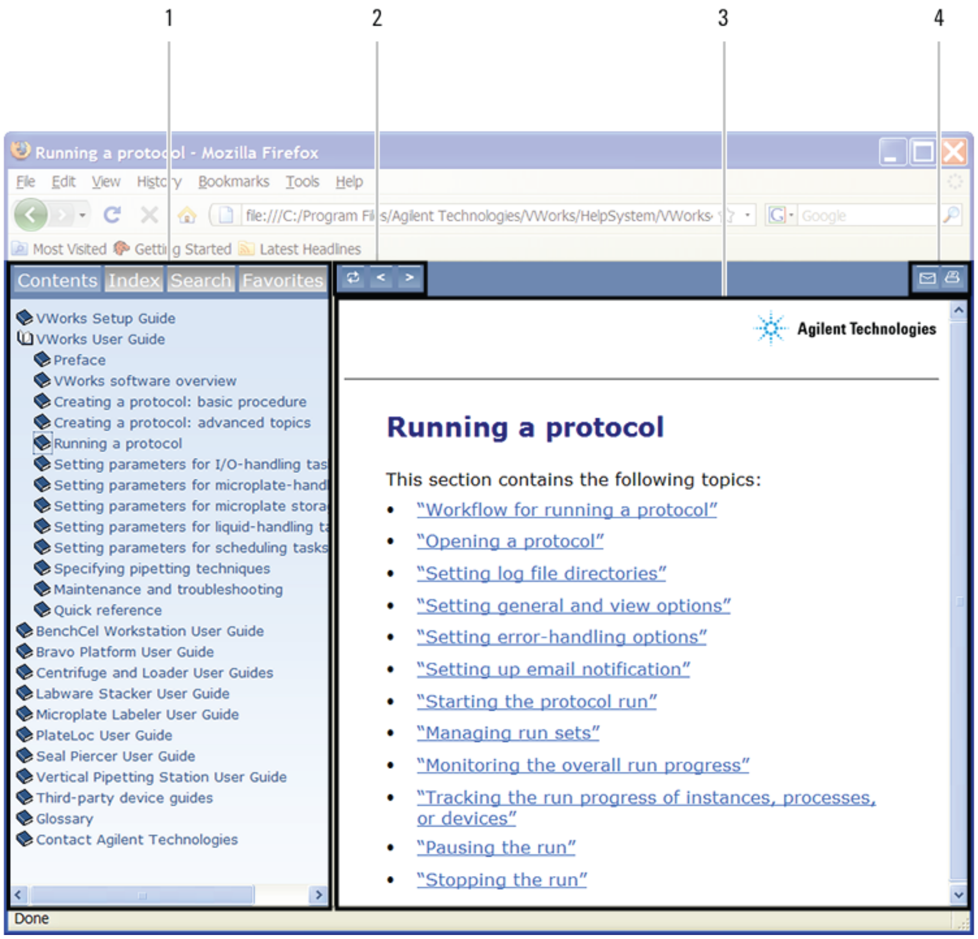
To access the context-sensitive help feature:

1 In the main window of the VWorks software, click the help button.

The pointer changes to ?. Notice that the different icons or areas are highlighted as you move the pointer over them.

2 Click an icon or area of interest. The relevant topic or document opens.

Features in the Knowledge Base window



Item	Feature
1	*Navigation area*. Consists of four tabs: • *Contents*. Lists all the books and the table of contents of the books. • *Index*. Displays the index entries of all of the books. • *Search*. Allows you to search the Knowledge Base (all products) using keywords. You can narrow the search by product. • *Favorites*. Contains bookmarks you have created.
2	*Navigation buttons*. Enable you to navigate through the next or previous topics listed in the Contents tab.
3	*Content area*. Displays the selected online help topic.
4	*Toolbar buttons*. Enable you to print the topic or send documentation feedback by email.

Related information

For information about...	See...
Who should read this guide, what this guide covers, software version covered, and related guides	“About this guide” on page x
Reporting problems with the software	“Reporting problems” on page 661

1
VWorks software overview

This chapter contains the following topics:

VWorks software description

Description

The VWorks Automation Control software (or VWorks software) manages and controls both standalone devices and integrated devices in a laboratory automation system. You use VWorks software to:

- *Set up and manage user accounts.* You can set up different user accounts to enforce access policies.
- *Define labware.* Labware definitions describe the labware you will use during protocol runs.
- *Manage inventory.* The Inventory Editor helps you track barcodes and labware as you move them into and out of storage or incubation.
- *Set up devices.* During setup, you need to add standalone or integrated devices in the software.
- *Create protocols.* Protocols determine the sequence of tasks you want to automate in a run. For example, you can use a protocol to apply barcode labels to 100 microplates.
- *Run, pause, monitor, and stop protocols.* You can start, pause, monitor, and stop a protocol run from the controlling computer.

Related information

For information about...	See...
VWorks software installation and setup	*VWorks Automation Control Setup Guide*
Defining labware and creating liquid classes	*VWorks Automation Control Setup Guide*
Managing and tracking labware in storage	*VWorks Automation Control Setup Guide*
Managing user accounts	*VWorks Automation Control Setup Guide*
Using VWorks ActiveX control	*VWorks Automation Control Setup Guide*
Device setup	Device user guide or third-party device driver user guide

Supported devices

The VWorks software manages and controls all Automation Solutions devices and a large number of third-party devices.

Automation Solutions devices

All of Automation Solutions devices can be used with the VWorks software, including:

- BenchBot Robot
- BenchCel Microplate Handling Workstation
- BioCel System
- Bravo Automated Liquid Handling Platform
- Direct Drive Robot
- Labware Stacker
- Microplate Barcode Labeler
- Microplate Centrifuge
- Microplate Seal Piercer
- PlateLoc Thermal Microplate Sealer
- Vertical Pipetting Station

Third-party devices

A large number of third-party devices can be used with the VWorks software. For a comprehensive list, go to www.agilent.com/lifesciences/automation.

If you would like to add other devices, contact Automation Solutions Customer Service.

Related information

For information about...	See...
Device setup	Device user guide or the third-party device driver user guide
Relationship of VWorks software components	“Relationship of VWorks components” on page 4
Introduction to the VWorks software user interface	“Overview of VWorks software user interface” on page 8
VWorks software installation and setup	*VWorks Automation Control Setup Guide*

Relationship of VWorks components

The VWorks software uses different components (files and databases) to run protocols. It is important to understand the way each of the components in the software relate. Changing settings or options in one component will affect one or more of the other components.

Component descriptions

The following table lists and describes the VWorks software components.

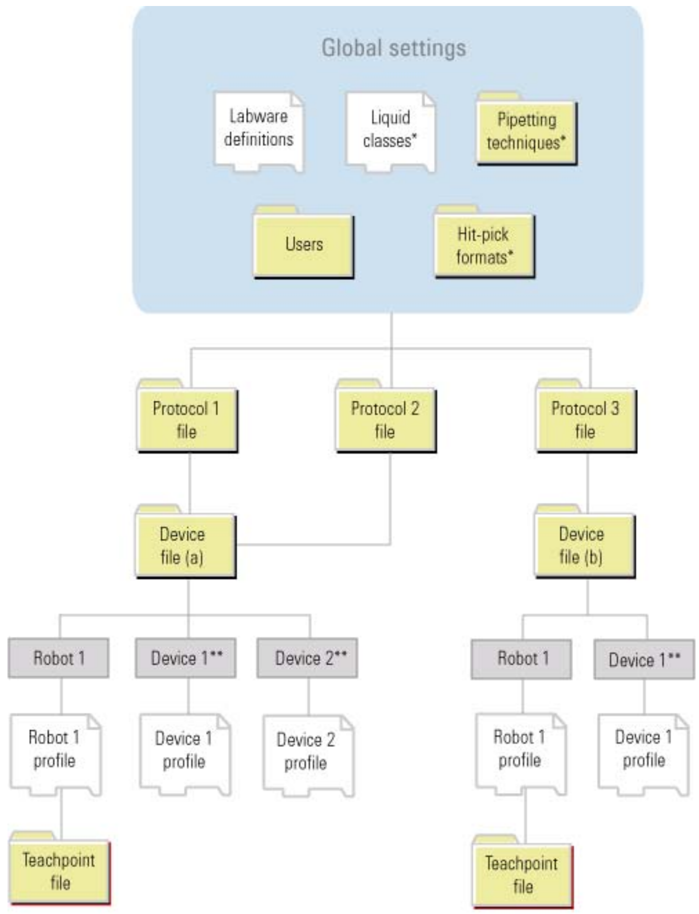
Note: The device, teachpoint, and protocol files are stored in locations you specify. Agilent Technologies recommends that you create folders within the c:\VWorks Workspace folder for storing these files.

Component	Description	Extension	Opening this component loads...
Device file	A file that contains: • The list of devices the software will communicate with and control • Configuration information of each device • Communication settings (profile)	.dev	• Device profile • Teachpoint file (applicable to some devices such as the robot in the BioCel System and the BenchCel device)
Device profile	A collection of settings, stored in the Windows registry, that the VWorks software uses to control a specific device.	None	Teachpoint file (applicable to some devices such as the robot in the BioCel System and the BenchCel device)
Teachpoint file	A device-dependent file that contains your teachpoint settings.	.xml	Teachpoint definitions
Protocol file	A file that contains instructions for performing a run.	.pro	• VWorks software (if it is not already open) • Device file
Labware definition	Labware properties stored in the Windows registry.	None	Labware definitions
Liquid class	Pipetting settings, setup for different liquid types, stored in the Windows registry.	None	Liquid class information
Pipette technique	A file that specifies the x- and y-axis offset when pipetting.	.xml	Pipette x- and y-axis offset information
Hit-pick format	A file used by the Hit Pick Replication task and specifies the dispense pattern in destination microplates.	.xml	Dispense information in destination microplates

Component	Description	Extension	Opening this component loads...
	A file that contains information about a user, including the user name, password, and account privileges.	.xml	Account information of a specific user

The following diagram summarizes the relationship of the components. Notice the following:

- Labware definitions and user information are used by all protocols. Liquid classes, pipetting techniques, and hit-pick formats are used by protocols containing liquid-handling tasks. In particular, hit-pick formats are used by the Hit Pick Replication task.
- Each protocol references a single device file that contains one or more devices.
- More than one protocol can reference the same device file.
- Each device (or robot) references a single profile.
- Some devices, such as the robot in the BioCel System and BenchCel device, reference a single teachpoint file.



* Applicable to protocols containing liquid-handling tasks only
** Individual device, not a device type

Impact of changes to the components

The following table describes the consequences of making changes to one or more components.

If you...	Then...
Make a change to the teachpoint file	• All profiles that use that teachpoint file are affected • All device files that use those profiles are affected • All protocols that use those device files are affected
Create a new profile	You must specify the new profile in your device file

If you...	Then...
Want to use two different teachpoint files	You must create two: • Device files • Profiles • Teachpoint files • Protocol files
Want to copy a protocol to another system or computer	Use the **File > Export** command to export all components: • Protocol file • Device file • Device profiles • Labware definitions and classes • Liquid classes • Pipette techniques • Hit-pick format or input files • Plate map database The exported .vzp file can be imported in another computer.

Related information

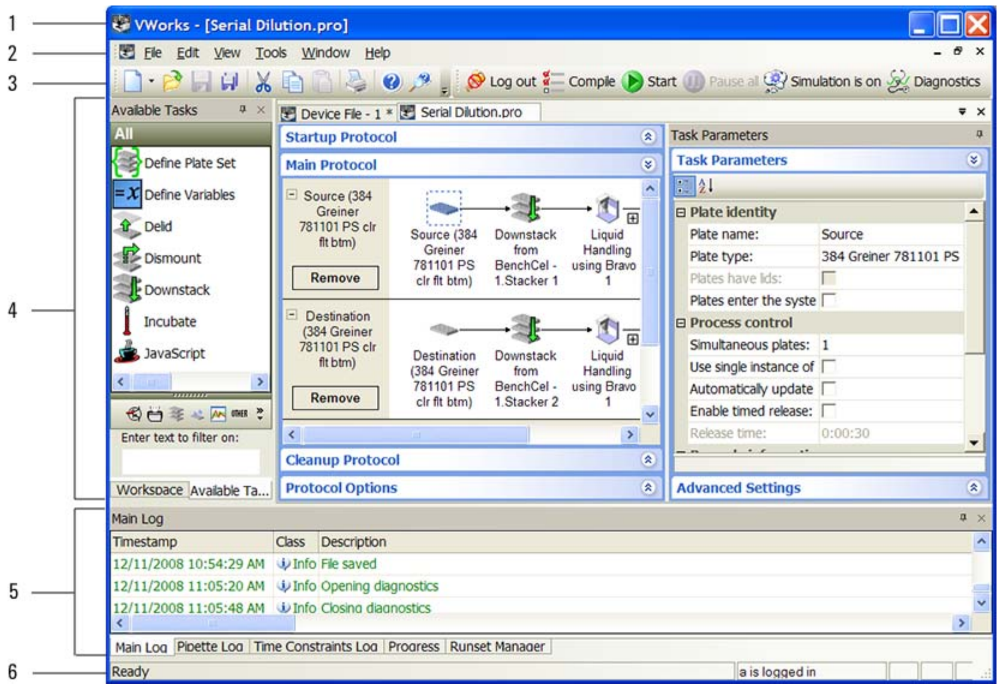
For information about...	See...
VWorks software description	"VWorks software description" on page 2
VWorks software user interface	"Overview of VWorks software user interface" on page 8
Supported devices	"Supported devices" on page 3
Device setup, including creating profiles and setting teachpoints	Device user guide or the third-party device driver user guide
VWorks software installation and setup	*VWorks Automation Control Setup Guide*
Defining labware and creating liquid classes	*VWorks Automation Control Setup Guide*
Managing and tracking labware in storage	*VWorks Automation Control Setup Guide*
Managing user accounts	*VWorks Automation Control Setup Guide*
Using VWorks ActiveX control	*VWorks Automation Control Setup Guide*
Exporting protocols and associated components	"Exporting and importing protocols and associated components" on page 626

Overview of VWorks software user interface

The VWorks software user interface consists of menus and menu commands, toolbars, tabbed areas, and a status bar. The content of each of these items can change depending on whether you are viewing a device file or protocol file.

Basic terminology

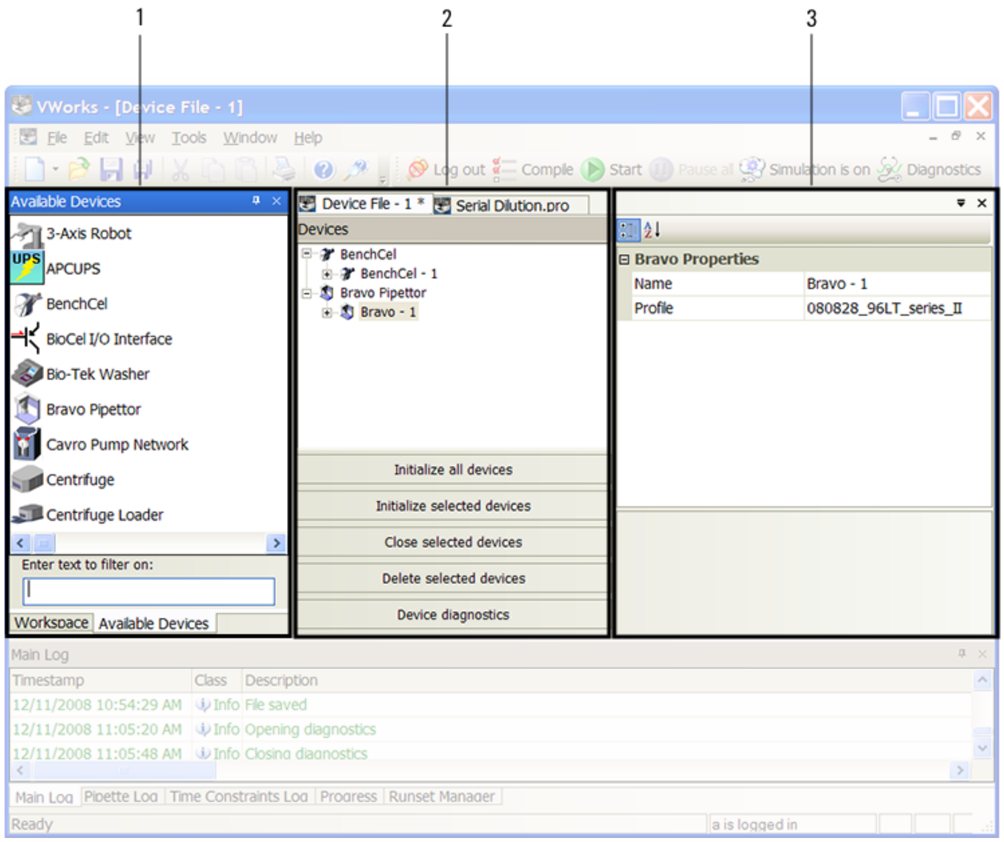
The following diagram shows the basic VWorks software user interface elements.



Item	Terminology	Description
1	Title bar	Displays the name of the window.
2	Menus	Lists menu commands.
3	Toolbars	Displays button commands.
4	Work area	Displays either device files or protocols.
5	Log and progress area	Displays the Main Log, Pipette Log, Time Constraint Log, Progress, and Runset Manager. You use these tabs primarily to set up multiple protocol runs in a sequence and monitor various aspects of the run.
6	Status bar	Displays the state of the software.

Device file terminology

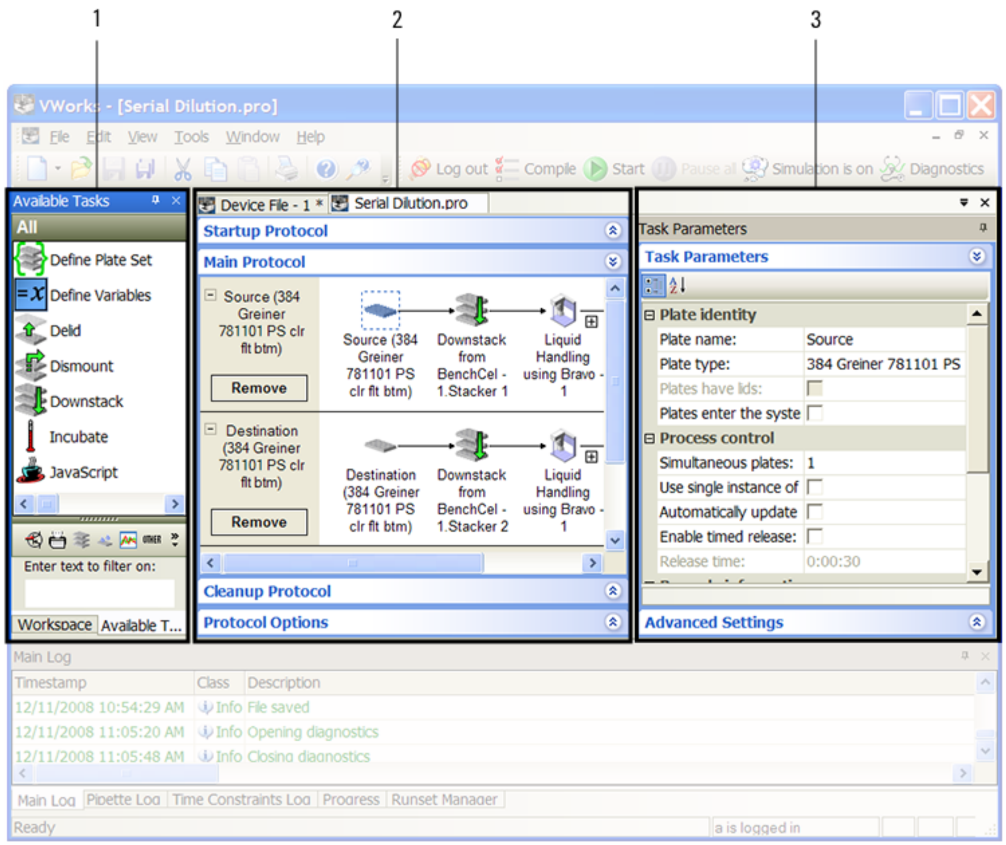
To display device file information, click the tab that displays the name of the device file. The following diagram shows the device file user-interface terminology. In the example, the device file name is Device File - 1.



Item	Terminology	Description
1	Available Devices area	Displays the following tabs: • *Available Devices*. The list of devices you can add to the device file. • *Workspace*. The list of protocols and devices that are currently open.
2	Device file area	Displays the list of devices that are added to the device file. The area also allows you to: • Initialize all devices. • Initialize selected devices. • Close selected devices. • Delete selected devices. • Open the diagnostics software of the selected device.
3	Device properties area	Displays the properties you can set for the selected device.

Protocol terminology

To display the protocol information, click the tab that displays the name of the protocol. The following diagram shows the protocol user-interface terminology. In the example, the protocol name is Serial Dilution.



Item	Terminology	Description
1	Available Tasks area	Displays the following tabs: • *Available Tasks*. The list of tasks you can add to a protocol. The list of tasks can vary, depending on the devices added in the device file and the process or subprocess selected. *Note:* You can use the filter buttons beneath the tasks to display only the tasks in a selected category. • *Workspace*. The list of protocols and devices that are currently open.

Item	Terminology	Description
2	Protocol area	Displays the following: • *Startup Protocol*. Contains tasks that must be run before the main protocol starts. • *Main Protocol*. Contains tasks of the protocol. • *Cleanup Protocol*. Contains tasks that must be run after the main protocol finishes. • *Protocol Options*. References the device file and contains additional information associated with the protocol.
3	Task Parameters area	Displays one or more of the following, depending on the task selected: • *Task Parameters*. Contains parameters associated with the selected process plate or task. • *Custom Parameters*. Allows you to create variables to be used by the selected task. Might also contain parameters or options associated with the task. • *Device Selection*. Contains the list of devices that will perform the selected task. Also allows you to set up a pool of devices for the same task to increase throughput and backup devices in case the primary devices are in an error state. • *Advanced Settings*. Allows you to add JavaScript to change the task parameters or pass information to and from an external database during a protocol run.

Related information

For information about...	See...
VWorks software description	"VWorks software description" on page 2
Quick reference of menu and toolbar commands	"Quick reference" on page 663
Supported devices	"Supported devices" on page 3
Agilent Technologies devices	Agilent Technologies device user guide

For information about...	See...
Third-party devices	Third-party device user guide and the corresponding device driver user guide
VWorks software installation and setup	*VWorks Automation Control Setup Guide*
Defining labware and creating liquid classes	*VWorks Automation Control Setup Guide*
Managing and tracking labware in storage	*VWorks Automation Control Setup Guide*
Managing user accounts	*VWorks Automation Control Setup Guide*
Using VWorks ActiveX control	*VWorks Automation Control Setup Guide*

2
Creating a protocol: basic procedure

Read this chapter if you are an administrator or technician who writes protocols.

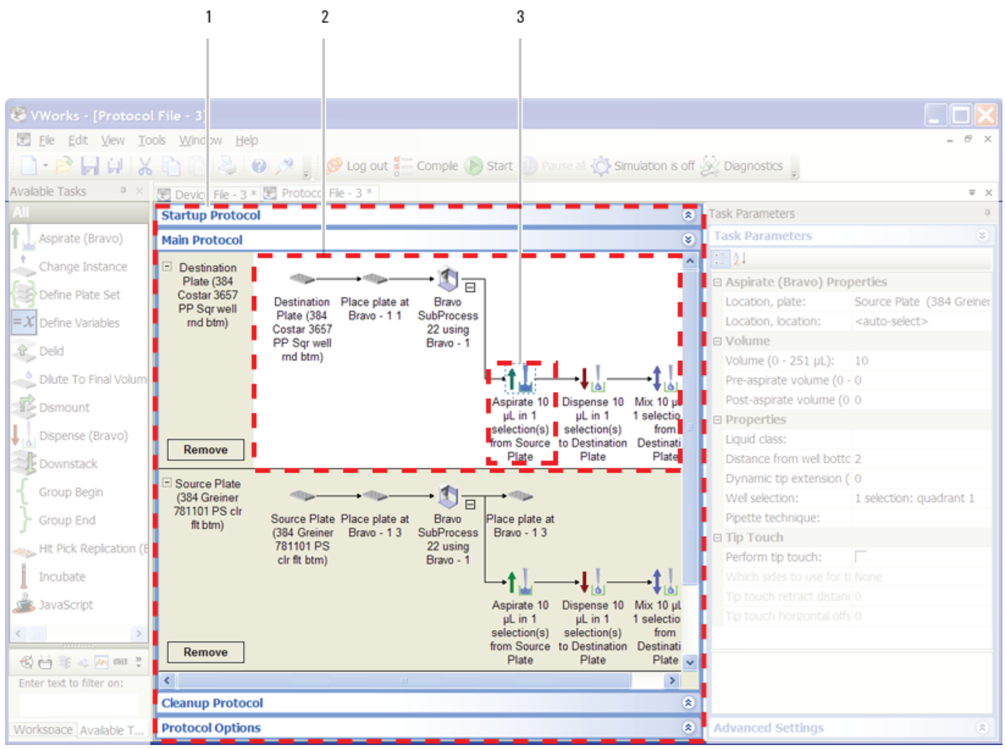
This chapter contains the following topics:

For details on using macros to help create your protocols, see "Using macros to create protocols" on page 133.

About protocols, processes, and tasks

Concept overview

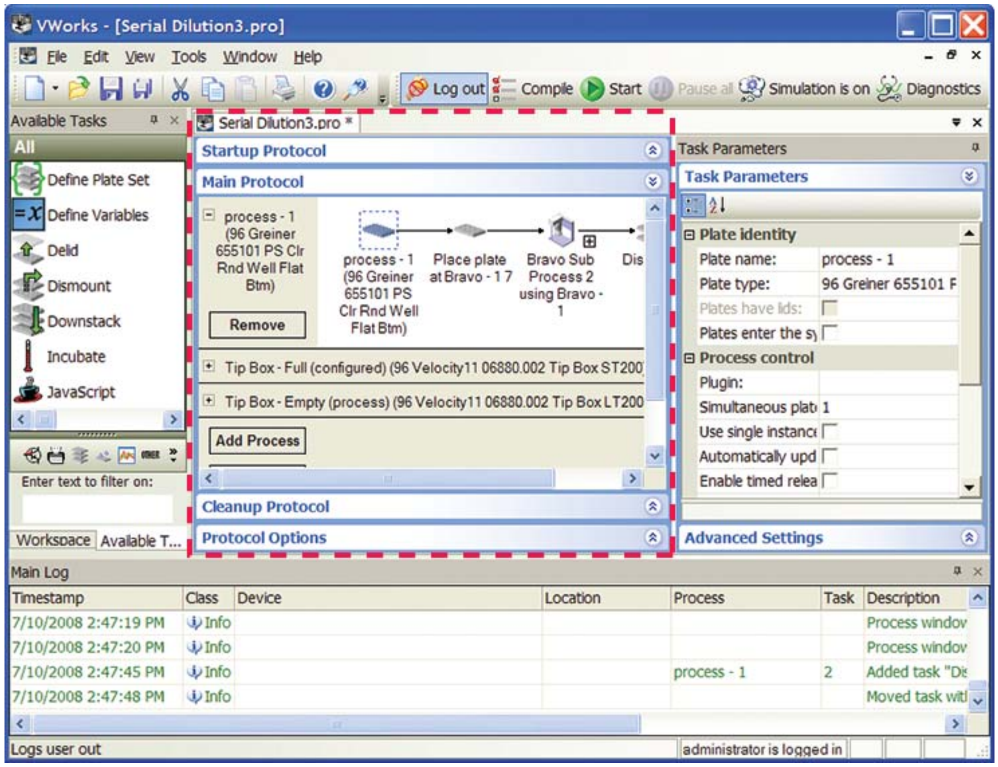
The VWorks software enables you to create protocols to specify the laboratory tasks to automate. A protocol (1) consists of one or more processes (2). Each process consists of one or more tasks (3).



Protocols

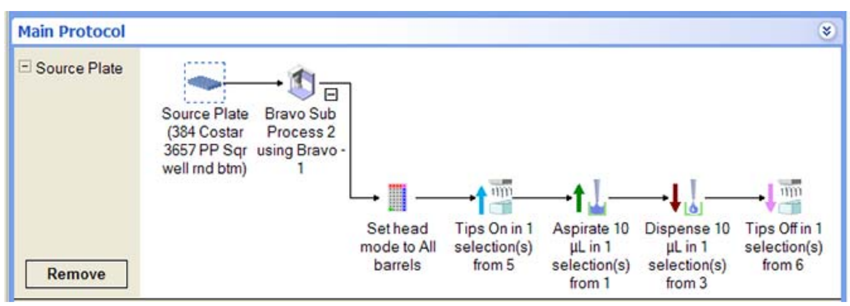
A protocol is a schedule of tasks to be performed by a standalone device or devices integrated in the lab automation system. The purpose of a protocol is to process or perform tasks on labware.

Protocols appear in the Protocol area of the VWorks window. The following example shows a protocol called Serial Dilution3 displayed in the VWorks window.



Processes

A process is a sequence of tasks that are performed on a particular labware or a group of labware. A process is represented by a lane with white background in the Protocol area.



Each process lane starts with a microplate icon. The icon typically represents the labware or group of labware that you are processing. The labware is called the process plate.

In the previous example, the process plate is called Source Plate and is associated with a specific labware type (the type is defined in the Labware Editor). The process contains a sequence of tasks for processing the Source Plate: Set head mode, Tips On, Aspirate, Dispense, and Tips Off. (For a description of the tasks, see "Setting parameters for liquid-handling tasks" on page 387.)

A process lane can also be used as a control to initiate other processes in the protocol. In this case, the microplate icon at the beginning of the process lane is not associated with any labware.

In the following example, the process lane named Control is used to initiate other processes in the protocol. Notice that the protocol can contain more than one process.

The example also shows a lane with gray background. The gray lane displays activities of configured labware. For more information about configured labware, see “Configuring labware” on page 40.

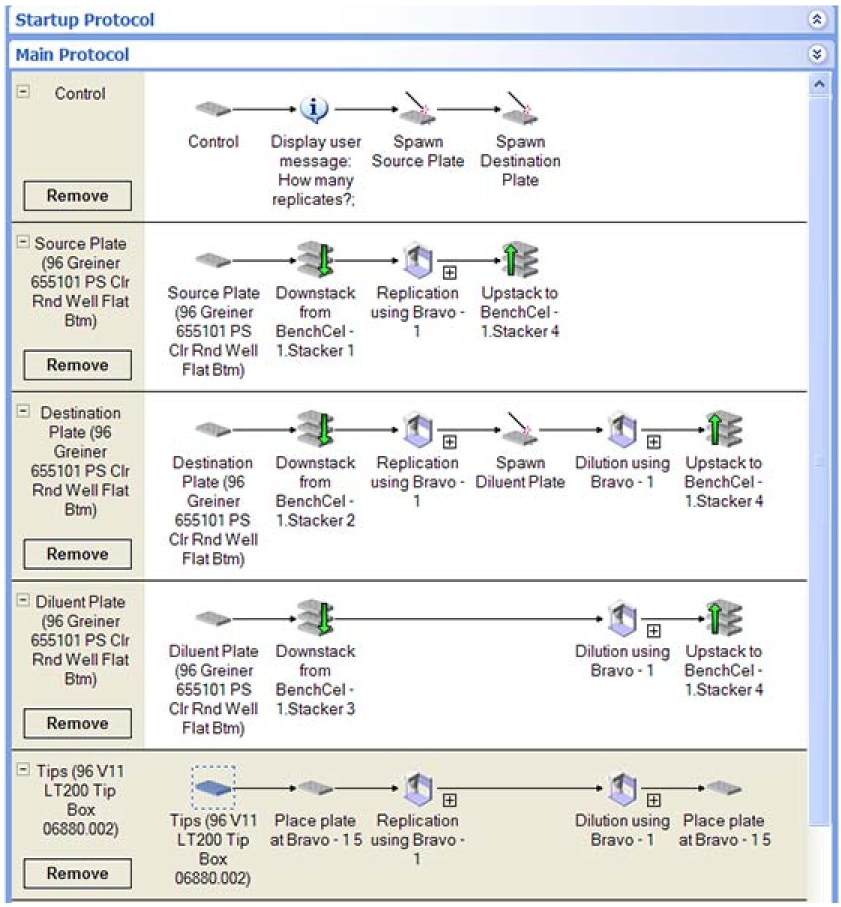


Plate instances

When a process plate icon represents a group of labware, each labware in the group is called a plate instance. Using the previous example, if you have 10 microplates that need to be processed as the Source Plate, then microplate 1 is Source Plate instance 1, microplate 2 is Source Plate instance 2, microplate 3 is Source Plate instance 3, and so on.

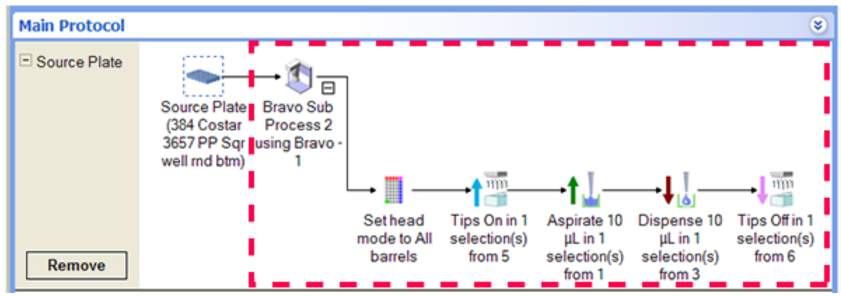
IMPORTANT All labware represented by a process plate must be the same labware type.

Subprocesses

A subprocess is a sequence of tasks performed as a subroutine within a protocol. A subprocess is performed by a single device type, such as the Bravo device.

IMPORTANT Within a given protocol, ensure that any main process and subprocess do not share the same name.

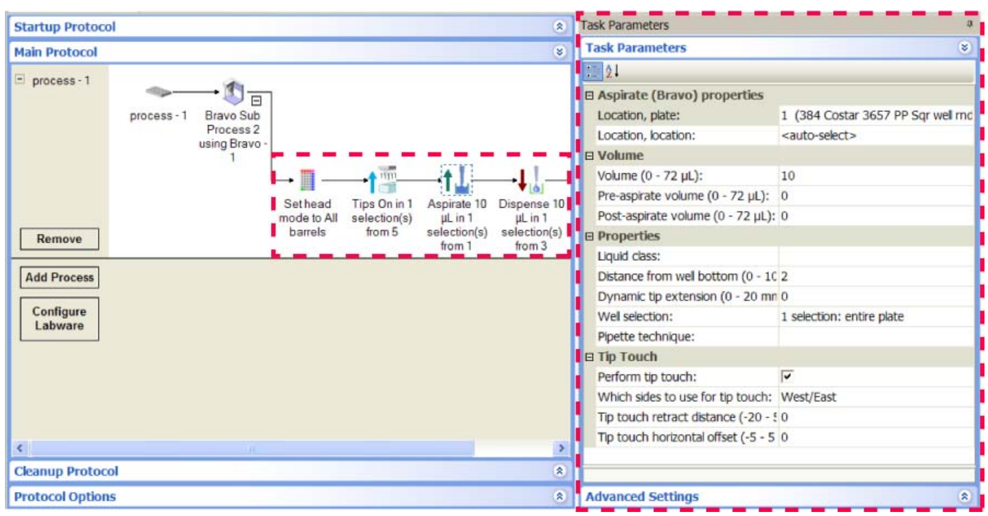
Subprocesses are represented by a subprocess icon in the protocol. You can expand or collapse the subprocess to show or hide the subprocess tasks. In the following example, the Bravo Subprocess is expanded to show the following tasks: Set Head Mode, Tips On, Aspirate, Dispense, and Tips Off. (For a description of the tasks, see "Setting parameters for liquid-handling tasks" on page 387.)



Tasks

A task is an operation performed on one or more labware, and is represented by an icon in the protocol. It has associated parameters that are set in the Task Parameters area.

In the following example, the tasks in the Bravo subprocess are highlighted. Notice that the Aspirate task is selected, and the parameters for the Aspirate task are displayed in the Task Parameters area on the right.



Related information

For information about...	See...
Defining labware	*VWorks Automation Control Setup Guide*

For information about...	See...
Preparing for protocol writing	“Preparing for protocol writing” on page 19
Workflow for creating basic protocol	“Workflow for creating a basic protocol” on page 18
Configuring labware	“Configuring labware” on page 40
Using advanced features	“Creating a protocol: advanced topics” on page 73

Workflow for creating a basic protocol

The following table presents the workflow for creating a protocol.

Step	For this task...	See...
1	Prepare for protocol writing.	“Preparing for protocol writing” on page 19
2	Log in to the VWorks software.	“Logging in, logging out, and changing passwords” on page 22
3	Add devices and create a profile for each device.	“Adding devices” on page 25
4	Create a protocol.	“Creating a new protocol” on page 29
5	Set protocol options.	“Setting protocol options” on page 30
6	Add an alarm.	“Adding an alarm” on page 37
7	Configure labware.	“Configuring labware” on page 40
8	Adding processes.	“Adding processes” on page 44
9	Set plate properties.	“Setting plate parameters” on page 46
10	Add tasks.	“Adding and deleting tasks” on page 53 To use macros for commonly used task sequences, see “About protocol macros and the macro library” on page 134.
11	*Optional.* Set up startup and cleanup protocols.	“Setting up Startup and Cleanup Protocol processes” on page 60
12	Save the protocol.	“Saving the protocol” on page 61

Step	For this task...	See...
13	Compile the protocol.	"Compiling the protocol" on page 63
14	Simulate the protocol run.	"Simulating the protocol run" on page 64
15	*Optional.* Print the protocol.	"Printing protocols" on page 70

Related information

For information about...	See...
Creating a simplified interface for operators to use	"Creating protocol forms for operators" on page 153
Protocols, processes, and tasks	"About protocols, processes, and tasks" on page 14
Using advanced features	"Creating a protocol: advanced topics" on page 73

Preparing for protocol writing

Before you create a protocol, determine the following:

- The devices and accessories you need for the protocol.

  A device is a robot, instrument, or location in the lab automation system that can hold a piece of labware. For more information, see "Adding devices" on page 25.

  An accessory is an option that can be added to a robot, instrument, or location to enhance existing functions and facilitate operation (for example, an Auto Filling Reservoir).
- The labware that will be used or processed during the protocol run and their starting and ending locations.
- Whether the macro library contains any task sequences that you can use. For details on using macros, see "Using macros to create protocols" on page 133.

This topic explains how you plan for and specify the different devices, accessories, and labware in a protocol and the terminology that is used.

Planning device and accessory use

When planning for devices and accessories:

- Determine the devices and accessories you will need in the protocol.

- Determine the locations of the accessories on devices such as the Bravo Platform or the Vertical Pipetting Station. You do this when you configure the accessory in the diagnostics software. For instructions, see the device user guide.

IMPORTANT Accessory locations are displayed in the diagnostics Configuration tab only. You need to remember their locations when configuring labware in the protocol.

Note: Devices and accessories stay at the same location throughout a protocol run.

For the list of available devices and accessories you can use in the lab automation system, see the device user guide or the Agilent Technologies website at www.agilent.com/lifesciences/automation.

Planning labware use

In a lab automation system, labware can either:

- Transfer into and out of the system for processing.
- Start and stay in the system during the entire protocol run.

When you plan a protocol, you should determine how the labware will be used or processed and how they will move in the system. For example, if you are writing a microplate replication protocol, you need to decide whether the source microplates or destination microplates will be moved into the system while the other will remain stationary in the system. The decision can depend on many factors, including your preferences.

In the VWorks software, labware can be categorized as follows:

- Process plates
- Configured labware
- Static labware

Process plates

A process plate is a labware that:

- Is transferred into the system automatically or manually during the protocol run.
- Is the object of one or more tasks in the protocol run.
- Might move to different locations during the run.
- Is transferred out of the system automatically or manually during the protocol run.

For more information about process plates, see “About protocols, processes, and tasks” on page 14. To create a protocol process, see “Workflow for creating a basic protocol” on page 18.

Configured labware

A configured labware is a labware that:

- Starts at a location on a device.
- Is used by one or more tasks in the protocol process.
- Might move to different locations during the run.
- Returns to the original location after the protocol run is finished.

For the Bravo Platform, configured labware represents the single physical labware on the deck, such as a tip box. For example, if you are using two different tip boxes in a protocol, you would configure two labware in the software, one for each tip box on the deck.

Like accessories, you must let the software know the labware's starting location. To do this, see "Configuring labware" on page 40.

Configured labware is displayed in a protocol with a gray background. If it is used by a task in a Bravo sub-process, a copy of the sub-process tasks are shown next to the configured labware. You cannot add or remove tasks in the duplicated process. However, whenever the sub-process is updated, the duplicate copy is also updated automatically.

You have the option of converting a configured labware into a process plate. For instructions, see "Configuring labware" on page 40.

Static labware

IMPORTANT The latest version of the VWorks software is backward-compatible with Bravo Platform protocols created in VWorks4 version 6.2.3 or earlier and will continue to support static labware configuration procedures. However, Agilent Technologies recommends that you use the configured labware when writing new protocols.

A static labware is a labware that will start on the Bravo deck and will remain at the same location during the protocol run. For example, a tip box can be a static labware.

To specify its starting location, you must configure the static labware using one of the following methods:

- The Bravo Sub-Process task in a Main Protocol
- The Configure Static Labware task in the Startup Protocol
- In general, you configure static labware before the first task in a protocol. If you have multiple processes in the protocol, configure the labware once before the first task of the first process.

Configure static labware in a Startup Protocol if it will be used in all the Main Protocol subprocesses. Configure static labware in the Main Protocol subprocess if you want to override the labware configuration in the Startup Protocol.

Related information

For information about...	See...
Protocols, processes, and tasks	"About protocols, processes, and tasks" on page 14
Workflow for creating a basic protocol	"Workflow for creating a basic protocol" on page 18
Using advanced features	"Creating a protocol: advanced topics" on page 73
Using macros to expedite protocol writing	"Using macros to create protocols" on page 133
Creating a custom interface for the operator who runs the protocol	"Creating protocol forms for operators" on page 153

Logging in, logging out, and changing passwords

About this topic

To create, modify, or run a protocol, you must first log in. Contact your lab manager or administrator to set up a user account or to find out your access privileges. If you are an administrator, see the *VWorks Automation Control Setup Guide* for instructions on setting up user accounts.

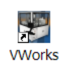
This topic explains the following:

- “Logging in” on page 22
- “Logging out” on page 24
- “Changing passwords” on page 24

Logging in

To log in VWorks software:

1 Start VWorks software. To do this, double-click the VWorks icon on the Windows desktop.

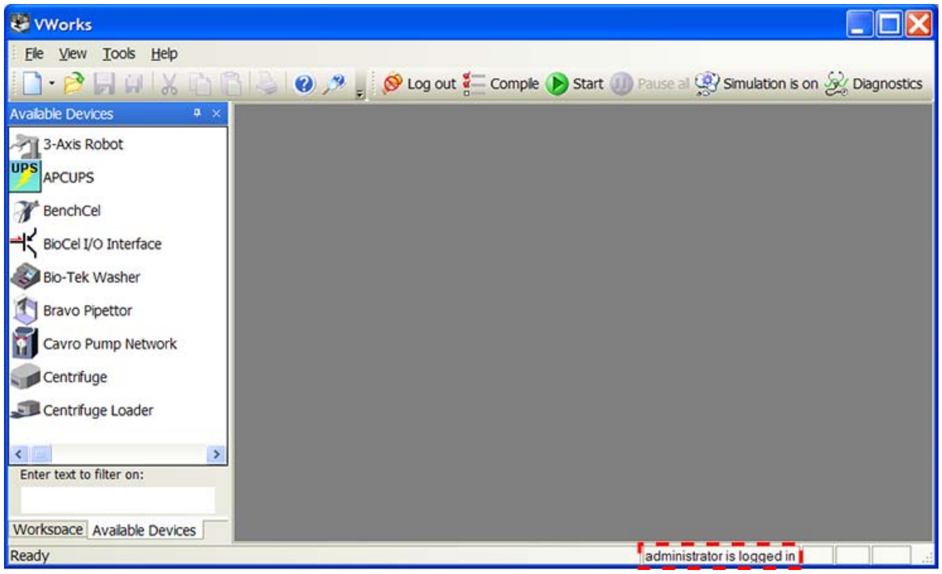


2 In the VWorks window that opens, click **Log in** on the toolbar.

The User Authentication dialog box opens.

3 Type your VWorks user name and password, and then click **OK.** (If no user account is set up, contact the administrator.)

In the VWorks window, the Log in button changes to Log out. In addition, the status bar indicates that the login is successful.


VWorks
File View Tools Help
Log out Compile Start Pause all Simulation is on Diagnostics
Available Devices
3-Axis Robot
APCUPS
BenchCel
BioCel I/O Interface
Bio-Tek Washer
Bravo Pipettor
Cavro Pump Network
Centrifuge
Centrifuge Loader
Enter text to filter on:
Workspace Available Devices
Ready
administrator is logged in

Logging out

To log out of VWorks software:

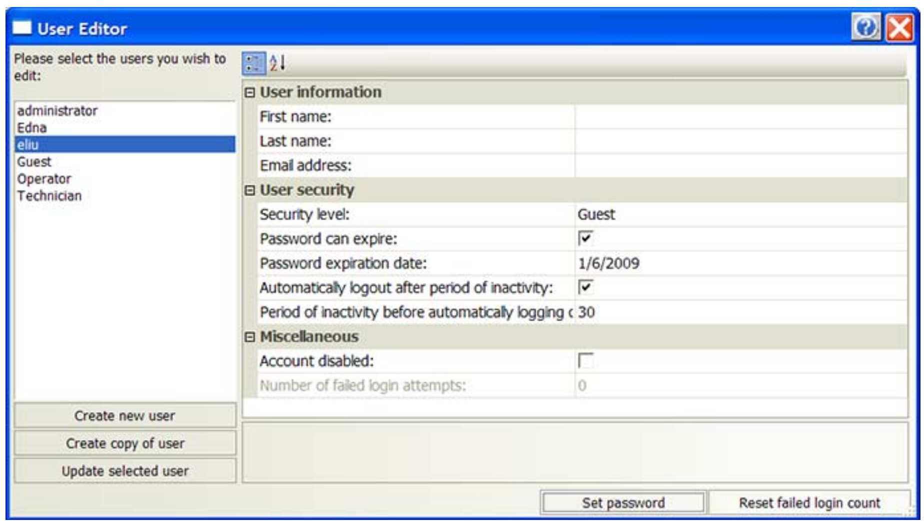
In the **VWorks** window, click **Log out** on the toolbar.

Changing passwords

You can change your password if you have administrator or technician privileges. If you have an operator or guest user account, you must contact the administrator to change your password.

To change your password:

1 In the **VWorks** window, select **Tools > User Management.** If you have administrator privileges, the User Editor dialog box opens.



If you have technician privileges, the User Management dialog box opens.

2 Click **Set Password.** If you have technician privileges, click **Change Password.** The Set Password or Change Password dialog box opens.

3 Type the new password in the **New password** and **Confirm new password** boxes, and then click **OK.** A message appears and lets you know that the password was successfully changed.

Related information

For information about...	See...
Setting up and managing user accounts	*VWorks Automation Control Setup Guide*
Workflow for running a protocol	"Workflow for running a protocol" on page 210
Creating a new protocol	"Creating a protocol: basic procedure" on page 13

Adding devices

About devices

In the VWorks software, a device is an item in your lab automation system that can be added to the VWorks device file. A device can be a robot, an instrument, or a location on the lab automation system that can hold a piece of labware. The following are examples of devices:

- The robot in the BioCel System
- PlateLoc Sealer
- Microplate Labeler
- Labware Stacker
- Platepad
- A third-party device integrated in the lab automation system

About device files

To communicate with and to control the robot and integrated devices, the VWorks software uses a device file that contains the following information:

- List of devices the software will communicate with and control

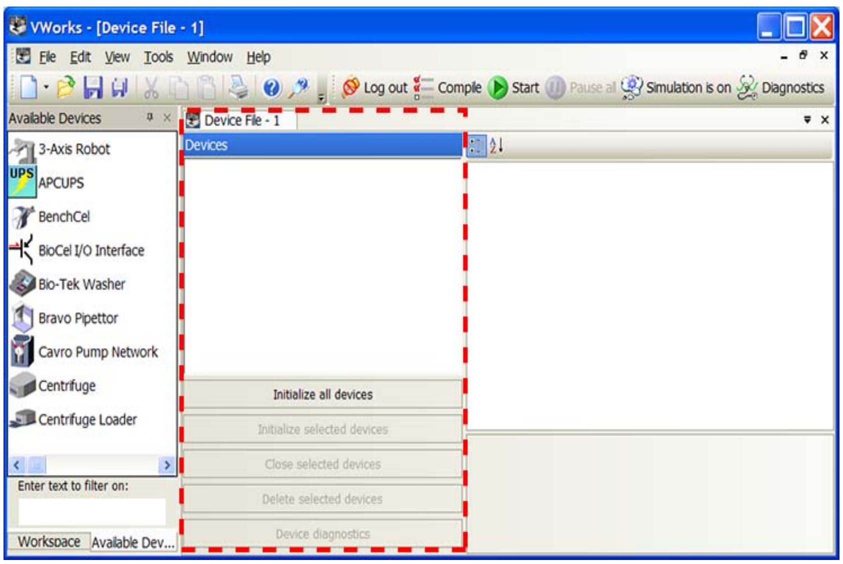
- Device type of each device (for example, the robot in the lab automation system, PlateLoc Sealer, and any integrated device)
- Configuration information of each device (for example, approach height, allowed or prohibited labware, barcode reader access, and so on)
- The communication settings (profile) needed for communication between the devices and the VWorks software

You provide the device information in the VWorks window. The device information is stored in a device (.dev) file that is located in a folder you specify when saving the file.

Creating a device file

To create a device file:

1. In the VWorks window, select **File > New > Device.** A Device File tab appears.
2. Select **File > Save** to save the device file. The file name appears in the Device File tab.



Adding devices to the device file

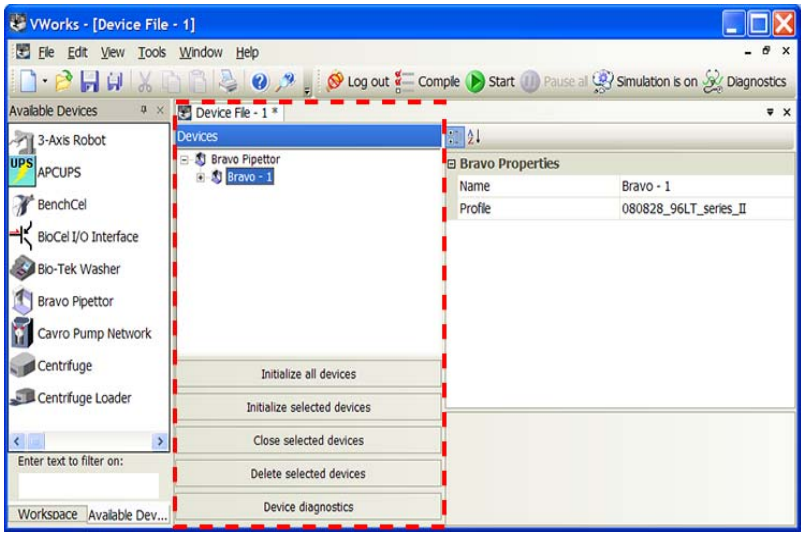
To add a device to the device file:

1. In the **Available Devices** area, double-click the device that you want to add. Alternatively, you can drag a device from the **Available Devices** area into the **Device File** area.

   If you do not see the device in the **Available Devices** list, check that the device plugin file is stored in the ...\Agilent Technologies\VWorks\Plugins folder.

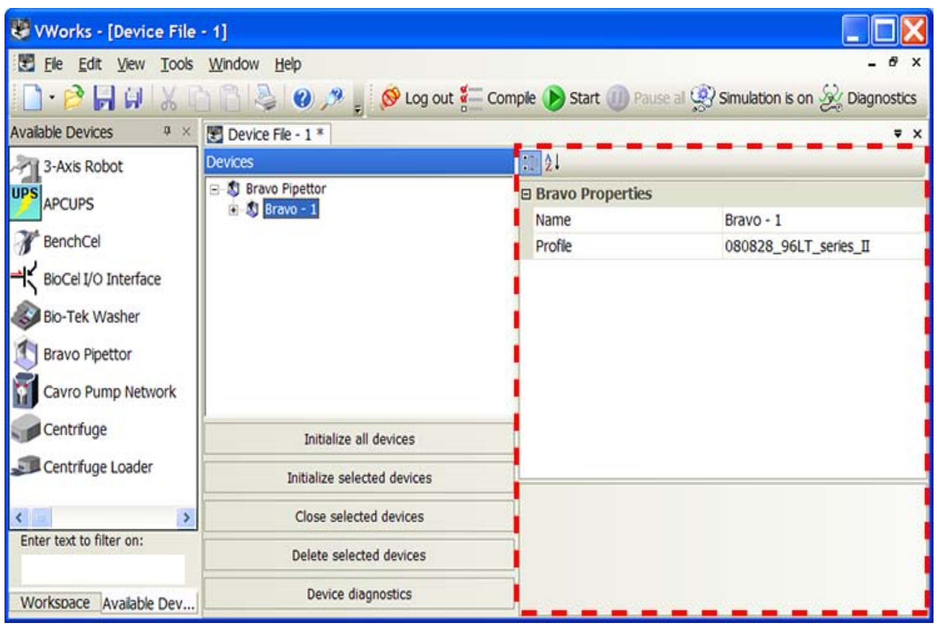
   If you added a device plugin file in the Plugins folder and you have already started the VWorks software, be sure to reload the plugin. To do this, close any open device files and protocol files, and then select **Tools > Reload Plugins.**

   In the following example, the Bravo Pipettor device type is added. Notice that the first Bravo device is labeled Bravo-1. If you add another Bravo device, it will appear as Bravo-2 under the Bravo Pipettor.



2 Type a name for the device and set the device properties. For detailed description of the properties, see the device user guide.

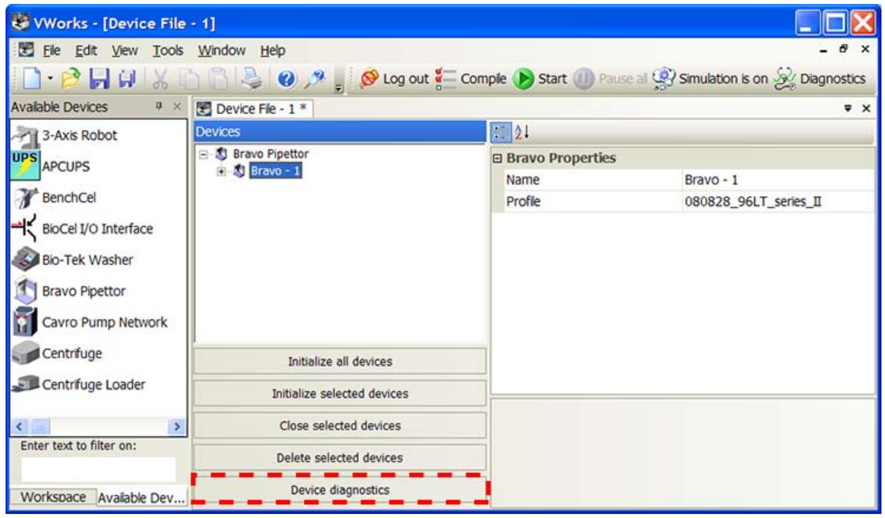
In the following example, Bravo-1 is the default name of the Bravo Pipettor device. The only property shown is the profile selection.



3 Create a profile for the device:

a Select the device in the **Devices** list.

b Click **Device diagnostics.**



c In the device diagnostics dialog box, name the profile, select the connection type (Ethernet or serial), and locate and connect to the device in the Discovered Bionet Devices dialog box (Ethernet connections only).

d For devices such as the Bravo Platform and the Vertical Pipetting Station, set the teachpoints.

For detailed instructions on creating device profiles and setting teachpoints, see the device user guide.

4 Select the profile in the device properties area.

5 Select **File > Save** to save the device file. The file name appears in the Device File tab.

6 Repeat steps 1 through 5 to add other devices.

7 In the **Device File** area, click **Initialize all devices** to establish communication with the devices.

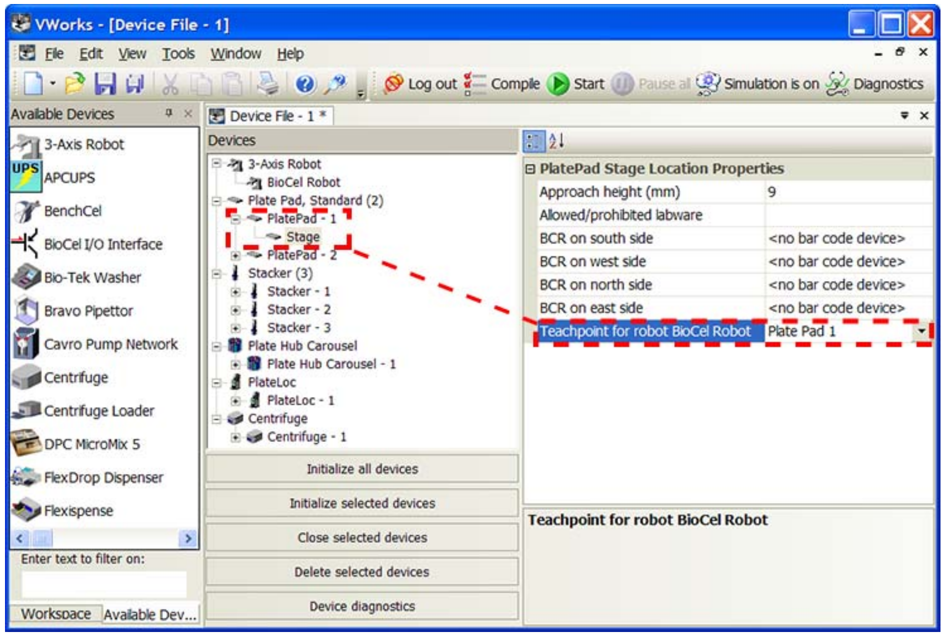
8 If you are adding devices to the BenchCel Workstation or the BioCel System, enable the robot to move to the correct locations during a protocol run as follows:

a In the BenchCel or the BioCel robot diagnostics dialog box, set teachpoints at each device or location. Save and reference the teachpoint file in the diagnostics dialog box.

For detailed instructions, see the *BenchCel Microplate Handling Workstation User Guide* or the *BioCel System User Guide*.

b In the VWorks window, select each of the devices in the device file, and select the correct robot teachpoint in the device properties area.

The following example shows a device file for a BioCel System. To ensure that the BioCel robot will move correctly to and from Platepad - 1, **Platepad - 1 Stage** is selected in the **Device File** area, and the defined teachpoint called Plate Pad 1 is selected in the **PlatePad Stage Location Properties** area.



9 Select **File > Save** to save the changes.

10 In the **Device File** area, click **Initialize all devices.**

Disabling devices in a device file

You can temporarily disable a device in a device file to run protocols using only a subset of devices or during troubleshooting.

For more information see, "Disabling and enabling a device in the device file" on page 215.

Related information

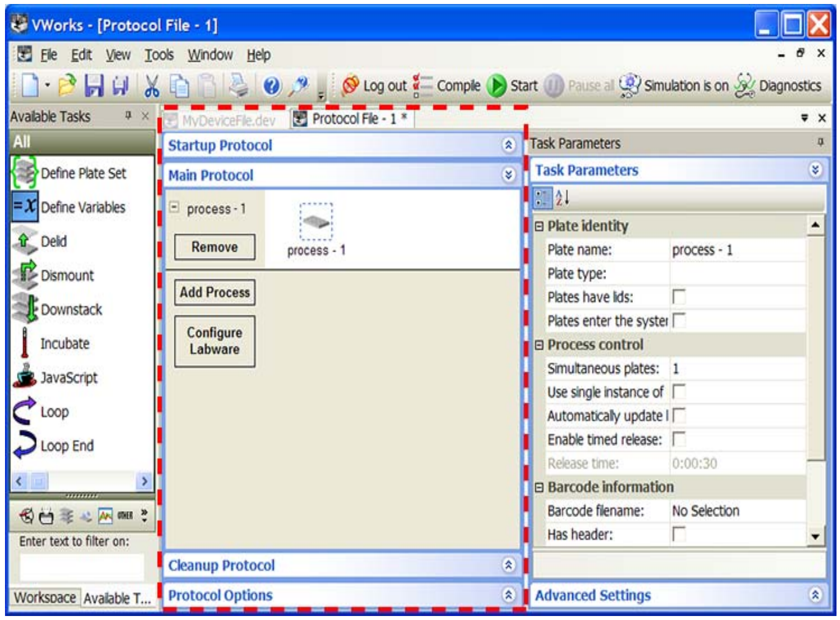
For information about...	See...
Device-specific properties	Device user guide
Creating a new protocol	"Workflow for creating a basic protocol" on page 18

Creating a new protocol

Procedure

To create a new protocol:

Select **File > New > Protocol.** A Protocol File tab appears.



Related information

For information about...	See...
Setting protocol options	"Setting protocol options" on page 30
Adding tasks	"Adding and deleting tasks" on page 53
Setting up startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60
Using advanced protocol features	"Creating a protocol: advanced topics" on page 73

Setting protocol options

When you create a protocol, you need to specify which device file to use with the protocol. Optionally, you can specify additional information to associate with your protocol.

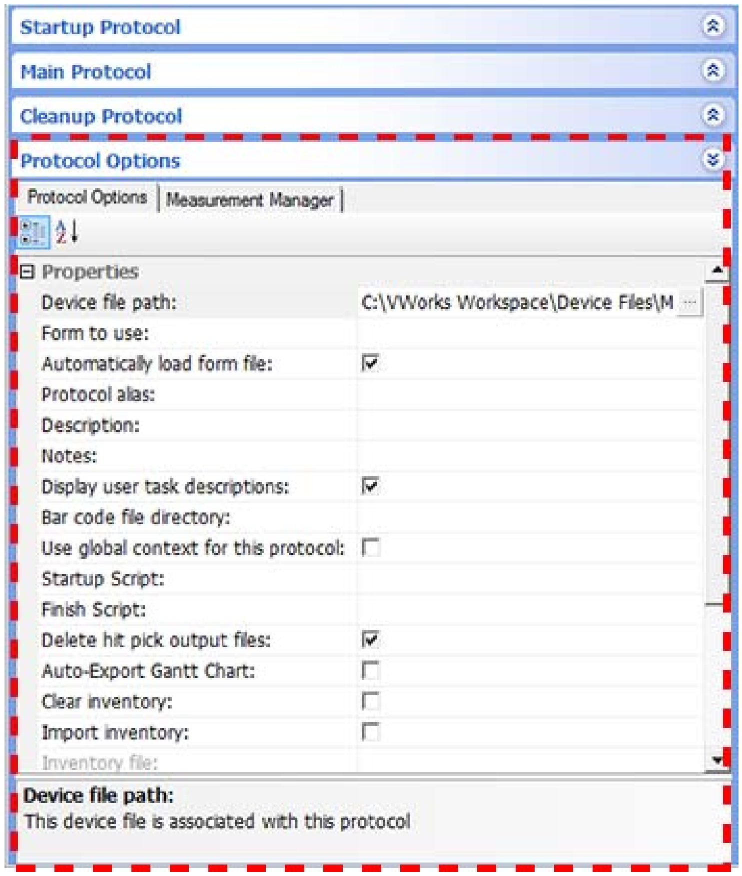
This topic explains the following:

- "Specifying the device file for a protocol" on page 30
- "Adding information about the protocol" on page 31
- "Specifying protocol rules" on page 34

Specifying the device file for a protocol

To specify the device file:

1 In the Protocol area, click **Protocol Options** to view the properties.



2 Click the field adjacent to **Device file path**, and then click the [...] button that appears. The Open dialog box opens.

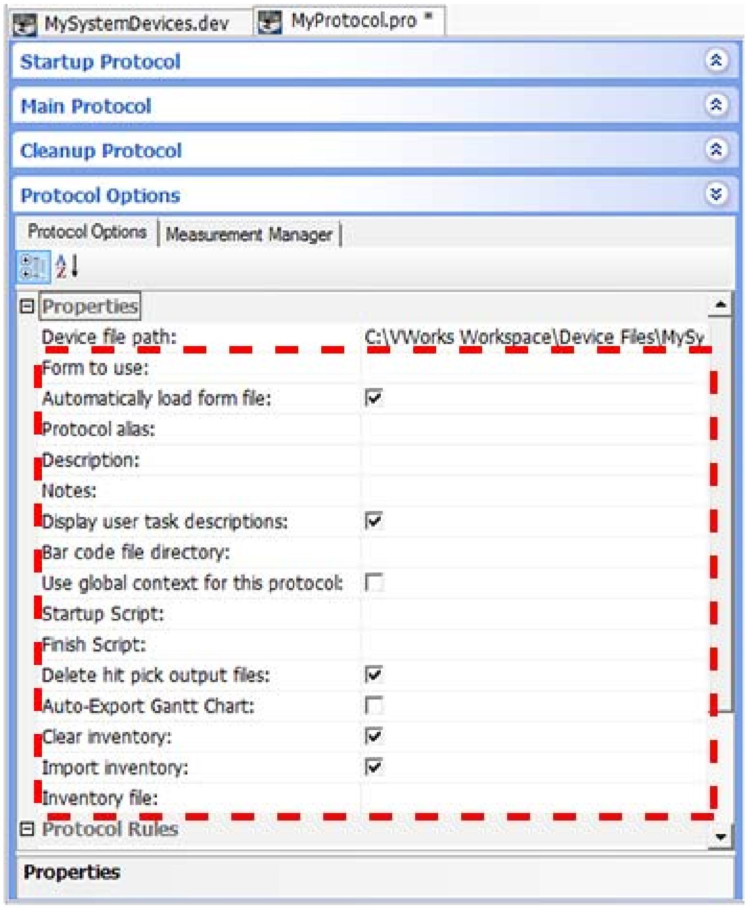
3 Locate and select the correct device (.dev) file, and then click **Open.** The path of the device file appears in the Device file path box.

Adding information about the protocol

You can add information about your protocol. For example, you can provide a description and some notes about the protocol.

To add information about the protocol:

In the **Protocol Options** area, type the following information:



Optional protocol information	Description
Form to use	The option that specifies a VWorks form for operators to use as the graphical interface to run a protocol. For details on how to create forms and specify which form to use, see "Creating protocol forms for operators" on page 153.
Automatically load form file	The option that opens the form that is specified in the Form to use field anytime the corresponding protocol is opened.
Protocol alias	Another name for the protocol. The alias is displayed in the software, but the protocol file name is not changed.
Description	A brief description of the protocol.
Notes	Special notes about the protocol.
Display user task descriptions	The option to display the task description (or task label) that you provide instead of the default task label. The label is displayed under the task icon in the protocol.

Optional protocol information	Description
Bar code file directory	The location of the files that contain the barcodes you want to track in the software. The list of files in this folder will be available in the Plate Properties area. For details, see "Setting plate parameters" on page 46 and "Tracking barcodes" on page 74.
Use global context for this protocol	The option that permits variables to be available across all protocols that also use the global context for variables. **CAUTION** Variables with the same name in other protocols, which also use the global context, will interfere with each other. Always make sure the variable values you want to use globally are applicable to the other protocols that use the global context.
Startup script	The JavaScript code you want to run before the Startup or Main Protocol begins.
Finish script	The JavaScript code you want to run after the Main or Cleanup Protocol finishes.
Delete hit pick output files	The option to delete hit-pick output files after the protocol run finishes.
Auto-export Gantt Chart	The option to automatically export the Gantt Chart after the protocol run is finished. The exported file is stored in C:\VWorks Workspace\VWorks\Logs\.
Clear inventory	The option to automatically clear the labware information from the inventory database at the start of the protocol run. *Note:* Plate and location group information are retained.
Import inventory	The option to automatically import and use the specified inventory file when starting the protocol run. You specify the inventory file to import in the **Inventory file** box.

Optional protocol information	Description
Inventory file	The inventory file you want to import and use when starting the protocol run. **1** Click the **Inventory file** box, and then click the browse button that appears. **2** Locate and select the desired inventory (SQL) file. *Note:* This box is only active if you selected **Import inventory**. **IMPORTANT** Make sure you import the most current inventory file. Importing an out-of-date inventory file can cause protocol runs to abort.

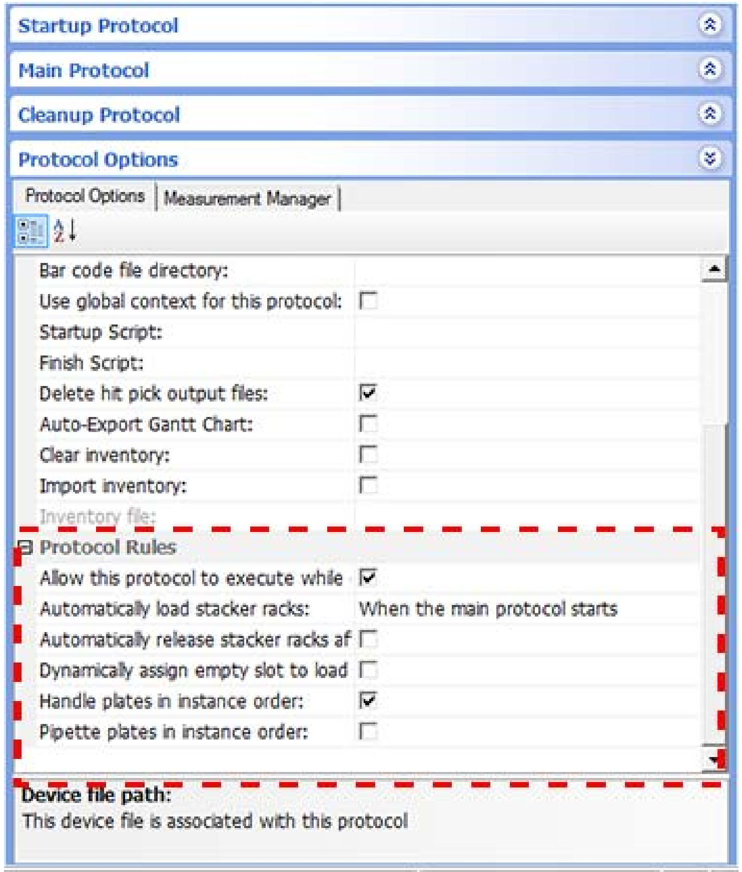
Specifying protocol rules

You can specify certain actions to occur before or after the protocol run.

To specify the protocol rules:

In the **Protocol Options** area, specify the following rules:

- Allow this protocol to execute while other protocols are running
- Automatically load stacker racks
- Automatically release stacker racks
- Dynamically assign empty slot to load to storage device
- Handle plate in instance order
- Pipette plates in instance order



Allow this protocol to execute while other protocols are running

This rule allows you to specify the following:

- *Select the check box (default).* Enables the Runset Manager to run the protocol simultaneously with another protocol.

  IMPORTANT Simultaneously running protocols must specify the same static or configured labware.

- *Clear the check box.* Prevents the protocol from running simultaneously with another protocol.

Automatically load stacker racks

Select the **Automatically load stacker racks** rule to require that all racks on stacking devices (such as the BenchCel Microplate Handler and the Labware Stacker) be automatically loaded before either one of the following starts:

- Startup Protocol
- Main Protocol

Automatically release stacker racks

Select the **Automatically release stacker racks** rule to require that all racks on stacking devices (such as the BenchCel Microplate Handler and the Labware Stacker) be automatically released after the protocol finishes.

Dynamically assign empty slot to load to storage device

Select the rule to have the software assign slots in the storage device according to what is available at that moment. Under this rule, labware can be placed in different locations (not grouped together).

Clear the check box if you want to store labware consecutively in the same carousel or stacker.

Handle plate in instance order

Select the **Handle plate in instance order** rule to require that microplates be processed in the order as they enter the system.

For example:

Two PlateLoc Sealers are used in a BioCel System to seal microplates. One of the PlateLoc Sealers runs out of seal material and stops on a microplate.

If this rule is selected, the second sealer would also stop until you load a new roll of seal, start the first sealer, and the microplate at the first sealer is upstacked to its position in the expected order.

If this rule is not selected, the second sealer would continue sealing and upstacking microplates. The microplate at the stopped sealer would be omitted, thus the sequence of microplates will be out of order.

Pipette plates in instance order

Select the **Pipette plate in instance order** rule to require that microplates be processed at pipetting devices in the order as they enter the system.

Select the check box if time-sensitive assays require that each microplate be processed the same way within the same length of time. Doing so ensures the data are comparable across the microplates.

Clear the check box if:

- The protocol has more than one pipette process that uses the same liquid-handling device.
- The duration of one of the pipetting operations is much longer than another.

For example:

A protocol has two processes and both have a pipetting operation that uses the same Vertical Pipetting Station.

If the rule is selected, the pipetting operations for one process are completed before the pipetting operations begin for the other process. The first pipetting operation takes significantly longer to complete than the second operation, and the protocol is run several times in succession. The overall time taken for the protocol to complete is much greater than it needs to be because during each cycle the system had to wait for the slower pipetting operations to complete for all the microplates in the process before it could continue.

If the rule is not selected, a microplate from the fast pipetting process can be delivered to the Vertical Pipetting Station after a microplate from the slow pipetting process, followed by another microplate from the slow pipette process, and so on. This reduces the bottleneck at the Vertical Pipetting Station because it allows the faster process to continue, and its second cycle in the series to start before the first cycle is complete.

Related information

For information about...	See...
Specifying a device file for the protocol	“Specifying the device file for a protocol” on page 30
Adding other information about the protocol	“Adding information about the protocol” on page 31
Adding an alarm	“Adding an alarm” on page 37

Adding an alarm

You can set an alarm to create an error message when a measurement falls outside the range that you specify. For example, you can specify that the alarm creates an error message when:

- The bottle on the Weigh Pad becomes too heavy or too light.
- A particular temperature, humidity, or gas concentration level is reached.
- A system door is open.
- The battery power in the uninterruptible power supply (UPS) is low.

Note: The alarm can only be used if your lab automation system is equipped with doors, a Weigh Pad, a StoreX incubator with environmental control, an iSeries controller in a BioCel System, or a UPS.

IMPORTANT You can add User Message tasks in the protocol to remind the operator to empty or fill containers or reservoirs at the appropriate steps in the protocol. For instructions on adding User Message tasks, see “User Message” on page 599.

Procedure

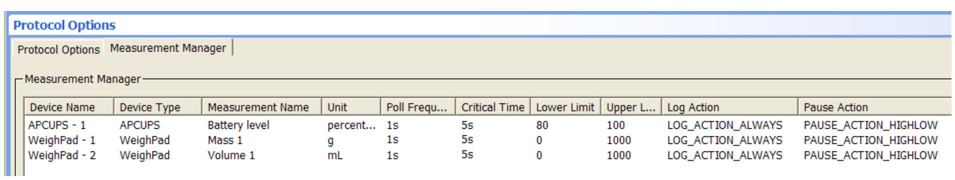
IMPORTANT The alarm settings apply only to the current protocol and do not impact other protocols.

To set up alarms:

1 In the **Protocol** area, click **Protocol Options.**

2 Click the **Measurement Manager** tab and specify the alarm parameters.

The Measurement Manager tab displays all the measurements for the devices you want to monitor (for example, the Weigh Pad). If you do not see the information in the tab, check that you have added the devices you want to monitor in the device file. See “Adding devices” on page 25 for instructions.



Parameter	Description
Device Name	The name of the device for which the alarm is set. If you have two devices of the same type, each device is distinguished by its device name. In the example shown, two Weigh Pads are listed: Weigh Pad - 1 and Weigh Pad - 2.
Device Type	The type of device. The device type can be the Weigh Pad, the UPS (ACPUPS), StoreX incubator, Cytomat incubator, or the iSeries controller.
Measurement Name	The measurement you want to monitor. For example, you monitor the battery level in the UPS. The measurement is device dependent. In the example shown, the two Weigh Pads have different measurement names.
Unit	The unit of measure. For example, if you are monitoring the battery, the unit of measure is percent (%). If you are monitoring the mass on the Weigh Pad, the unit of measure is grams (g).
Poll Frequency	The frequency at which the software takes the measurements. For example, 1 s means the software will check the device every second and display the measurement in the log.
Critical Time	The length of time the measurement is allowed to be above the upper limit or below the lower limit before it is considered to be out of range. For example, the upper limit of a measurement is 5 volts and you specify a 2 s critical time. When the software takes a measurement, the reading fluctuates and is at 6 volts for longer than 2 s overall. So it is considered to be out of range.
Lower Limit	The low value at which the alarm is turned on.
Upper Limit	The high value at which the alarm is turned on.

Parameter	Description
Log Action	The action to record the error: • *LOG_ACTION_ALWAYS*. Records the measurements in the log. For example, if the Poll Frequency is 1 s, the measurement for every second is recorded in the log. • *LOG_ACTION_NONE*. Does not record the measurement in the log. • *LOG_ACTION_LOW*. Records the measurement when it exceeds the lower limit. • *LOG_ACTION_HIGH*. Records the measurement when it exceeds the upper limit. • *LOG_ACTION_HIGHLOW*. Records the measurement when it exceeds the upper or lower limit.
Pause Action	The action to pause the run: • *PAUSE_ACTION_NONE*. Does not pause the run when the measurement is out of range. • *PAUSE_ACTION_LOW*. Pauses the run when the measurement exceeds the lower limit. • *PAUSE_ACTION_HIGH*. Pauses the run when the measurement exceeds the upper limit. • *PAUSE_ACTION_HIGHLOW*. Pauses the run when the measurement exceeds the upper or lower limit.

3 When you are finished, click **Apply.**

Related information

For information about...	See...
BioCel I/O Interface	*BioCel System User Guide*
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Adding User Message tasks	"User Message" on page 599
Protocol options	"Setting protocol options" on page 30

Configuring labware

About this topic

After you create a new protocol, you can configure labware for devices that have locations for labware placement. When labware is configured at a device location, the labware might be moved from its location for deadlock avoidance or be used as a counterweight, but returned to the same location by the end of the run.

For example, you can configure labware that will be stored at Bravo deck location 2. During the protocol, the labware will be moved to deck location 5 temporarily for processing, and then returned to deck location 2.

This topic explains how to configure labware. For more information about configured labware and how it is used in a protocol, see “Planning labware use” on page 20.

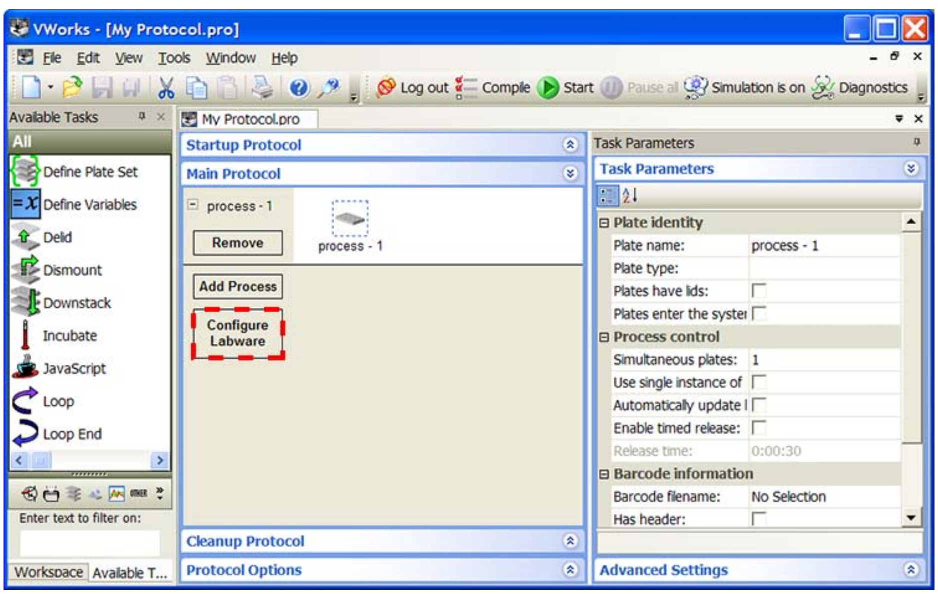
Configuring labware for Auto managed counterweight

If you are configuring labware for use as an Auto managed counterweight, see “Example: Auto managed counterweight mode” on page 290 for instructions on how to configure labware for this feature.

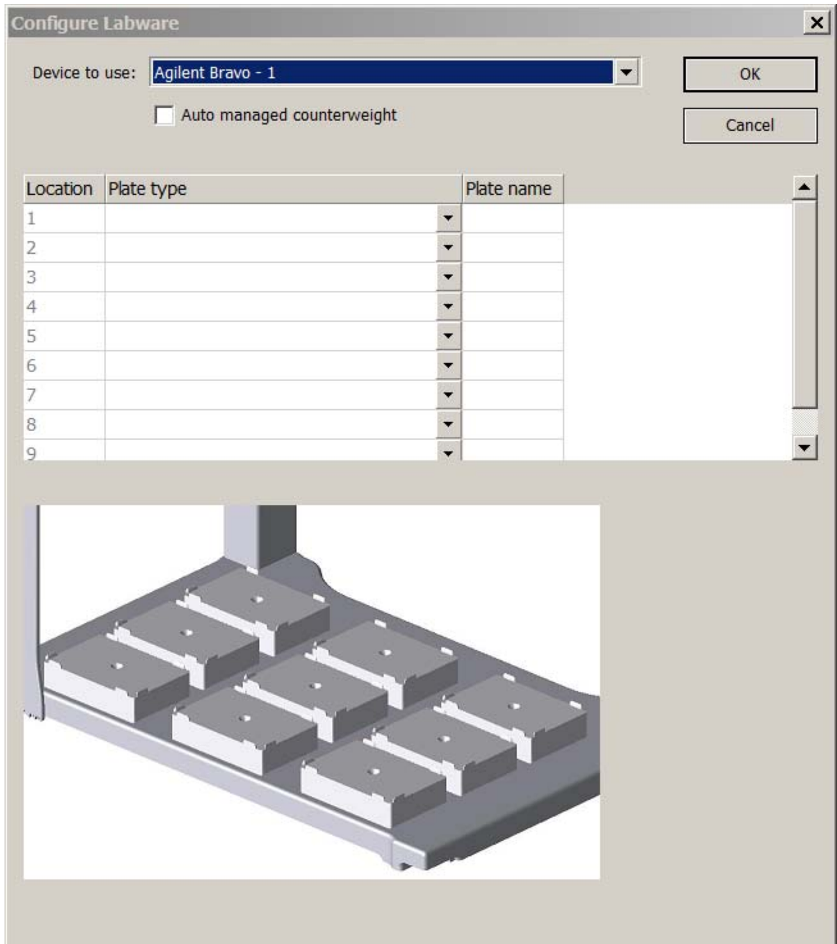
Procedure

To configure labware:

1 In the **Main Protocol** of the VWorks window, in the area below the protocol process, click **Configure Labware.**



The Configure Labware dialog box opens.



2 In the **Device to use** list, select the device on which you want to configure labware.

Note: Devices that are in the device file and have labware placement locations appear in the list. Exceptions include storage devices such as the Plate Hub Carousel or incubators, centrifuge devices, and stacking devices.

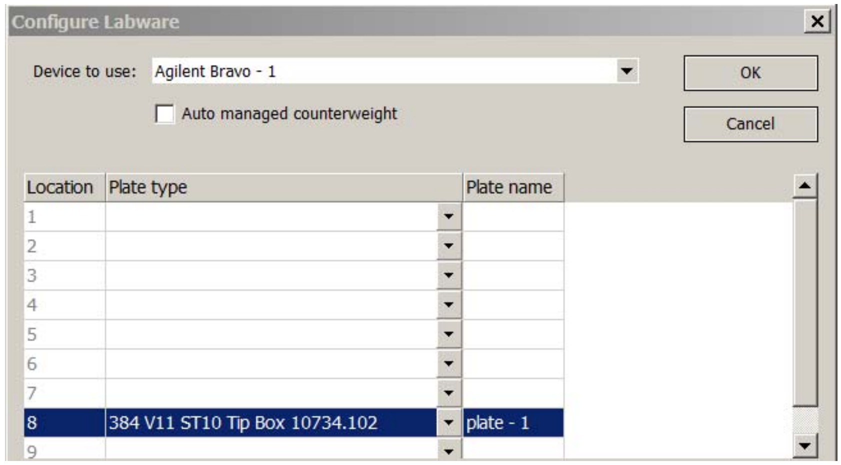
3 If you are configuring labware as an auto-managed counterweight for use with the Centrifuge or Centrifuge with Loader, select **Auto managed counterweight.** See "Example: Auto managed counterweight mode" on page 290 for further information and instructions on using this feature.

4 In the **Location** table:

a In the **Plate type** column, select the labware for the corresponding location.

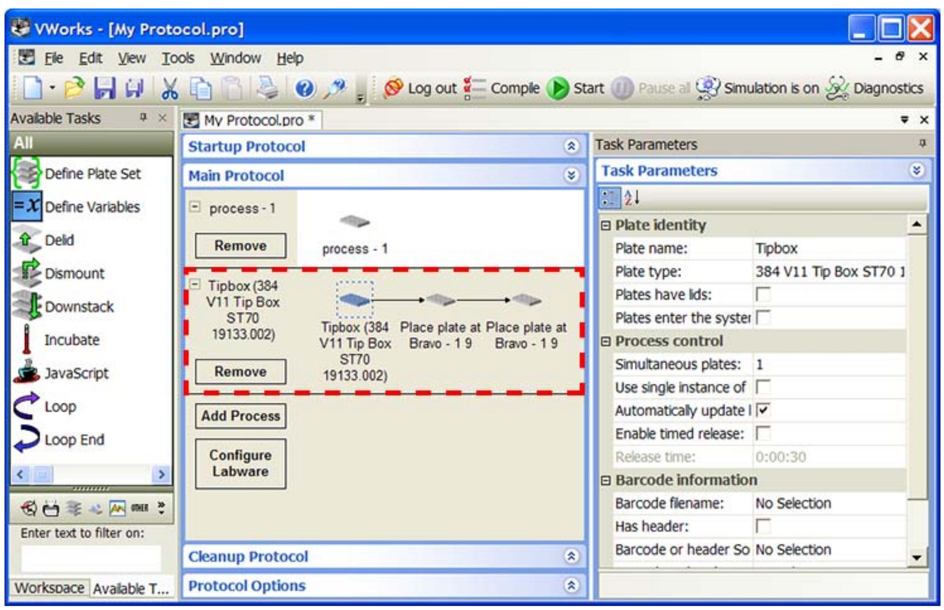
b In the **Plate name** column, double-click the text box, and then type a name for the labware you selected.

Note: For the Bravo Platform, configured static labware (from protocols created in VWorks4 version 6.2.3 or earlier) will appear in the table and graphic. However, configured accessories do not appear in the table and graphic.



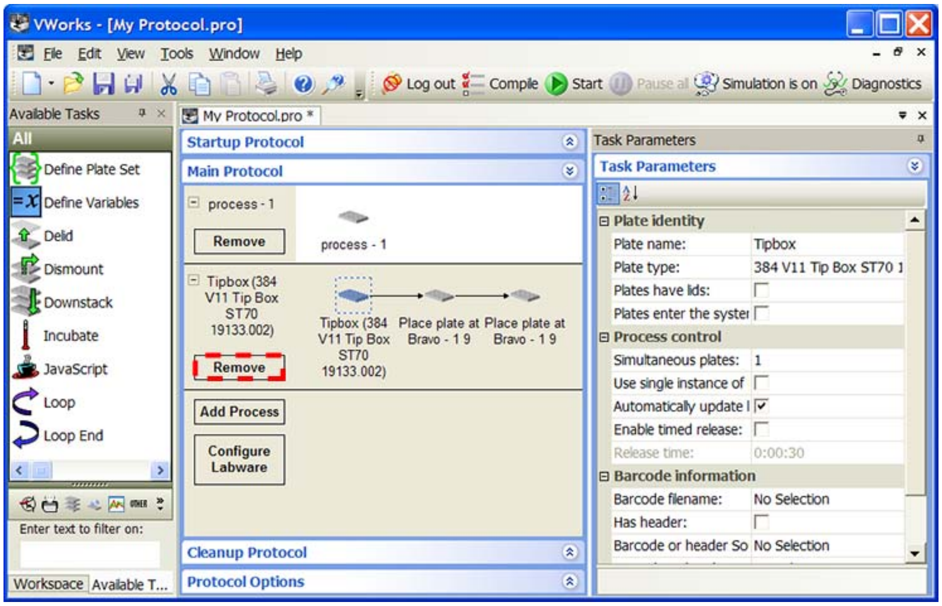
5 When you are finished, click **OK** to save the information and close the Configure Labware dialog box.

In the Main Protocol area, the labware appears. Click + to expand and view the labware icon. Notice that two Place Plate tasks are automatically added, one to indicate its starting location and the other to make sure it returns to its starting location at the end of the protocol.



6 Select the configured labware icon. In the **Task Parameters** area, set the plate properties. For a description of the plate properties, see "Setting plate parameters" on page 46.

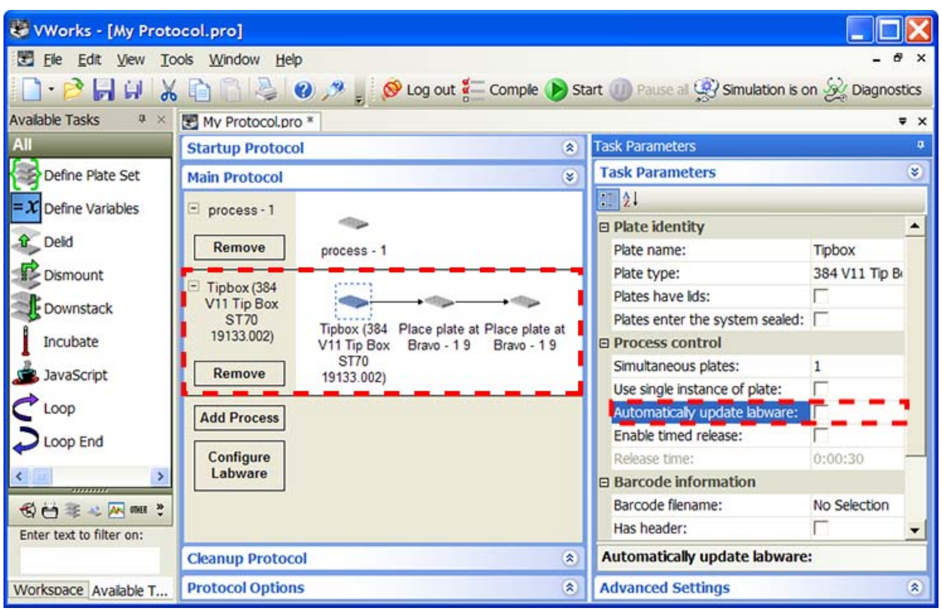
7 If you want to remove the configured labware, click **Remove.**



8 To convert a configured labware to a process plate (labware that is transferred into the system and out of the system during the run), in the **Process Control** area, clear the **Automatically update labware** check box.

The background in the configured labware process area becomes white, indicating that it is converted to a protocol process.

Note: You can use the **Automatically update labware** option to update a labware location throughout the protocol. For example, clear the check box, and then update a configured labware location from Bravo deck location 3 to location 5. Then select **Automatically update labware.** The software automatically makes the changes throughout the protocol.



Using configured labware

When you add a task in the process, you can use available configured labware by selecting it from the **Location, plate** list. In addition, if more than one subprocess uses the same configured labware, and the subprocesses are in different protocol processes, you can specify the sequence in which the subprocesses will be performed. For instructions, see "SubProcess (Bravo, Vertical Pipetting Station)" on page 388.

Related information

For information about...	See...
Planning labware use	"Planning labware use" on page 20
Process plates	"About protocols, processes, and tasks" on page 14
Planning for devices and accessories in the protocol	"Planning device and accessory use" on page 19
Static labware	"Configure Static Labware" on page 294

Adding processes

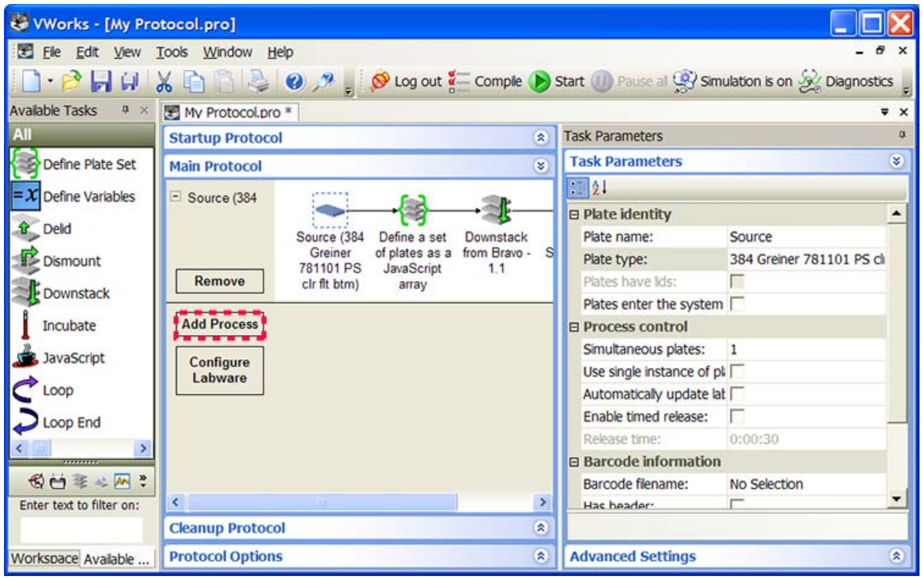
About processes

You can add processes at any time when you are creating the protocol. In addition, you can add multiple processes in one protocol.

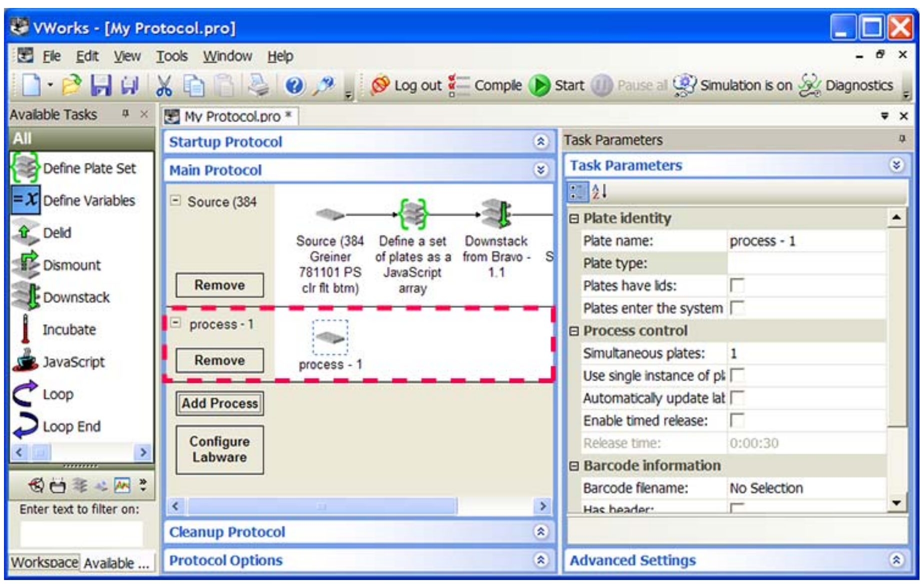
Procedure

To add a new process:

In the **Main Protocol,** click **Add Process.**



A new process lane appears.



Related information

For information about...	See...
Processes	"About protocols, processes, and tasks" on page 14
Process plates	"About protocols, processes, and tasks" on page 14
Configured labware	"Configuring labware" on page 40

Setting plate parameters

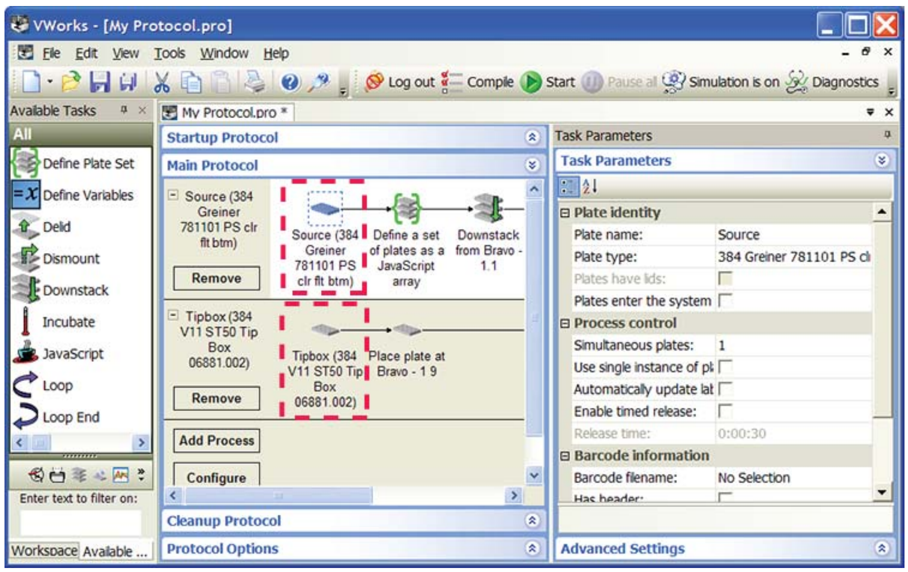
About this topic

You need to set parameters for process plates and configured labware. This topic explains how to set the parameters for both.

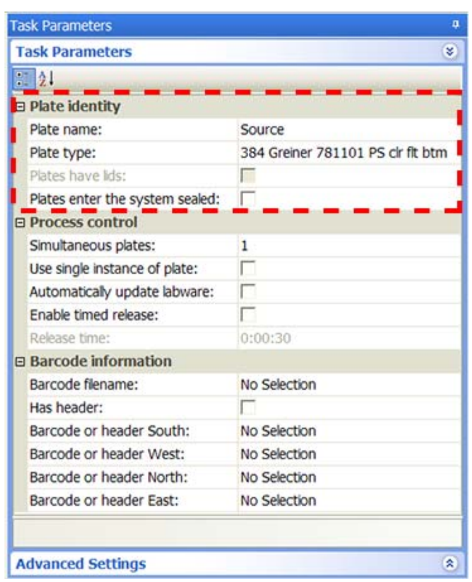
Procedure

To set plate parameters:

1 In the **Protocol** area, select the process plate icon or the configured labware icon.

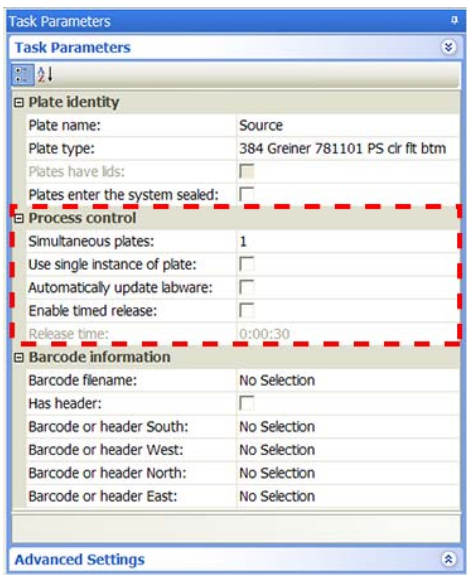


2 In the **Plate identity** area, set the following parameters:



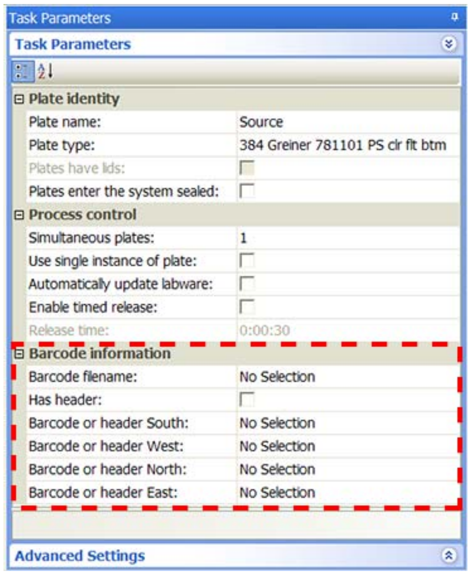
Property	Description
Plate name	The name of the labware that will help you identify the labware. For example, you can use Source Plate or Destination Plate.
Plate type	The type of labware. The list of labware types is created in the Labware Editor.
Plates have lids	The indicator that the labware entering the system has a lid. Select the check box if the labware has a lid. Clear the check box if the labware does not have a lid. *Note:* The selection is only available if the **Can have lid** option is selected in the Labware Editor.
Plates enter the system sealed	The indicator that the labware entering the system is sealed. Select the check box if the labware is sealed. Clear the check box if the labware is not sealed. *Note:* The selection is only available if the **Can be sealed** option is selected in the Labware Editor.

3 In the **Process control** area, set the following parameters:



Property	Description
Simultaneous plates	The maximum number of labware instances that are processed in the system at one time. For details about how to determine the optimum number, see "Determining the number of simultaneous plates" on page 50.
Use single instance of plate	The indicator that the plate type has only one instance and will be used repeatedly during the run. For example, you might have a single source plate from which you will aspirate repeatedly during a protocol. Select the check box if the labware has only one instance. Clear the check box if the labware has more than one instance.
Automatically update labware	The indicator that the labware is configured labware. Select the check box if the labware is a configured labware and will be used by one or more tasks in the protocol process. When you revise the protocol, the changes are automatically reflected in the configured labware's process. Clear the check box if you want to convert the configured labware to a process plate.
Enable timed release	The indicator that the labware is delivered into the system in timed intervals. If you select this property, you also need to set the Release time. Select the check box if you want the system to deliver the labware in timed intervals. Clear the check box if the system can deliver the labware without waiting the specified time.
Release time (Sec)	The length of time to wait before delivering the next labware instance into the system. For details about how to determine the release time, see "Determining the correct microplate release rate" on page 51.

4 In the **Barcode information** area, set the following parameters:



Property	Description
Bar code file name	The file you want to use to verify barcodes on incoming labware. For more information, see “Input files” on page 74. Select from the list of barcode files. If you do not see a list, make sure you specified the file location in the Protocol Options.
Has header	The indicator that the first line of the barcode file contains a header. Select if the code file name contains a header. Clear the check box if the file does not contain a header.
Barcode or header South/ West/North/East	The column ID in the barcode file. Select the column ID for the side of the labware. For example, if the barcode is on the east side of the labware, select the column ID in the **Barcode or header East** field. Select from the list of column IDs. If you do not see a list, make sure you specified the file location in the Protocol Options and selected the file from the **Bar code file name** property.

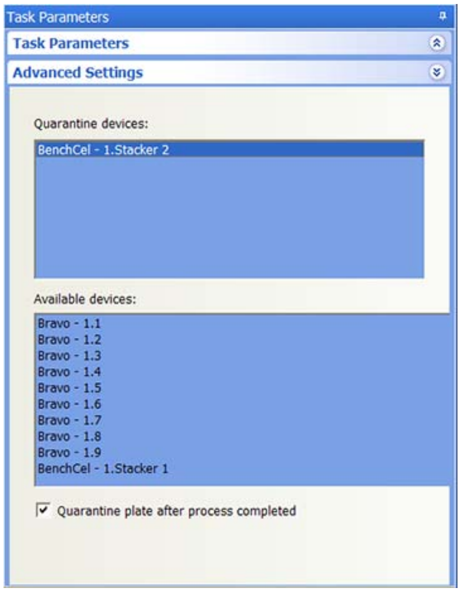
5 *Optional.* Reserve a device to store quarantined labware. Labware can be quarantined (moved aside and not processed) if a barcode label is misread, the labware orientation is incorrect, or the wrong labware type is detected. The quarantine option allows the system to continue running the protocol even though it is unable to resolve problems with the labware.

To reserve a device to store quarantined labware:

a Click **Advanced Settings.**

b In the **Available devices** list, double-click the device you want to use to store quarantined labware. The device appears in the **Quarantined devices** list.

c Select **Quarantine plate after process completed** if you want to quarantine labware after the protocol is finished. Clear the check box if you want to quarantine the labware as soon as it is encountered in the run.



For more information about how to set up quarantine criteria, see "Setting up automated error responses" on page 654.

Determining the number of simultaneous plates

Factors to consider

The number of simultaneous plates you specify depends on the following:

- The number of positions available during a protocol process.

  In general, you can specify one simultaneous plate for every task in the protocol, because each task typically uses one microplate position. For example, if your protocol downstacks a microplate, seals the microplate, labels the microplate, and then upstacks the microplate, you have three positions available: one in the robot grippers, one on the microplate sealer, and one on the microplate labeler.

  Exceptions to this generalization include cases where the same microplate position is used for more than one task and when a Vertical Pipetting Station is used. Several microplates can be positioned on a Vertical Pipetting Station at the same time.

- The number of positions in the system that will incubate microplates.

  A 10-position plate hotel contains 10 possible microplate positions. If your protocol downstacks a plate, dispenses liquid, incubates the microplate at a 10-position plate hotel, and then dispenses more of the same liquid, you have 12 positions available: one in the robot grippers, one on the dispenser, and 10 in the plate hotel.

- The number of microplates in a Reorder task.

If the protocol includes a Reorder task, the number of simultaneous plates must be equal to, or greater than, the number of microplates in the Reorder task.

- The slowest or rate-limiting task.

  A task such as a long read step or wash task on a single device can impact the number of microplates that can enter the system.

How throughput is impacted

The number of simultaneous plates you specify can impact throughput as follows:

- *The value is too high.* The protocol run might slow down because the robot will move around to avoid a deadlock. (A deadlock occurs when the number of locations available in the system is less than the number of microplates in the system, and the protocol stops.)
- *The value is too low.* The time for the protocol run can be longer than desired.

Determining an optimum number of simultaneous plates

By default, the number of simultaneous plates is set to one for each protocol process. If your protocol process uses multiple devices, you can increase the throughput of the system by increasing the number of plate instances to be processed simultaneously. The optimum number should balance high throughput and deadlock avoidance.

To determine the optimum number of simultaneous plates:

1. Run the protocol in simulation, noting the protocol process time in the log.
2. Increase the number of simultaneous plates.
3. Repeat steps 1 and 2 until the simulated process time no longer decrease. The simultaneous plates value is optimum when the process time no longer decreases.

   IMPORTANT If “Attempting to avoid deadlock by…” messages appear in the log, the protocol might have too many simultaneous plates. Decrease the number of simultaneous plates to decrease the likelihood of a deadlock during the protocol run.

Determining the correct microplate release rate

You can use the **Enable timed release** and **Release time** plate properties to control the microplate release rate and prevent bottlenecks and deadlocks. An optimum release rate should balance the number of microplates entering into the system with the number of microplates exiting the system.

Bottlenecks can occur when tasks or incubation times cause the number of incoming microplates to be greater than the number of outgoing microplates. To prevent the bottleneck, you can limit the rate of microplates entering into the system.

IMPORTANT Make sure you determine the optimum number of simultaneous plates before you change the microplate release rate.

To determine the optimum microplate release rate:

1. Run the protocol using empty labware.

2 During or after the run, open the Gantt Chart dialog box to find the task that is causing the bottleneck and determine the length of time for the task to finish. (For information about the Gantt Chart dialog box, see "Tracking the run progress of instances or devices" on page 255.)

3 Select **Enable timed release.** Type the length of the bottlenecking task time in the **Release time** box.

4 Run the protocol again using empty labware.

5 Determine whether the bottleneck is still occurring.

   - If the bottleneck is resolved, determine if previous tasks can process microplates during the wait time so that you can improve throughput. If so, decrease the release time.
   - If the bottleneck still occurs, increase the release time.

6 Repeat steps 1 through 5 until the bottleneck is resolved and the throughput is optimized.

For example:

A protocol is running with three simultaneous plates. A pipetting task in the middle of the protocol takes 3 minutes. During this time, two other microplates have already entered the system and must wait for the pipetting task to finish. A deadlock error occurs, because the system is unable to find a storage location for the third microplate that entered the system.

By turning on the timed release property and setting the release time to 180 seconds (or 3 minutes), the system will wait 3 minutes after the first microplate is in the system before delivering the next microplate. However, a microplate-piercing task and a microplate-shaking task take a total of 1 minute before the pipetting task. So the release time can be decreased to 120 seconds (or 2 minutes) to improve throughput.

Note: Bottlenecks in a protocol might not always cause a deadlock error. Instead, microplates might wait in plate hotels or on platepads. Monitor the dry protocol run to check for non-error-causing bottlenecks.

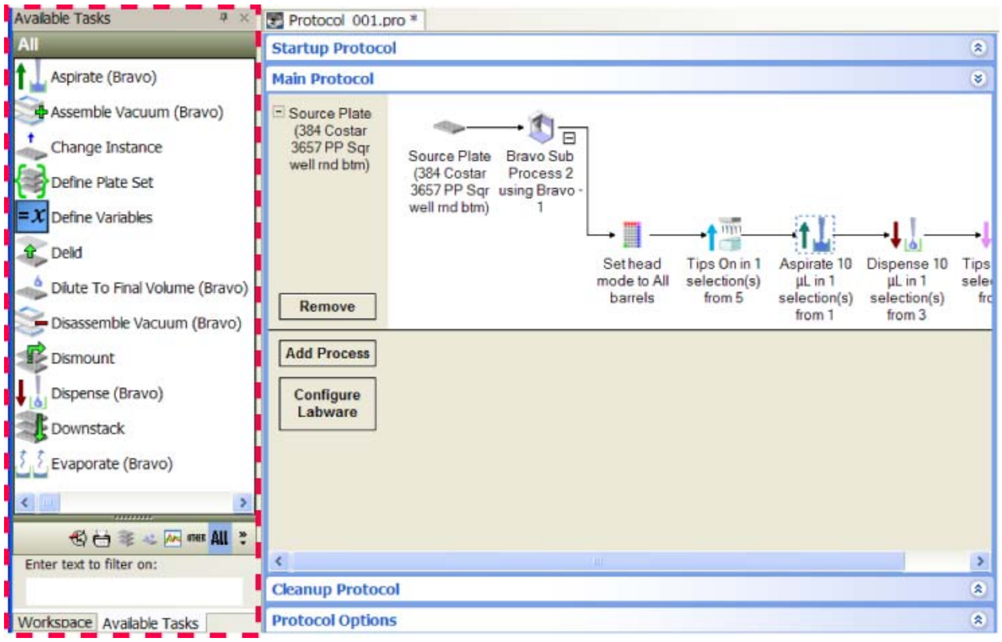
Related information

For information about...	See...
Labware Editor	*VWorks Automation Control Setup Guide*
Barcode file folder location	"Setting protocol options" on page 30
Monitoring run progress of instances, processes, and devices	"Tracking the run progress of instances or devices" on page 255
Setting up quarantine criteria	"Setting up automated error responses" on page 654
Quarantining labware that are downstacked from a BenchCel stacker	"Downstack" on page 354

Adding and deleting tasks

Viewing the list of available tasks

The list of available tasks you can use in the protocol are in the Available Tasks area.



The tasks that are available depend on a number of factors:

- *Devices*. Tasks are associated with devices. Only the tasks associated with the devices in the current device file are displayed.
- *Startup or Cleanup protocol*. Some tasks appear only in the Startup or Cleanup protocols. For a description of Startup and Cleanup protocols, see "Setting up Startup and Cleanup Protocol processes" on page 60.
- *Subprocess*. Some tasks are grouped in a device subprocess, so you can view the tasks only when you select the subprocess.

Filtering the list of available tasks

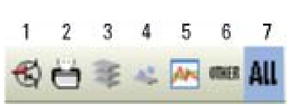
You can filter and display a subset of the task icons using one of the following methods:

- Click a task filter button.
- Type filtering text.

Using the task filter buttons

To use the filter buttons:

Click one of the following filter buttons below the list of available tasks:



Item	Task display filter	Description
1	IO Device Handling	Displays tasks that perform signal input and output operations.
2	Plate Handling	Tasks that move labware or might change the labware characteristics without changing its contents. For example, Centrifuge, delid, and Seal are plate-handling tasks.
3	Plate Storage	Tasks that move labware into or out of storage devices such as the Plate Hotel. For example, Downstack and Reorder are plate storage tasks.
4	Liquid Handling	Tasks that perform liquid-handling operations. For example, Aspirate and Dispense are liquid handling tasks.
5	Reading	Tasks that perform microplate-scanning operations. For example, the Read task is a microplate-scanning task using the Perkin Elmer Fusion Reader.
6	Other	Displays scheduling tasks, such as Define Variable, Loop, and Spawn Process.
7	All	Displays all the available tasks.

Typing filtering text

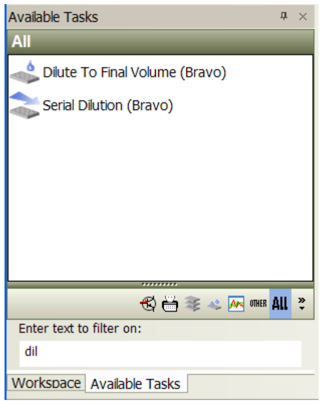
To type filtering text:

1 Click one of the filter buttons. See “Using the task filter buttons” on page 53.

For example, if you want to search all the tasks, click **All.** However, if you only want to search the liquid-handling tasks, click the **Liquid Handling** filter button.

2 Type text you want to use to filter the task list in the **Enter text to filter on** box below the list of available tasks. As you type each character, the Available Tasks list changes to meet the filter requirement.

For example, if you type `dil`, only the dilution tasks remain in the Available Tasks list.



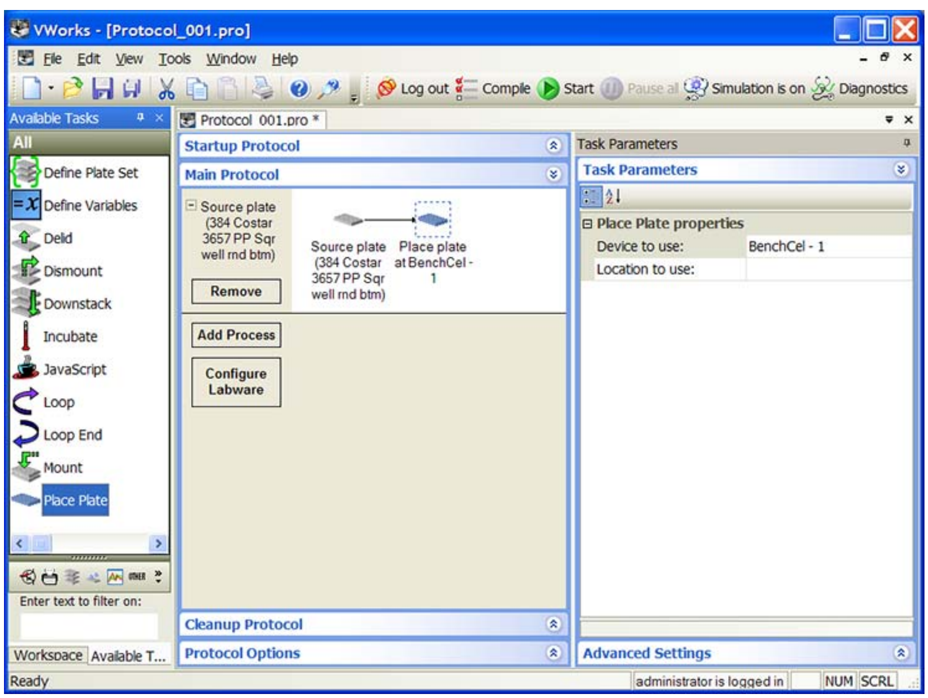
Adding tasks in a protocol

For details on how to use macros to insert a series of tasks, see "Using macros to create protocols" on page 133.

To add a task in a protocol:

1 In the **Available Tasks** area, double-click a task. Alternatively, drag a task from the **Available Tasks** area into the Protocol area. The task icon appears in the Protocol area.

In the following example, the Place Plate task is added to the protocol.



2 Set the task parameters. For a description of the task parameter, see one of the following:

- "Setting parameters for I/O-handling tasks" on page 269
- "Setting parameters for microplate-handling tasks" on page 277
- "Setting parameters for microplate storage tasks" on page 349

- "Setting parameters for liquid-handling tasks" on page 387
- "Setting parameters for scheduling tasks" on page 557

Deleting tasks from a protocol

To delete tasks from a protocol:

1. In the **Protocol** area, select a task.
2. Press **DELETE.** A confirmation message dialog box opens.
3. Click **Yes** to confirm the delete action. The task icon is removed from the protocol.

CAUTION Be aware of dependencies between tasks before you delete them. For example, the Tips On (Bravo) task relies on the pipette channel specifications in the Set Head Mode (Bravo) task to determine the number and position of pipette tips to install.

Related information

For information about...	See...
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
Using a macro to add a series of tasks	"Using macros to create protocols" on page 133

Specifying time constraints between dependent tasks

About dependent tasks and time constraints

After you add tasks to a protocol, you can specify a time interval between two dependent tasks. For example, you can specify that a microplate must be read within 30 minutes after a stop solution is added.

You can specify time constraints between any two non-scheduling tasks in the Startup, Main, or Cleanup Protocol. In addition:

- Both time-dependent tasks can be within a subprocess or a loop. If a Change Instance task is used in a loop, the time limit is used for every plate instance.
- One of the tasks can be within a subprocess and the other outside of the subprocess.
- One of the tasks can be within a loop and the other outside of the loop. If the constraining task is before a loop and the constrained task is within the loop, the constraint applies to the first loop only. If the constraining task is within the loop and the constrained task is after the loop, the constraint applies to the last loop only.
- Any number of tasks can be added between the two dependent tasks. However, if the time it takes to complete the enclosed tasks exceeds the time constraint of the dependent tasks, a compiler error results.

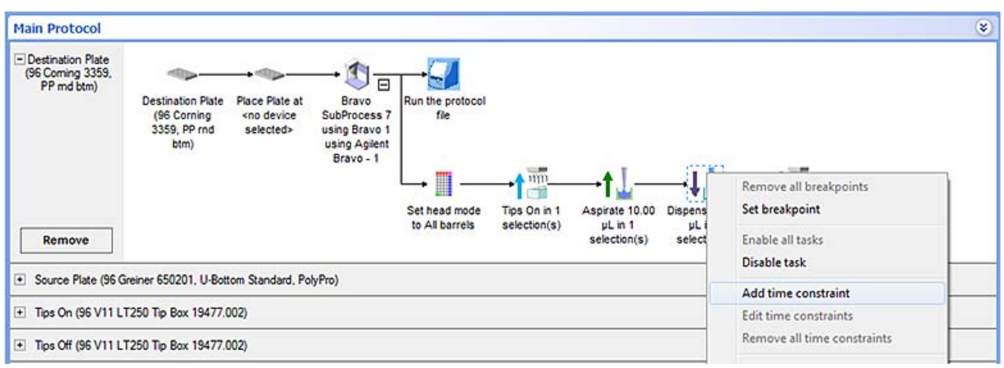
During a protocol run, if the time limit is exceeded, the run continues, but a message appears in the Time Constraint Log to alert you of this occurrence. If the task is finished before the time limit, the software waits until the minimum time limit is reached before proceeding to the next task.

Adding a time constraint

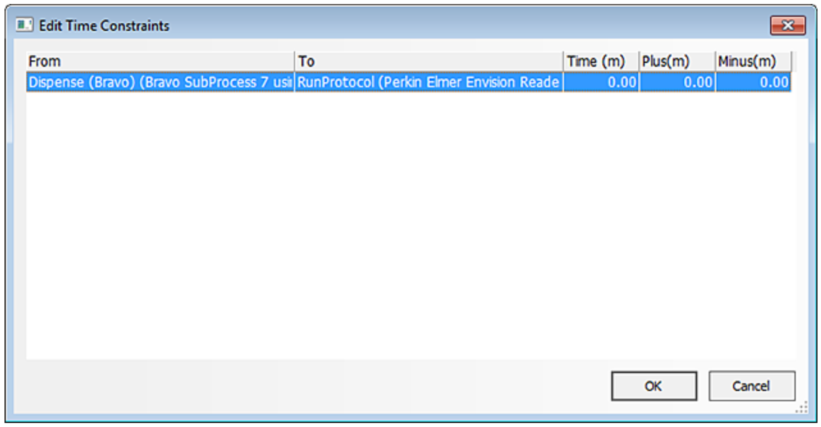
To set a time constraint between two dependent tasks:

1 In the protocol, CTRL+click the two dependent tasks to select both tasks.

2 Right-click one of the two tasks, and then select **Add time constraint.**



The Edit Time Constraint dialog box opens. Notice that the task that runs first is the constraining task, and the second task is the constrained task.



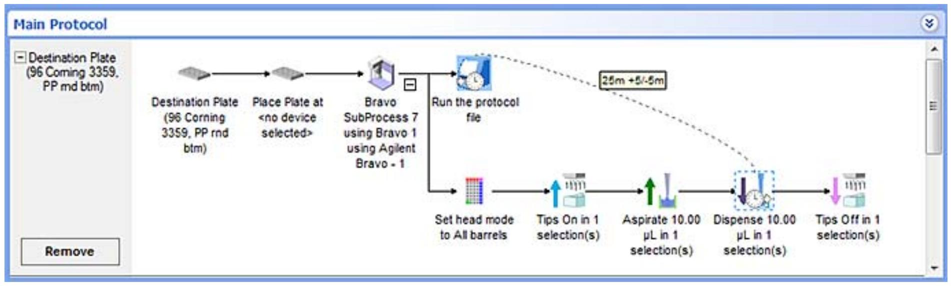
3 Specify the length of time between the two dependent tasks.

Parameter	Description
From	The first of the two dependent tasks you selected. You cannot edit this field.
To	The second of the two dependent tasks you selected. You cannot edit this field.
Time (m)	The length of time, in minutes, that is permitted for the first task. The minimum value is 1 minute.
Plus (m)	The time tolerance upper limit. For example, if you specified a total time of 25 minutes +5 minutes, the maximum time permitted for the first task is 30 minutes. The software will proceed to the next task after allowing 30 minutes for the first task to complete. The minimum tolerance is 0.5 minute.
Minus (m)	The time tolerance lower limit. For example, if you specified a total time of 25 minutes -5 minutes, the minimum time permitted for the first task is 20 minutes. The software will wait 20 minutes before proceeding to the next task even if the first task is completed in less time. The minimum tolerance is 0.5 minute.

4 Click **OK** to save the changes and return to the protocol.

A clock appears on each of the two task icons to indicate that they are time-dependent. An arrow pointing away from the clock indicates that the task sets a constraint. (In the following example, the Dispense task is the constraining task.) An arrow pointing into the clock indicates that the task is constrained by an earlier task. (In the example, the Run the protocol file task is the constrained task.)

When you rest the cursor on either task icon, a dotted line and the specified time interval appear.



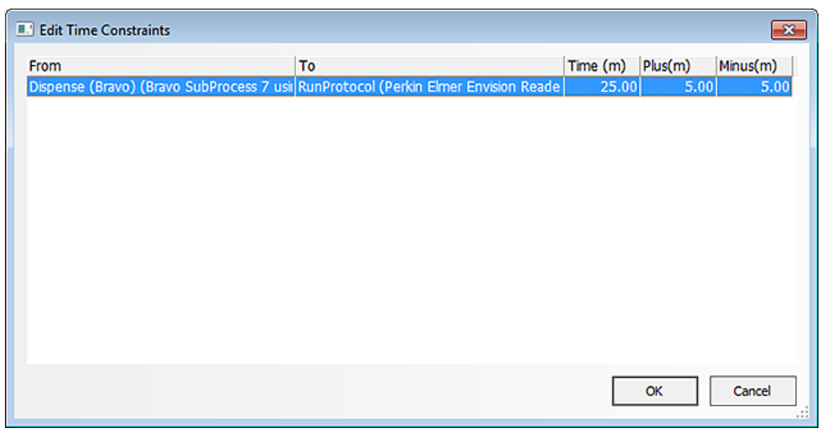
Note: A task can both constrain a later task and be constrained by an earlier task within a protocol. A clock icon with an arrow pointing into and away from it indicates that the task serves both time constraints.

Editing a time constraint

Note: The software allows you to edit time constraints during protocol runs. The time constrained tasks must be downstream from the task that is currently running. However, an edited time constraint that is too close to the running task will not likely be applied.

To edit a time constraint between two dependent tasks:

1. In the protocol, right-click one of the two dependent tasks, and then select **Edit time constraint.** The Edit Time Constraint dialog box opens. All constraints relevant to the selected task appear in the dialog box.



2. Change the Time, Plus and Minus values, and then click **OK** to save the changes and return to the protocol.

To display and edit all time constraints in a process or subprocess:

1. In the protocol, right-click the process plate or subprocess icon, and then select **Edit time constraint.** The Edit Time Constraint dialog box opens. All constraints relevant to the selected task appear in the dialog box.
2. Change the Time, Plus and Minus values, and then click **OK** to save the changes and return to the protocol.

Removing time constraints

To remove a time constraint between two dependent tasks:

In the protocol, right-click one of the two dependent tasks, and then select one of the following:

Command	Description
Edit time constraint	Allows you to remove a selected time constraint between two tasks. When you use this command, the Edit Time Constraints dialog box opens. Select the time constraint you want to remove, and then click **Remove.**
Remove all time constraints	Removes all time constraints relevant to the selected task.

Related information

For information about...	See...
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

Setting up Startup and Cleanup Protocol processes

About Startup and Cleanup Protocols

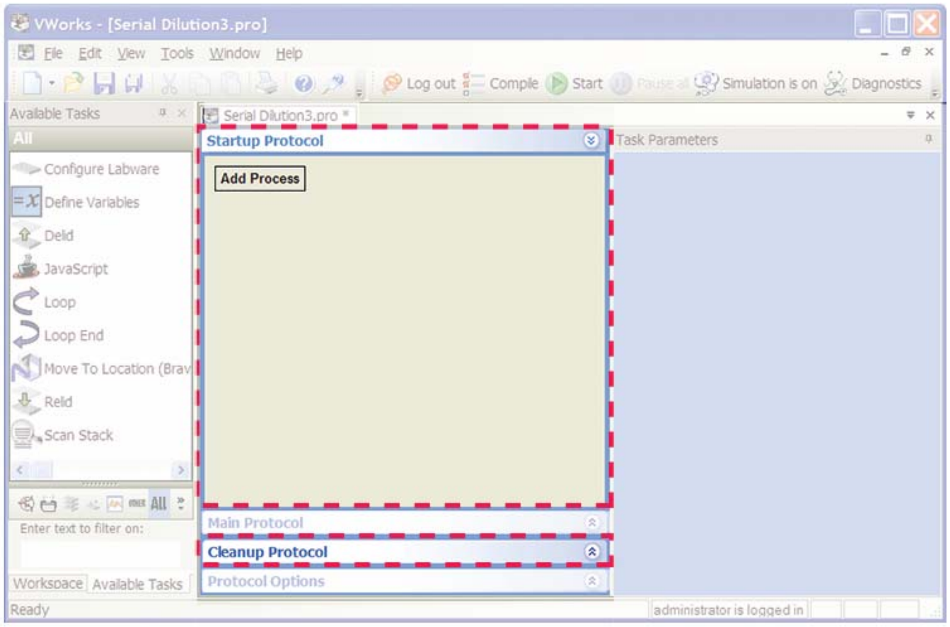
Startup Protocols are processes that are run before the Main Protocol starts. For example, you can use the Startup Protocol to prime pumps with fluid.

Cleanup Protocols are processes that are run after the main protocol finishes. For example, you can use the Cleanup Protocol to purge lines with a buffer or cleaning solution.

Procedure

To set up a startup protocol or cleanup protocol process:

1 In the protocol area, click **Startup Protocol** or **Cleanup Protocol.**



2 Add processes, set plate properties, and add tasks as you would in the Main Protocol.

Related information

For information about...	See...
Adding processes	"Adding processes" on page 44
Setting plate properties	"Setting plate parameters" on page 46
Adding tasks	"Adding and deleting tasks" on page 53

Saving the protocol

CAUTION When you edit a protocol, the changes take effect immediately. You must save the changes before you exit the VWorks software or the changes will be lost.

IMPORTANT You must have administrator or technician privileges to save a protocol.

To save a protocol:

Select **File > Save.**

If you just created a new protocol, the Save As dialog box opens to allow you to assign a name to the protocol before saving the file.

IMPORTANT You should regularly back up the protocol and associated components in case they become damaged or lost. For instructions, see "Exporting and importing protocols and associated components" on page 626.

Related information

For information about...	See...
User accounts and privileges	*VWorks Automation Control Setup Guide*
Adding notes about a protocol	"Setting protocol options" on page 30
Compiling a protocol	"Compiling the protocol" on page 63
Simulating a protocol run	"Simulating the protocol run" on page 64
Backing up protocol and associated components	"Exporting and importing protocols and associated components" on page 626

Opening a plugin

About this topic

The VWorks can interact with plugins that have been developed using the VWorks Hooks Interface. You might want to open a plugin to further configure a protocol or perform some additional tasks at run time.

If your team has developed plugins and the plugins are installed on the computer, you can use the following procedure to open the plugin.

Procedure

To display the graphical interface for your plugin:

1. In the VWorks main window, choose **Tools > Open Hooks Plugin**, and then click the file name (.dll) of the plugin that you want to open.
2. Perform the tasks required in the plugin.

Related information

For information about...	See...
Using a plugin	User documentation for the plugin
How to save a protocol	"Saving the protocol" on page 61

For information about...	See...
VWorks Hooks Interface	User guide for the VWorks Hooks Interface

Compiling the protocol

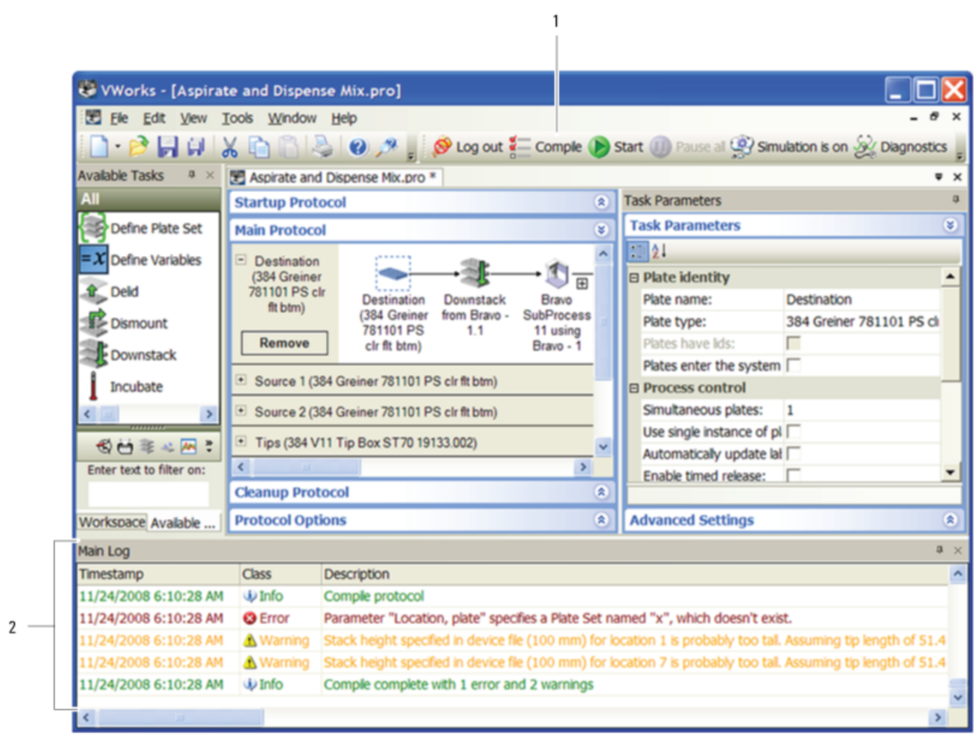
You can compile a protocol before you run it. During the compiling process, the software reports errors found in the protocol. You can use the error information to troubleshoot the protocol.

Note: The software automatically compiles the protocol whenever you start a run.

To compile a protocol:

1 On the toolbar, click **Compile**. The Main Log area displays any errors found. For troubleshooting information, see “Maintenance and troubleshooting” on page 623.

2 Review the error and warning messages in the Main Log. You should fix all the errors. Depending on the protocol-writing stage, you can choose to ignore some of the warnings.

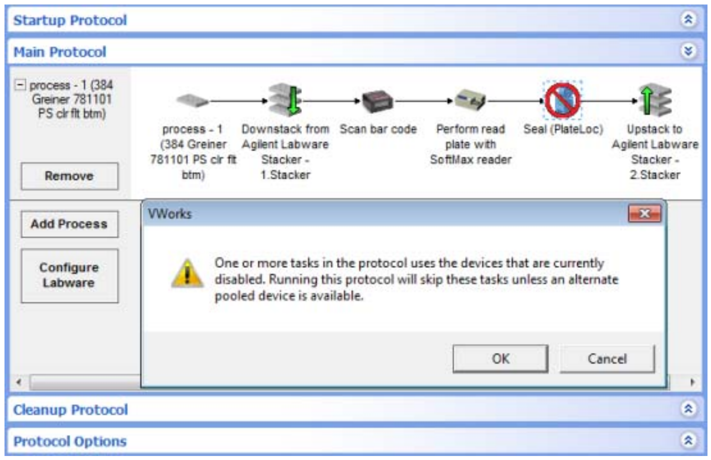


3 Repeat steps 1 and 2 until the protocol compiles error-free.

4 Save the changes you made to the protocol.

Compiling a protocol with disabled devices

When you compile a protocol that contains tasks pointing to a disabled device, and no equivalent devices are available in the pool, the software displays a warning message and a is displayed on the affected task.



Click **OK** to compile the protocol.

For more information about temporarily disabling devices in a device file, see "Disabling and enabling a device in the device file" on page 215.

Related information

For information about...	See...
Troubleshooting compile errors	"Maintenance and troubleshooting" on page 623
Disabling a device in a device file	"Disabling and enabling a device in the device file" on page 215
Saving the protocol	"Saving the protocol" on page 61
Simulating the protocol run	"Simulating the protocol run" on page 64
Printing the protocol	"Printing protocols" on page 70

Simulating the protocol run

About run simulations

A protocol simulation is a virtual run where the software performs the tasks without moving robots or labware. Simulation runs are useful for troubleshooting scheduling and placement errors and optimizing throughput.

After checking for compiler errors, you can start a simulation to verify that tasks are completed and sequenced correctly. In addition, the simulation can help you find:

- Deadlocks
- Periods of inefficiency, such as when the robot is not being used
- Microplates spending different times at critical steps when they should be run under identical conditions
- A number of simultaneous plates that is too high or too low

Note: You can simulate the scheduling of multiple protocol runs, including those that run simultaneously.

Simulation time

The software uses the following lengths of time to simulate robot motions:

Robot movement	Time (sec)	
Direct Drive Robot and 3-Axis Robot	Slow	8
	Medium	6
	Fast	4
Peak KiNEDx Robot	Slow	16
	Medium	10
	Fast	4
Phantom Robot	Slow	16
	Medium	10
	Fast	4
Task	Specified in the task Advanced Settings area. The default value is 5.0.	

Note:

- The robot movement times are averages and might be conservative. The robots might take more or less time during a real protocol run.
- When using the Phantom Robot in a simulation, the software does not prompt the operator to move labware. Instead, the software uses the simulation time shown in the table.

You can override the default task simulation time in two ways:

- Perform a number of dry runs and allow the software to determine the average task time as it reaches a steady-state. The software then uses the average of the times during the subsequent simulation. See "Optimizing simulation time" on page 65.
- Manually set the desired task time in the Advanced Settings area of a task. See "Setting desired task times" on page 66.

Optimizing simulation time

To increase the accuracy of the run time, you can determine the optimum simulation time.

To determine the optimum task simulation time:

1. Start the simulation with the default task run times and the number of microplates expected for a run.
2. Resolve deadlocks and major errors in the protocol. Use the Gantt Chart to identify rate-limiting tasks and make adjustments to improve throughput. For instructions, see "Tracking the run progress of instances or devices" on page 255.
3. Turn off the simulation mode and perform a dry run with empty labware. Make sure the number of plate instances processed equals the number of simultaneous plates allowed. Doing so allows the run to reach a steady state and enables the software to update the task simulation times with actual times.

   IMPORTANT If the number of plate instances does not reach the number of simultaneous plates allowed, the software will not be able to update the task simulation times.

4. Save the protocol to save the updated task simulation times.

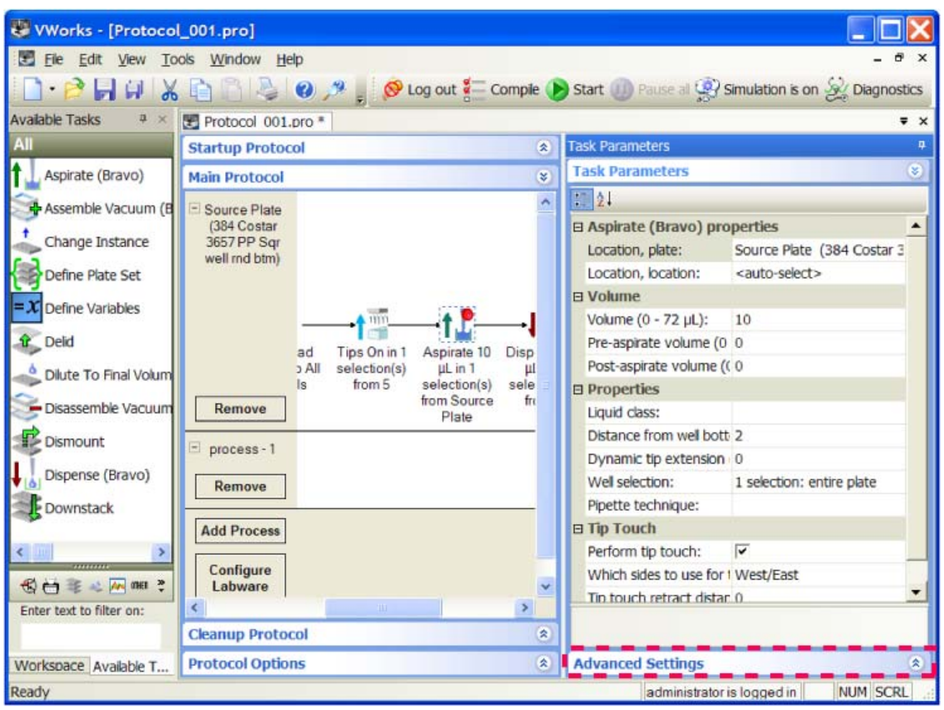
Setting desired task times

You can manually set task times to override the default task time or the optimized task time.

IMPORTANT The manually specified time is only retained and used for simulations. The optimized time will override the manually specified time if you turn off the simulation mode and run the protocol.

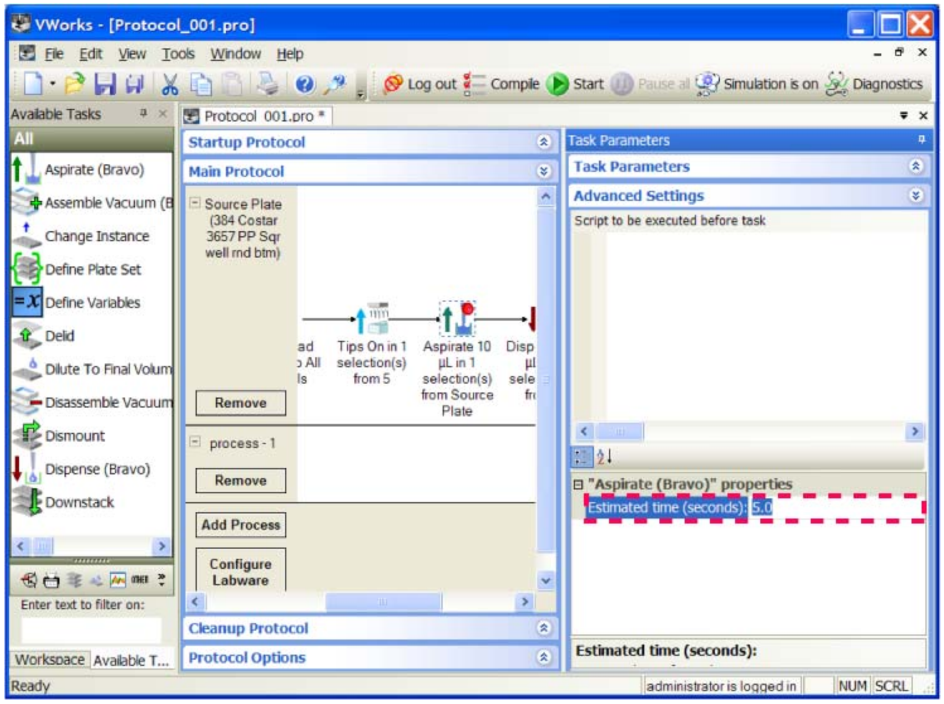
To set a desired task simulation time:

1. Select a task in the protocol.
2. In the **Task Parameters** area, click **Advanced Settings.**



3 At the bottom of the **Advanced Settings** area, view the existing value or type a new value for **Estimated time** for the selected task. To provide optimal simulation times, see "Optimizing simulation time" on page 65.

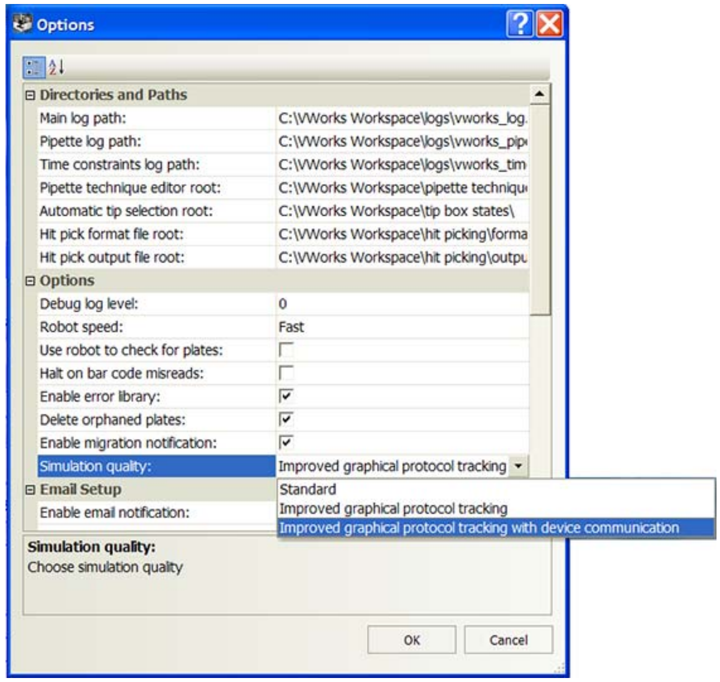
Note: Only some tasks have adjustable run times.



Procedure

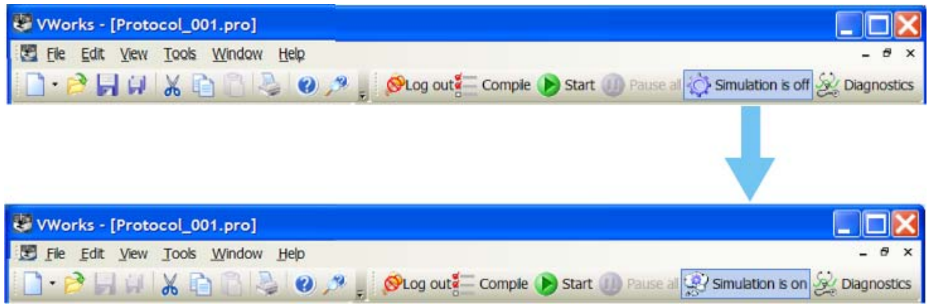
To run the simulator:

1 Select **Tools > Options.** In the **Options** dialog box, select the simulation quality you want to use:



Simulation quality	Description
Standard	Protocol tracking (green dots) is close to accurate, but not exact. This option results in faster simulation, because certain physical constraints are not simulated. In addition, less time is devoted to refreshing the graphical user interface.
Improved graphical protocol tracking	Protocol tracking is more accurate but results in longer simulation time than the Standard quality option. Similar to the Standard quality option, the software does not query the device plugins for physical constraints. However, the software ensures the green dots in the Protocol area are positioned accurately during the simulation.
Improved graphical protocol tracking with device communication	Protocol tracking is the most accurate. The software queries the device plugins to ensure every move is physically permissible, resulting in a slower but more accurate simulation.

2 Turn on the simulation mode: Click **Simulation is off** on the toolbar. The button changes to **Simulation is on.**

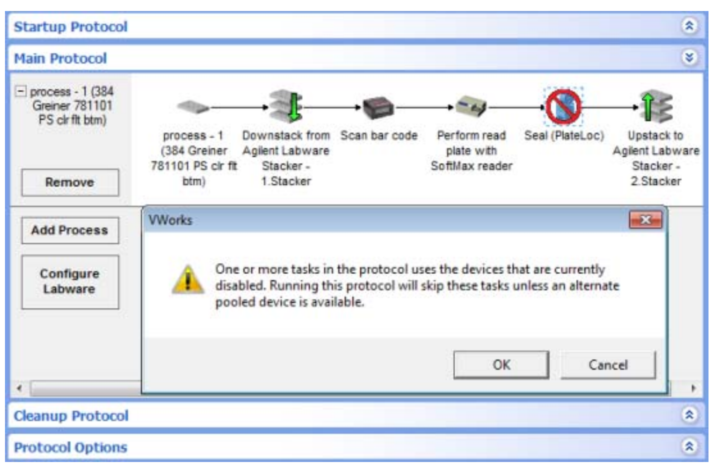


3 Start the protocol run. See “Starting the protocol run” on page 237 for the different ways you can start a run.

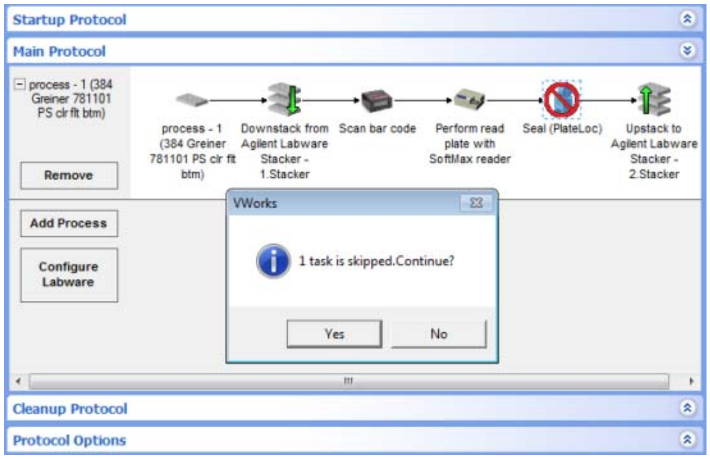
The simulation starts. The Run Configuration Wizard and other dialog boxes open as they would in a real protocol run. Follow the instructions in the dialog boxes to proceed with the simulation. For detailed information about Run Configuration Wizard, see “Starting the protocol run” on page 237.

Simulating a run with a disabled device

When you run a protocol in simulation that contains tasks pointing to a disabled device, and no equivalent devices are available in the pool, the software displays a warning message and a  is displayed on the affected task.



Click **OK** to run the protocol. The following dialog box appears indicating how many tasks will be skipped:



Click **Yes.** The Run Configuration Wizard opens. Follow the instructions in the wizard. For details on the wizard, see "Starting the protocol run" on page 237. During the run, the tasks associated with the disabled device are skipped.

For more information about temporarily disabling devices in a device file, see "Disabling and enabling a device in the device file" on page 215.

Related information

For information about...	See...
Compiling the protocol	"Compiling the protocol" on page 63
Selecting simulation qualities	"Setting general and view options" on page 224
Changing the robot speed	"Setting general and view options" on page 224
Disabling a device in a device file	"Disabling and enabling a device in the device file" on page 215
Scheduling multiple runs	"Starting the protocol run" on page 237
Managing runsets	"Managing runsets" on page 243
Monitoring runs	"Monitoring the overall run progress" on page 254
Pausing runs	"Pausing the run" on page 261
Deadlock recovery	"Recovering from deadlocks" on page 648

Printing protocols

You can use a printout of your protocol to:

- Troubleshoot the protocol
- Present in a report

Before you start

Make sure:

- The computer is connected to the printer you want to use.
- The printer driver is installed on the computer that has VWorks software installed.

Procedure

To print a protocol:

1 Open the protocol you want to print.

2 In the VWorks window, select **File > Print.**

The protocol printout shows the following:

Serial Dilution3.pro

Plate Process: (96 Greiner 655101 PS Clr Rnd Well Flat Btm) called process - 1:
Place Plate: Place plate at Bravo - 1 7

Plate Process: (96 Velocity11 06880.002 Tip Box ST200) called Tip Box - Full (co
nfigured):
Place Plate: Place plate at Bravo - 1 9
Place Plate: Place plate at Bravo - 1 9

Plate Process: (96 Velocity11 06880.002 Tip Box LT200) called Tip Box - Empty (
process):

****
SubProcess:'Bravo Sub Process 2':
Set head mode to 4 rows: A-D, 3 columns: 1-3
Tips On in 1 selection(s) from Tip Box - Full (configured)
Serial dilute process - 1, putting tips-on in 6, putting tips-off in 3
Tips Off in 1 selection(s) from 3

SubProcess:'Bravo Sub Process 3':
Tips On in 1 selection(s) from Tip Box - Empty (process)

****
Bravo Pipettor
- Bravo - 1

Section	Description
Protocol name	The name of the protocol.
Process name	The name of the process and the associated labware type.
Tasks	The tasks in the process, the devices used in the task, and location information.

Section	Description
Subprocesses	The subprocess and all tasks in the subprocess.

Related information

For information about...	See...
Compiling protocols	"Compiling the protocol" on page 63
Simulating protocol runs	"Simulating the protocol run" on page 64
Troubleshooting problems	"Maintenance and troubleshooting" on page 623

3
Creating a protocol: advanced topics

Read this chapter if you are an administrator or technician who writes protocols.

This chapter contains the following topics:

Tracking barcodes

About this topic

The VWorks software allows you to track barcodes that are read or applied at a device. This topic explains how to set up barcode tracking in the software.

Barcode readers

Barcode readers can be installed on a number of devices in the lab automation system. For example, a barcode reader can be installed on a platepad so that every time a labware is placed there, the barcode is automatically read.

If your lab automation system contains a Microplate Labeler, you have the ability to print and apply barcode labels on microplates. If the Microplate Labeler includes an optional reader, barcode labels can be read.

To track barcodes in the system, you need the following:

- "Input files" on page 74
- "Data files" on page 75

Input files

Description

Created by upstream applications such as LIMS or created manually, barcode input files can be used to verify barcodes on incoming labware. As barcoded labware enter the system through tasks such as Downstack, the barcodes are scanned and compared against the barcodes in the barcode input file. You specify the input file to use when you set the process plate parameters. You can also specify a device to use to quarantine plates for which the barcode label is misread or incorrect. For details see "Setting plate parameters" on page 46.

Note: The software remembers where you are in the input file. For example, if you have 10 entries in the file, and the first protocol run processes the first six microplates, the next run will start at entry seven. To reset the file to the first row, reload the file. Barcode files are reloaded when you open a protocol file, when you update the Barcode file directory in Protocol Options, or when you update the Barcode filename or Has header option when setting up the plate parameters.

Requirements

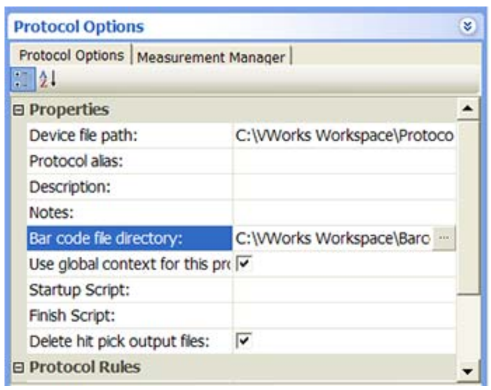
Input files must meet the following requirements:

- The file must be a comma-separated value format with the .bar, .csv, or .txt file name extension.
- *Optional.* The file can contain a header that describes the columns in the file (for example, Plate Barcode).
- The file must contain at least one column.
- One of the columns must list the barcodes.

The following example shows an input file that is displayed in Excel. The file contains a header that labels two columns: Plate ID and Barcode.

	A	B
1	Plate ID	Barcode
2	Plate001	100001
3	Plate002	100002
4	Plate003	100003
5	Plate004	100004
6	Plate005	100005
7	Plate006	100006
8	Plate007	100007
9	Plate008	100008
10	Plate009	100009
11	Plate010	100010
12	Plate011	100011

The input file can be stored anywhere on the computer that runs the VWorks software. However, you must specify its location in the Protocol Options area.



Data files

Description

Created by upstream applications such as LIMS or created manually, barcode data files are used by the Print task or the Print and Apply task to:

- Print barcodes on labels. For example, the software reads a row in the file and prints the barcode presented in that row.
- Look up barcodes to print on labels. For example, the software reads the east-side barcode on a labware, looks up the barcode in the file, and prints the north-side barcode that is in the corresponding column (same row).

You specify the data file to use when you specify the task parameters for the Print or Print and Apply task. See “Print” on page 578 and “Print and Apply” on page 318.

Requirements

Barcode data files must meet the following requirements:

- Be a comma-delimited text file with the .csv file name extension
- *Optional*. Contain a header, which can be in any format.
- Contain at least one column. For example, the file can contain four columns, each representing a side of the microplate.

The following example shows a data file displayed in Excel. The file contains four columns, each representing a side of the microplate. In addition, the file contains a header in row 1.

	A	B	C	D
1	North	South	East	West
2	A00001	B00001	C00001	D00001
3	A00002	B00002	C00002	D00002
4	A00003	B00003	C00003	D00003
5				

The data file can be stored anywhere on the computer that runs the VWorks software.

Related information

For information about...	See...
Microplate Labeler	*Microplate Barcode Labeler User Guide*
Print task	"Print" on page 578
Print and Apply task	"Print and Apply" on page 318

Using simple variables

About this topic

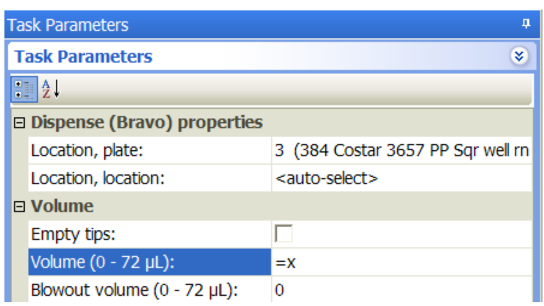
This topic presents the following:

- "About variables" on page 77
- "Variable and snippet syntax" on page 78
- "Defining variables and adding code snippets" on page 78
- "Adding user message prompts" on page 82

About variables

You can use variables to assign operator-supplied values to task parameters, provide an initial value for looping, or reference an array in another task. An example of variable usage is as follows: During a protocol run, the software will prompt the operator for aspirate and dispense volumes. The software will use the operator-supplied values during the run.

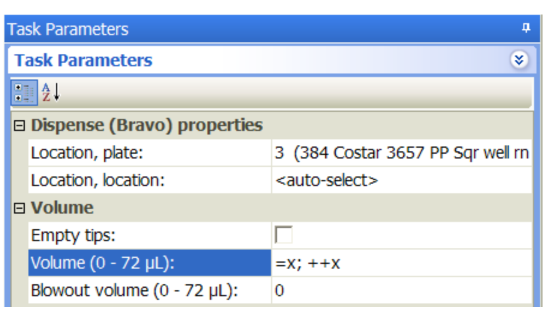
Instead of writing detailed JavaScript code in the JavaScript task Advanced Settings area, you can type a simple variable assignment and a code snippet directly in a field in the Task Parameters area. The following example shows how to assign a variable called x to the Dispense Volume parameter in the Task Parameters area.



Assigning the variable in the Task Parameters area is equivalent to providing the following line of code in the Advanced Settings area:

```
task.Volume = x;
```

You can also add a snippet of code after the variable assignment. For example, you can assign the Dispense Volume to x, and then increment it by the same amount in each loop during the protocol run.



To prompt the operator for the values to use during the protocol run, you can select the option in the Define Variables task or add User Message tasks at the desired points in the protocol. See "Adding user message prompts" on page 82.

Variable and snippet syntax

When adding a variable and code snippet in the Task Parameters area, use the following syntax:

```
=x; <code snippet>
```

where `x` is the variable name and `<code snippet>` is additional code you want to run during the task. Use standard JavaScript rules for the variable name and for the code snippet.

Note: Variables defined here must be predefined in the protocol in the Define Variables task, User Message task, a previous JavaScript code snippet, or in the Advanced Settings area.

Defining variables and adding code snippets

You can define variables using the following:

- *Define Variables task*. In the Define Variables task, you can define multiple variables and set initial values. For more information, see "Define Variables" on page 564.
- *Define Plate Set task*. In the Define Plate Set task, you can define a variable that identifies a group of process plates to be processed identically. For more information, see "Define Plate Set" on page 560.
- *Task parameter variables*. You can define a variable for almost any parameter. For example, in the Aspirate task, you can define a variable for the Volume parameter. For more information, see "Task parameter variables" on page 78.
- *Startup Protocol JavaScript task*. Using the JavaScript task in the Startup Protocol, you can define multiple variables and assign initial values to them. For more information, see "Startup Protocol JavaScript task" on page 80.

Task parameter variables

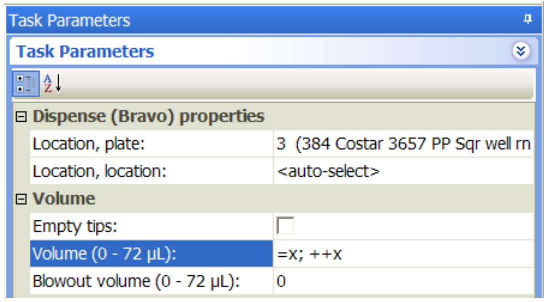
The way you define a variable for other tasks depends on the field input style:

- Text box format
- List format
- Time format

Note: Parameters that require fixed values do not accept variables. For example, passwords, deck locations, and IP addresses cannot be assigned the value of a variable.

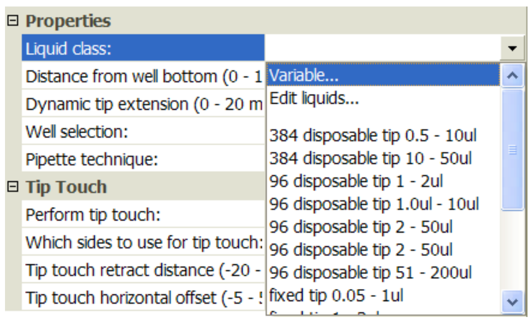
To assign a variable to a parameter whose value is displayed in a text box:

In the parameter value text box, type the variable assignment and optional code snippet. For syntax requirements, see "Variable and snippet syntax" on page 78.

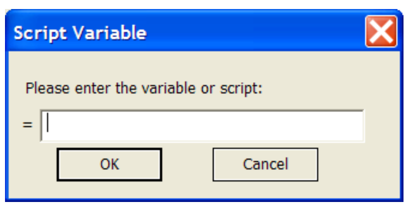


To assign a variable to a parameter whose value can only be selected from a list:

1 In the list, select **Variable.**



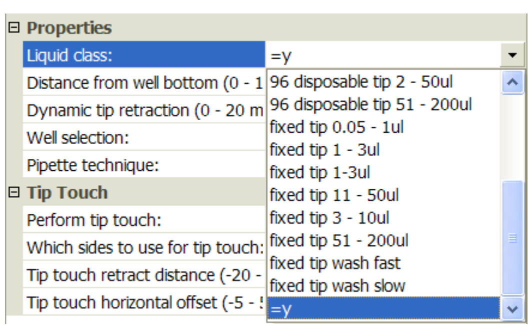
The Script variable dialog box opens.



2 Type the variable assignment and optional code snippet. For syntax requirements, see "Variable and snippet syntax" on page 78.

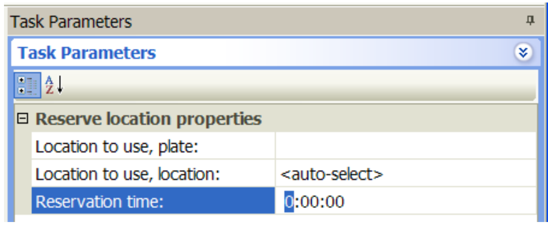
Note: The = symbol in front of the text box indicates that the software will automatically add the = symbol in front of the variable.

3 Click **OK.** The new variable and optional code snippet appear in the parameter value list.

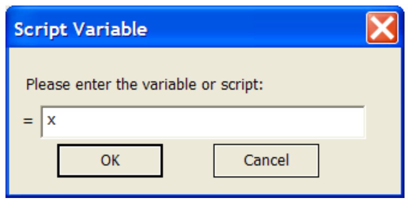


To add a variable to a parameter whose value is in the time format (h:mm:ss):

1 In the parameter value box, type = in the h, mm, or ss field. The Script variable dialog box opens.

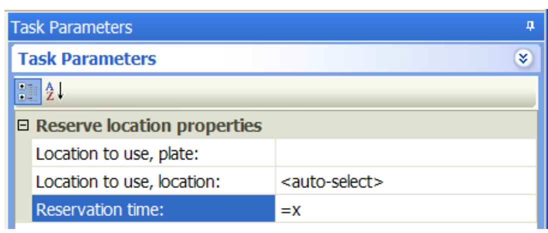


2 Type the variable assignment and optional code snippet. For syntax requirements, see "Variable and snippet syntax" on page 78.



Note: The = symbol in front of the text box indicates that the software will automatically add the = symbol in front of the variable.

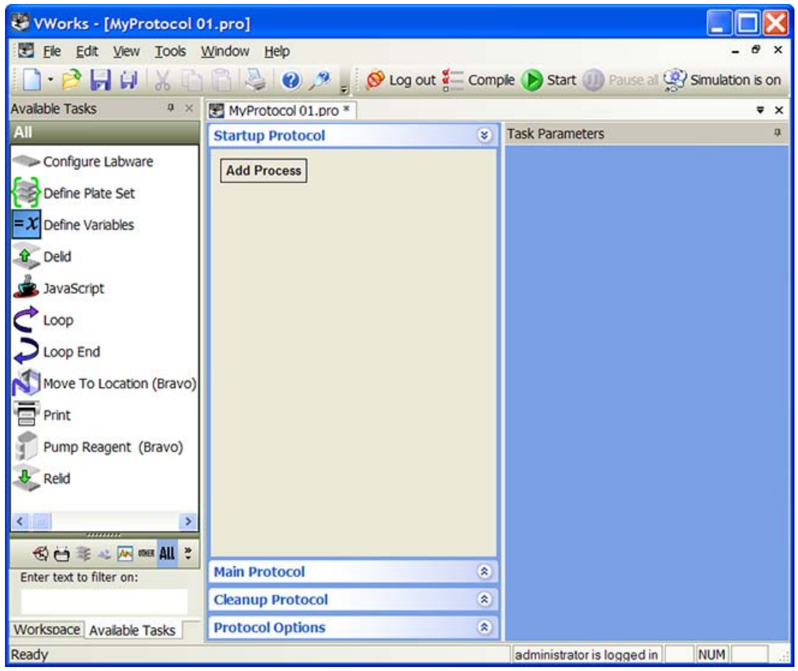
3 Click **OK.** The new variable and optional code snippet appear in the parameter field.



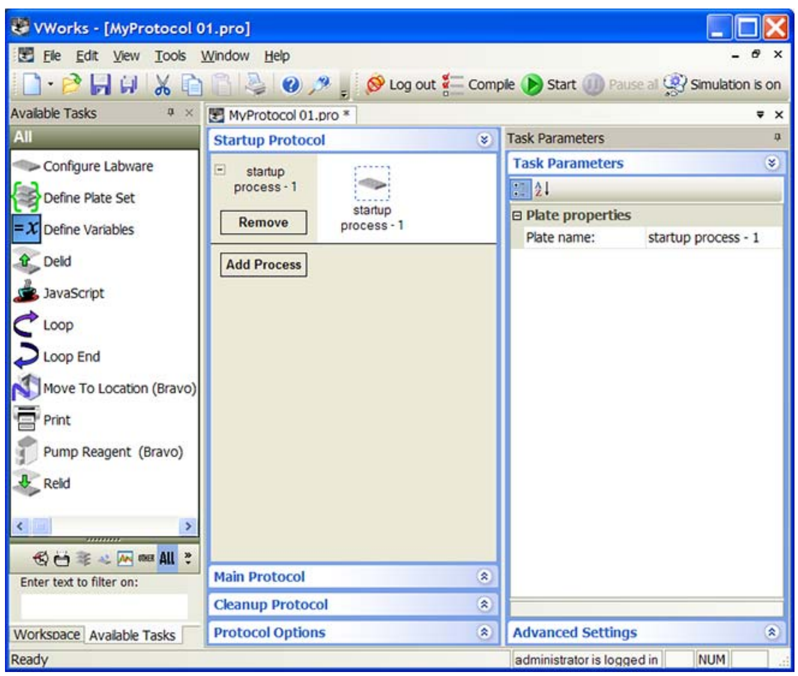
Startup Protocol JavaScript task

To create multiple variables and assign initial values to them:

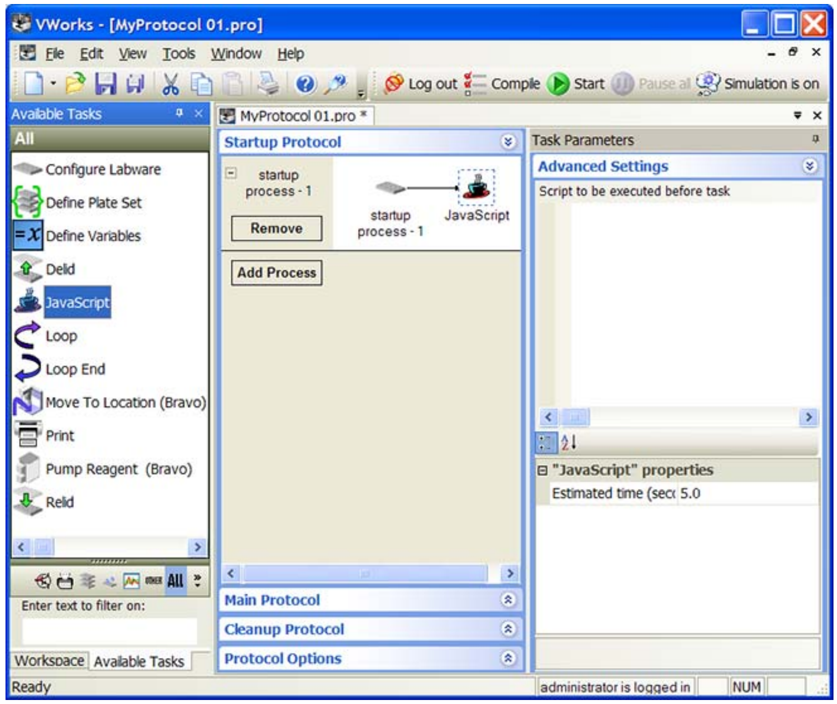
1 In the protocol, click **Startup Protocol.**



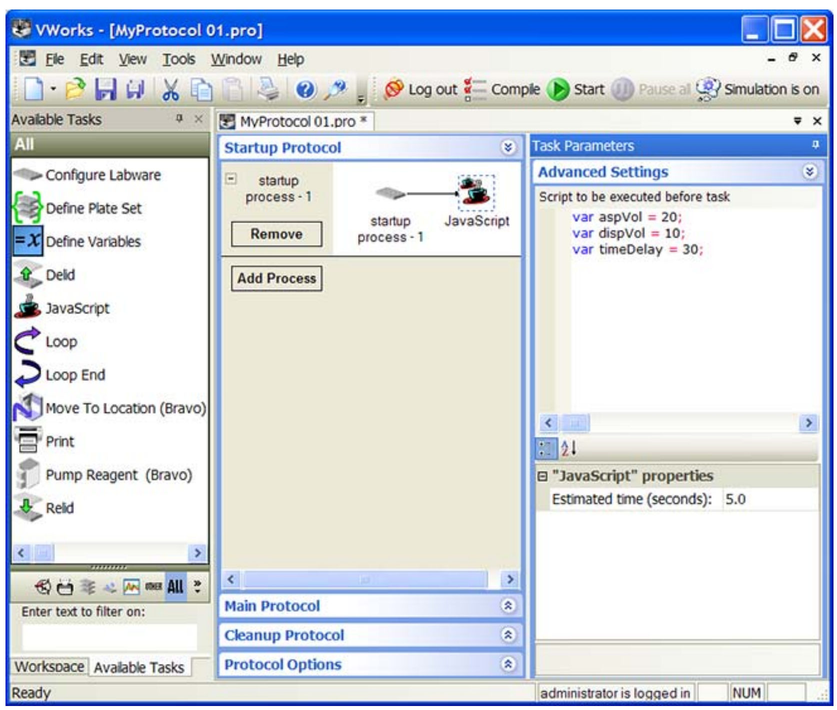
2 Click **Add Process.** A Startup process appears.



3 Add the **JavaScript** task.



4 In the **Advanced Settings** area, type the variable assignments.



Adding user message prompts

After adding variables to the desired task parameters, you can add User Message tasks to prompt operators for values at the beginning of the protocol run or at the desired points during the protocol run.

Note: You do not need to add the User Message task when using the Define Variables task. The Define Variables task contains an option to add a user message prompt.

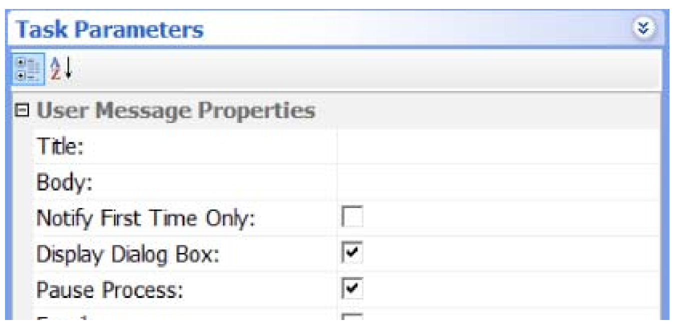
To add User Message tasks:

1 In the Startup Protocol:

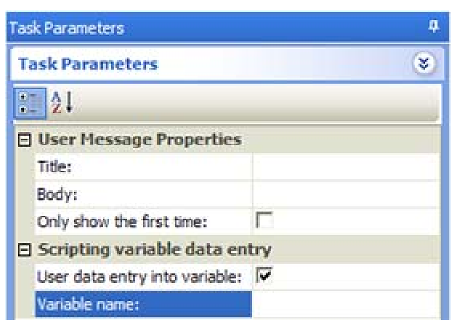
- Add a User Message task for each variable assigned in a task parameter input field.
- Add a User Message task for any variable that you assigned in the Advanced Settings area of a given task. If you added variables in the Advanced Settings area, add the User Message task after the JavaScript task in the startup protocol.

Alternatively, at the desired points in the Main Protocol, add a User Message task for each variable assigned in a task parameter input field.

2 In the **Task Parameters** area, type values and select the desired options in the **User Message Properties** table. For a description of the properties, see “User Message” on page 599.



3 Select **User data entry into variable** and type the name of the variable in the **Variable name** box. During the protocol run, the software will prompt the operator for a value and assign it to this variable.



IMPORTANT If you added multiple variables in the startup protocol, do not use the Variable name property. Instead, use the Body property to instruct the operator to set the variable values in the Advanced Settings area. For details on the Body property, see “User Message” on page 599.

Related information

For information about...	See...
User Message task	"User Message" on page 599
JavaScript task	"JavaScript" on page 573
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
Using JavaScript utilities	"Using JavaScript utilities" on page 128
Startup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60

Using JavaScript

About this topic

This topic explains how to use JavaScript in the VWorks software. For a full description of the JavaScript language, see the Mozilla Developer Center at http://www.mozilla.org/js/.

Read this topic if you have administrator or technician privileges. This topic assumes that you know how to write programs in JavaScript or have basic programming knowledge.

How JavaScript is used in the software

A comprehensive JavaScript engine is implemented in the VWorks software so that you can customize the software as follows:

- Change the existing task parameters.
- Skip a task if certain conditions are met.
- Repeat a task if certain conditions are met.

The task parameters can change dynamically during a run based on conditions such as:

- Information passed from an external source, such as a database
- The number of times the protocol has cycled
- Feedback on changing conditions or data values during the run

Examples of use

You can use JavaScript in the VWorks software to:

- Print the parameters of a task to the main log.
- Run a command that launches an external application, such as a batch file or database-updating program.
- Reduce the length of the protocol.

Where to write JavaScript

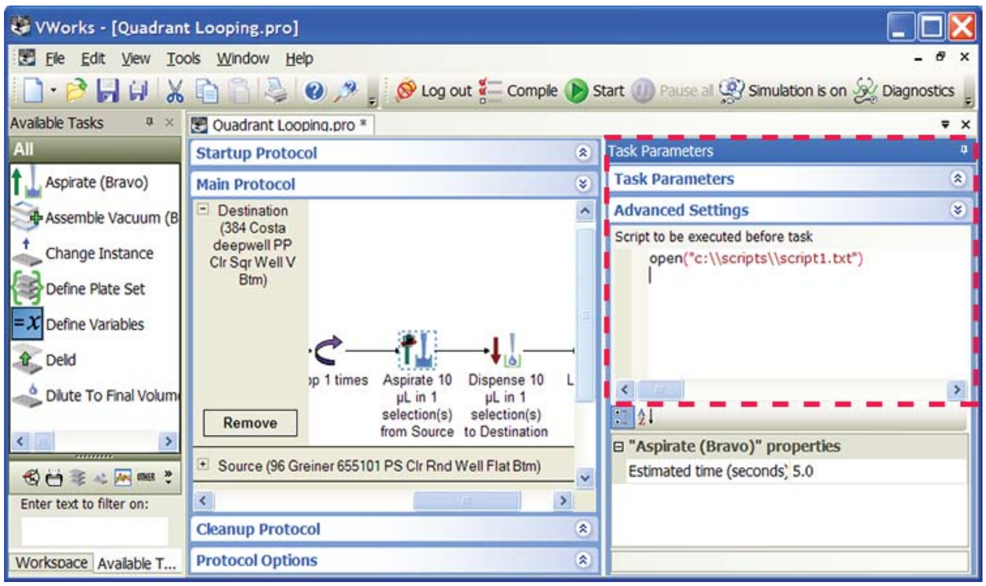
You can write JavaScript in the Advanced Settings area of any task or in the JavaScript task in the Startup, Main, and Cleanup Protocols. JavaScript written in any task is run differently than code written in the JavaScript task:

- *Any task*. During a protocol run, the software will run the JavaScript first before evaluating whether the task is possible.

  Note: In the previous version of the VWorks software, the software ran the script after evaluating whether the task was possible.

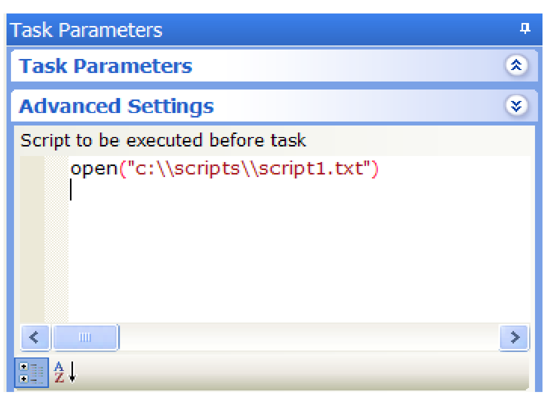
- *JavaScript task*. You can add the JavaScript task to run a program that is independent of any task. During a protocol run, the software will run the JavaScript as it reaches the JavaScript task.

The following example shows JavaScript written in the Advanced Settings area of the Aspirate task. Notice that the script is written directly in the text box.



You can add JavaScript in the following ways:

- Directly in the text box in the Advanced Settings area (see the previous example)
- As an external file that is called by the open( ) function in the text box



About JavaScript variables

By default, the values of all variables are cleared (set to undefined) before the next protocol is run. To retain the value of all variables from protocol to protocol, select the **Use global context for this protocol** option in the Protocol Options tab. When the option is turned on, a variable and its value assigned in one protocol can be used by other protocols that have the same variable until you exit the software. If using the global context option, the values of variables are not reset until the software is restarted.

CAUTION If you select the Use global context for this protocol option, variables with the same name in different protocols will overwrite each other.

Cautions

Before running a protocol containing JavaScript, compile the protocol and run it in simulation mode.

During the compiling process, the software uses values displayed in the Task Parameters area and not the values set by JavaScript. Therefore, some errors might not be detected during compilation. The values that appear in the Task Parameters area do not reflect the effects of the JavaScript.

Be aware that JavaScript does not check pipetting volumes before the run begins. Therefore, you must make sure that the pipetting steps make logical sense. For example, the software does not alert you if a JavaScript attempts to aspirate 1 µL from a microplate well that can only hold 0.5 µL.

Related information

For information about...	See...
JavaScript language	Mozilla Developer Center
Using script variables directly in task parameters	“Using simple variables” on page 77
JavaScript task	“JavaScript” on page 573
Using JavaScript utilities	“Using JavaScript utilities” on page 128
Startup and cleanup protocols	“Setting up Startup and Cleanup Protocol processes” on page 60

VWorks-defined functions

Function descriptions

The VWorks JavaScript interpreter supports the JavaScript 1.5 or later core functions and objects. VWorks also has its own functions and objects that can be used.

The following VWorks-defined functions are available globally, so they are not restricted to a particular object or programming context.

Function	Description

GetGlobalObject()	Provides the ability to get the global context from a JavaScript that is executed in a context other than the global context. Returns an object that has all the variables of the global context. For example, in a protocol that uses the local context for variables, you can have a JavaScript task that includes a variable in the global context.
	
	Example:
	
	`//(0) Create a reference to global object from this context.`
	
	`var g_ref = GetGlobalObject();`
	
	`//(1) Declare a var xx in GlobalObject.`
	
	`g_ref.xx = 4`
	
	`//(2) Increment the variable.`
	
	`g_ref.xx++`
	
	`//(3) Print the variable.`
	
	`print(g_ref.xx)`
	
	Note: In this example, lines (2) and (3) could be executing from different protocols, each in their own private context, with shared data results.
	
	Note: You can set a breakpoint on the task that uses this function to open the Debugger dialog box and view the variable (g_ref). For details on setting breakpoints, see “Using breakpoints to monitor and troubleshoot tasks” on page 643.

open()	Opens a text file and immediately executes the file contents as script.
	
	Parameter: Text string
	
	Examples:
	`open("c:\\VWorks workspace\\script.js")`
	
	`open("c:\\VWorks workspace\\script.txt")`
	
	Make sure the file is in ANSI text format. The file extension is not important.
	
	Note: Microsoft Notepad saves in ANSI text format by default.

Function	Description
print()	Prints time-stamped messages to the VWorks log. Parameter: Text string Example: `print(plate.name)`
run()	Runs a Microsoft Windows script as though it is being called from a command line. Parameters: • *Text string*. Required. Allows you to initiate a command that you could otherwise enter into the Windows Run dialog box, such as notepad text.txt (opens a file named text.txt in Windows Notepad). • *Boolean True/False*. Optional. Default is False. If True, the software waits for the function to complete before continuing (blocking). Examples: • `run("notepad")` • `run("notepad",true)` The first example starts Notepad. The second example starts Notepad and pauses the VWorks software until you exit Notepad.

Related information

For information about...	See...
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined plate objects, properties, and methods	"plate object" on page 91
VWorks-defined task objects, properties, and methods	"task object" on page 98
Using JavaScript utilities	"Using JavaScript utilities" on page 128
JavaScript task	"JavaScript" on page 573
Startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60
Using script variables directly in task parameters	"Using simple variables" on page 77
How to specify the JavaScript context for a protocol	"Setting protocol options" on page 30
JavaScript context and form design	"Understanding JavaScript context in form design" on page 204

InventoryLabware object

About the InventoryLabware object

The VWorks JavaScript interpreter provides the following objects that can be accessed by a script.

- InventoryLabware object
- plate object
- plates[] object
- task object
- plateDB object
- runset object
- forms object

This topic describes the InventoryLabware object properties.

InventoryLabware represents an individual labware in the inventory database. By passing InventoryLabware objects to plateDB methods in a JavaScript, you can manage specific labware in the inventory. To see an example of this implementation, see the importLabwareToInventory method example in "plateDB object" on page 112.

Properties

Syntax:

InventoryLabware.*property*

Property	Data type	Description
name	String	Name of the labware.
device	String	Name of the device. **IMPORTANT** The device name must match the name in the device file. Be sure to include any spaces, underscores, dashes, and capitalization in the name.
cassette	Integer	Cassette number.
slot	Integer	Slot number.
type	String	Labware type.
westbc	String	Barcode on the west side of the labware.
eastbc	String	Barcode on the east side of the labware.
southbc	String	Barcode on the south side of the labware.
northbc	String	Barcode on the north side of the labware.

Property	Data type	Description
status	String	Status of the labware: 0—OK 1—Error 2—Invalid 3—Barcode misread You can set or get the status of the labware. When you add the labware information to the database using the importLabwareToInventory plateDB method, you can set the labware status. When you retrieve labware information using the getLocationByBarcode plateDB method, you can get the labware status.
plategroup	String	Plate group to which the labware belongs.
locationgroup	String	Location group to which the labware belongs.

Methods

Methods are JavaScript functions invoked through an object. The InventoryLabware object has the following methods, available on systems that have the inventory database option. Use these methods to track labware-specific data.

Syntax:

InventoryLabware.*method( )*

Method	Description
nextPlateGroupName( )	Returns the name of the next plate group. A labware can belong to multiple plate groups. Use this method to determine all the plate groups to which a labware belongs.
nextLocationGroupName( )	Returns the name of the next location group. A labware can belong to multiple location groups. Use this method to determine all the locations groups to which a labware belongs.

Related information

For information about...	See...
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined functions	"VWorks-defined functions" on page 87
Other VWorks-defined objects	• "plate object" on page 91 • "runset object" on page 123 • "plateDB object" on page 112 • "task object" on page 98 • "forms object" on page 126
Using JavaScript utilities	"Using JavaScript utilities" on page 128
JavaScript task	"JavaScript" on page 573
Startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60

plate object

About the plate object

The VWorks JavaScript interpreter provides the following objects that can be accessed by a script.

- plate object
- "plates[ ] object" on page 95
- "task object" on page 98
- "InventoryLabware object" on page 89
- "plateDB object" on page 112
- "runset object" on page 123
- "forms object" on page 126

This topic describes the plate object properties and methods.

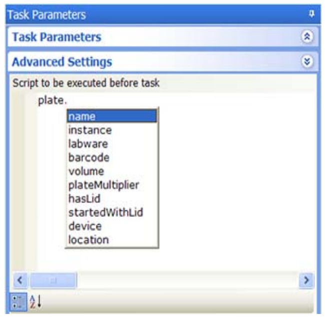
plate provides access to properties of the labware that the current task is operating on. You use the plate object in non-subprocess tasks such as the Place Plate task.

IMPORTANT To access labware properties in a subprocess task, use the plates[ ] object.

Properties

To see the properties of a plate object:

1. Open a protocol and select a task in the protocol area.
2. In the **Advanced Settings** area, type `plate.` The software automatically displays the list of available properties for the plate object.



Instead of displaying the list of plate properties in the Advanced Settings area, you can also list them in the Main Log. Doing so allows you to print the log file and retain a copy of the properties for reference.

To list the plate properties in the Main Log:

1. Open a protocol.
2. Select a task in the **Protocol** area.
3. In the **Task Parameters** area, click **Advanced Settings.**
4. Type the following in the **Script to be executed before task** area:

```
for (x in plate) {
   print ("plate." + x + " = " + plate[x]);
}
```

5. Run the protocol. The plate properties appear in the Main Log.

The following table lists the plate properties.

Property	Data type	Description
plate.name	String	Name of the plate. The property is read-only.
plate.instance	Integer	Plate instance number. The property is read-only.
plate.labware	String	Name of the labware type. The property is read-only.
plate.barcode	Array	Array of four strings where, SOUTH=0, WEST=1, NORTH=2, EAST=3. The property is read-only, and can be changed by the plate.setBarcode() method. Example: `plate.barcode[SOUTH]`

Property	Data type	Description
plate.volume	Array of array of floats	An array consisting of one or more arrays of floating point numbers. The array size depends on the number of wells in the labware (96, 384, or 1536), arranged in row, column format. This property is available only in VWorks software that has the volume-tracking database option. The property is read-only. Example: `plate.volume[row][column]` For 96-well microplates, the maximum row value is 8, and the maximum column value is 12. For 384-well microplates, the maximum row value is 16, the maximum column value is 24. For 1536-well microplates, the maximum row value is 32, the maximum column value is 48.
plate.plateMultiplier	Integer	The number of microplates needed for the run. This number is based on the number of times the protocol is run and the specifications in the Loop and Change Instance tasks. The property is read-only. Examples: • A simple protocol processing Plate_A is run twice. The plateMultiplier is 2. • A protocol loops four times to move contents from the Source Plate to the Destination Plate, and changes instance on the Source Plate only. The protocol is run once. The plateMultiplier is 4 for the Source Plate and 1 for the Destination Plate.
plate.hasLid	Boolean	The indicator that the microplate currently has a lid. The property is read-only.
plate.startedWithLid	Boolean	The indicator that the microplate entered the system with a lid. The property is read-only.
plate.device	String	The device that is processing the plate. The property is read-only.
plate.location	String	The location on the device. The property is read-only.

Methods

Methods are JavaScript functions invoked through an object. The plate object has the following methods, available on systems that have the volume-tracking database option. Use these methods to track microplate-specific data.

Note: You can use the plate.getUserData() method to retrieve information stored by the plates[n].setUserData() method.

Method	Description
plate.setUserData(string *key*, string *value*)	Stores value under the key in a database record associated with this microplate. The data is retained across runs. The method is only available if the database is installed.
plate.getUserData(string *key*)	Returns the 'value' stored earlier using plate.setUserData or plates[n].setUserData. (See "plates[] object" on page 95.) The method is only available if the database is installed.
plate.reportErrorToPlugin(string *error*)	Calls the VWorks Hooks ScriptPlateError method to pass error to the Hooks plugin.
plate.setBarcode(SIDE, string *barcode*)	Manually stores the barcode data in plate.barcode[SIDE]. For example: `plate.setBarcode(WEST, "A1234")`

Related information

For information about...	See...
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined functions	"VWorks-defined functions" on page 87
Other VWorks-defined objects	• "InventoryLabware object" on page 89 • "runset object" on page 123 • "plates[] object" on page 95 • "plateDB object" on page 112 • "task object" on page 98 • "forms object" on page 126
Using JavaScript utilities	"Using JavaScript utilities" on page 128
JavaScript task	"JavaScript" on page 573

For information about...	See...
Startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60

plates[] object

About the plates[] object

The VWorks JavaScript interpreter provides the following objects that can be accessed by a script.

- plates[] object
- "plate object" on page 91
- "task object" on page 98
- "InventoryLabware object" on page 89
- "plateDB object" on page 112
- "runset object" on page 123
- "forms object" on page 126

This topic describes the plates[] object properties and methods.

plates[] is a VWorks-defined array of plate objects that provides access to properties of one or more microplates upon which the current task is operating. You use the plates[] object only for subprocess tasks that reference at least one microplate. For subprocess tasks that do not employ microplates, such as the curly-brace task, the plates[] object is not relevant.

IMPORTANT To access microplate properties in a non-subprocess task, use the plate object.

In a subprocess, the plates[] object is defined only for those tasks that employ a microplate, such as Aspirate or Dispense. Each element in the plates[] object references a single microplate that is referenced by the current instance of the task. The array elements refer to actual process plates, not just instances of a plate. The first element of the plates[] object is the first plate associated with that subprocess task. Therefore, plates[0] in the Aspirate task refers to the Source plate, while in the Dispense task, plates[0] is the Destination plate.

The array elements are in the order that they appear in the task. For example, the Dilute to Final Volume task uses two microplates: Source (plate[0]) and Destination (plate[1]). Suppose the task is in a loop and a Change Instance task changes the Destination microplate each time through the loop. The Source microplate remains plates[0] and the Destination microplate remains plates[1] each time the Dilute to Final Volume task is performed.

Properties

IMPORTANT You can list plates[] properties only for a subprocess task that has at least one microplate associated with it.

You can list the plate properties of each plate[n]. For example, you can list the properties for plate[0] and plate[1] to identify and determine the differences between them.

To list the plates[] properties in the Main Log:

1. Open a protocol.
2. In the **Protocol** area, select a subprocess task that references at least one microplate.
3. In the **Task Parameters** area, click **Advanced Settings.**
4. Type the following in the **Script to be executed before task** area:

```
for (x in plates[n])
{
   print ("plates[n]." + x + " =" + plates[n][x]);
}
```

   Where `n` is the array index.
5. Run the protocol. The plate properties appear in the Main Log.

   The plates[] properties are the same as the plate properties. For the list and description of the plate properties, see "plate object" on page 91.

Methods

Methods are JavaScript functions invoked through an object. The plates[] object has the following methods, available on those systems with the volume-tracking database option. Use these methods to track microplate-specific data.

Note: You can use the plates[n].getUserData() method to retrieve information stored by the plate.setUserData() method.

Method	Description
plates[n].setUserData(string *key*, string *value*)	Stores the *value* under the *key* in a database record associated with this microplate. The data is retained across runs. The method is only available if the database is installed.
plates[n].getUserData(string *key*)	Returns the *value* stored earlier using plates[n].setUserData or plate.setUserData. (See "plate object" on page 91.) The method is only available if the database is installed.
plates[n].reportErrorToPlugin(string *error*)	Calls the VWorks Hooks ScriptPlateError method to pass *error* to the Hooks plugin.
plates[n].setBarcode(SIDE, string *barcode*)	Manually stores the barcode data in plates[n].barcode[SIDE]. For example: `plates[0].setBarcode(WEST, "A1234")`

Related information

For information about...	See...
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined functions	"VWorks-defined functions" on page 87
Other VWorks-defined objects	• "InventoryLabware object" on page 89 • "plate object" on page 91 • "runset object" on page 123 • "plateDB object" on page 112 • "task object" on page 98 • "forms object" on page 126
Using JavaScript utilities	"Using JavaScript utilities" on page 128
JavaScript task	"JavaScript" on page 573
Startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60

task object

About the task object

The VWorks JavaScript interpreter provides the following objects that can be accessed by a script.

- task object
- “plate object” on page 91
- “plates[ ] object” on page 95
- “InventoryLabware object” on page 89
- “plateDB object” on page 112
- “runset object” on page 123
- “forms object” on page 126

This topic describes the task object properties and methods.

task refers to the currently running task. It allows the properties of the task to be accessed using a standard syntax. Depending on which task is running, a different set of properties might be available.

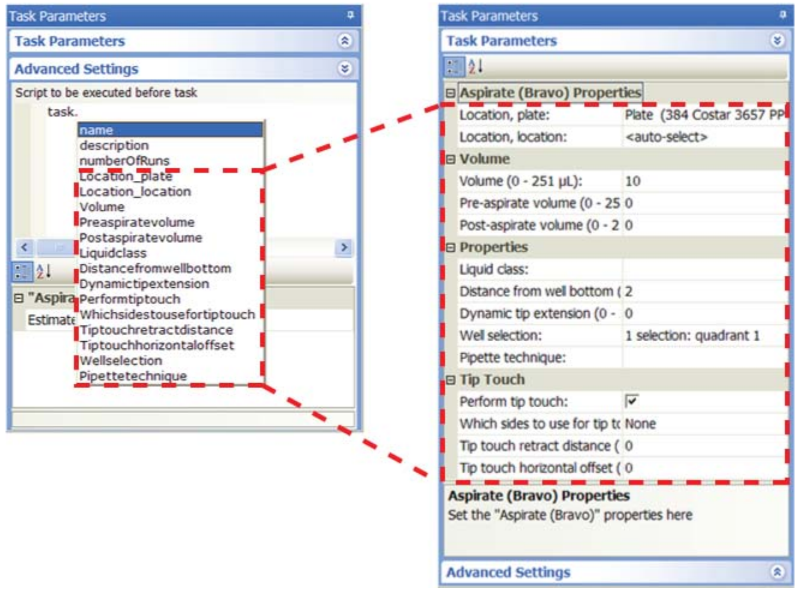
Properties

The properties available for a task correspond to its task parameters in the Task Parameters area.

To see the properties of a task:

1. Open a protocol that contains the task.
2. Select the task in the **Protocol** area.
3. In the **Advanced Settings** area, type `task.` in the **Script to be executed before task** box. The software automatically displays the list of available properties for the task object.

   In the following example, the properties for the Aspirate (Bravo) task is shown in the Advanced Settings area. Notice that they correspond to the parameters in the Task Parameters area.



Instead of displaying the list of task properties in the Advanced Settings area, you can also list them in the Main Log. Doing so allows you to print the log file and retain a hardcopy of the properties for reference.

To list the task properties in the Main Log:

1 Open a protocol that contains the task.

2 Select the task in the **Protocol** area.

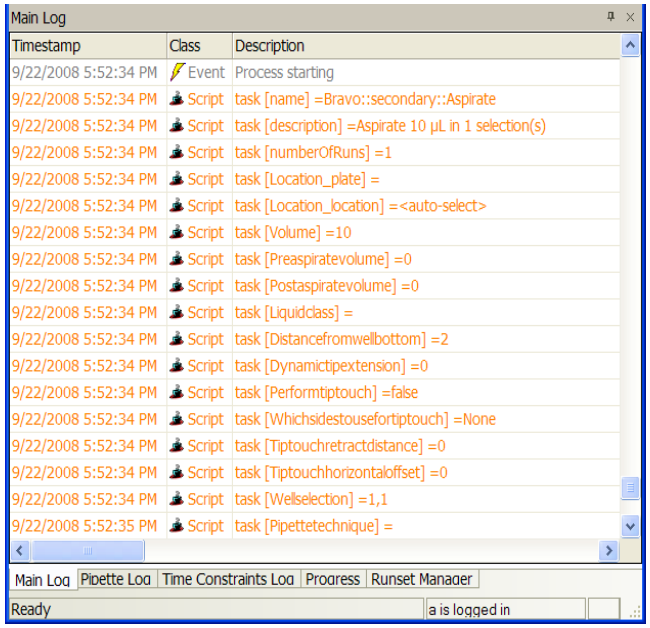
3 In the **Task Parameters** area, click **Advanced Settings.**

4 Type the following in the **Script to be executed before task** area:

```
for (x in task) {
   print("task.[" + x + "] =" + task[x]);
}
```

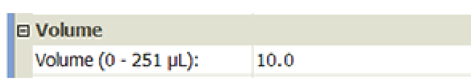
5 Run the protocol. The task properties appear in the Main Log.

The following example shows the properties for the Aspirate (Bravo) task.



Task property data types

A task property can be one of the following data types. You can use the default task parameter value to determine the data type.

Data type	Example task parameter
Array	Aspirate task, well selection  For more information about the array data type, see "Array data types" on page 101.
Boolean	Aspirate task, Perform tip touch option 
Float	Aspirate task, Volume parameter 
Integer	Loop task, Number of times to loop parameter 
String	Name of a process plate 

Array data types

In JavaScript, an array is a built-in object that stores a collection of like values called elements. Each element is accessed by an index value that is enclosed in square brackets. Index values must be non-negative integers.

The following example script declares an array with three elements:

```
var vehicle_type = new Array(3);
vehicle_type[0] = “car”;
vehicle_type[1] = “truck”;
vehicle_type[2] = “van”;
```

In the VWorks software, the array data type applies to properties that specify:

- The sides of a microplate for barcode reading or labeling
- The well selection in liquid-handling tasks, such as the Aspirate task

Sides of a microplate

In properties that specify the sides of a microplate, the array contains four elements, each representing a side of the microplate, for example:

- task.side[SOUTH]
- task.side[EAST]
- task.side[NORTH]
- task.side[WEST]

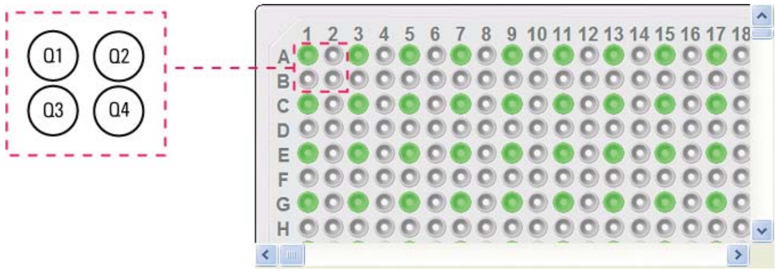
Well selection

In properties that specify well selection, the array identifies the location of wells and quadrants in a microplate. Well locations are represented by a pair of integers that describe its row and column coordinates: [row, column].

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head. The following table lists the types of pipette heads and the number of accessible quadrants in various microplates.

Pipette head channels/ pin tool pins	Microplate	Number of quadrants
96	96-well	1
	384-well	4
	1536-well	16
384	384-well	1
	1536-well	4
1536 (pin tool only)	1536-well	1

The following diagram demonstrates the concept of quadrants. The diagram shows a portion of a 384-well microplate and highlights the four quadrants (Q1, Q2, Q3, and Q4) that are accessible by the A1 tip of a 96-channel pipette head. Notice that the green color highlights all of the quadrant 1 (Q1) wells across the microplate.



When specifying well selection, you provide both the well location and quadrant location. For example, the designation for the four quadrants highlighted in the previous diagram is:

```
[[1,1],[1,2],[2,1],[2,2]]
```

where the Q1 coordinates are [1,1], Q2 coordinates are [1,2], Q3 coordinates are [2,1], and Q4 coordinates are [2,2].

You can also use variables to represent the row and column coordinates, as the following example shows:

```
task.Wellselection = [[disp_row,disp_col]]
```

IMPORTANT Quadrant specifications in the Well Selection dialog box will override task.Wellselection values assigned in the Advanced Settings area.

To cycle through the quadrants in a liquid-handling task, you can use a series of *if* statements as the following example shows.

```
var Dispense_Loop;
if (Dispense_Loop == undefined) {
       Dispense_Loop = 1;
}
if (isNaN(Dispense_Loop)){
   Dispense_Loop = 1;
}
if ((Dispense_Loop < 1)||(Dispense_Loop > 4)){
   Dispense_Loop = 1;
}

if (Dispense_Loop == 1){
   task.Wellselection = [[1,1]];
}
if (Dispense_Loop == 2){
   task.Wellselection = [[1,2]];
}
if (Dispense_Loop == 3){
   task.Wellselection = [[2,1]];
}
if (Dispense_Loop == 4){
   task.Wellselection = [[2,2]];
}
```

```
Dispense_Loop++;
print("Dispensing to quadrant "+task.Wellselection+" of
Destination plate.");
}
```

Methods

The task object methods, and the properties of the methods, are specific to the selected task. The following table lists some of the shared methods:

Method	Description

task.skip()	Skips the current task. Use this function to conditionally run a task, such as in this example which skips the task if the simulator is not running:
	`if (!task.isSimulatorRunning()) {`
	`task.skip();`
	`}`

task.pause()	Pauses the protocol and opens a dialog box that asks you whether you want to continue or abort the run. Use this function if you need to pause the protocol to, for example, replenish the fluid in a static reservoir. You could use the print() function to add a note to the log toolbar describing the action to take when the VWorks software has paused.
task.isSimulatorRunning()	Returns true if this is a simulated run. Has no arguments.
task.repeat()	Schedules the task to be repeated.
task.repeatDelay(int *timeseconds*)	Requests the task to wait the specified time (in seconds), and then repeats. For example, you can use this method to wait for a value to change.
task.setGlobalData(string *name*, string *value*)	Stores *value* in *name* in the Global database.
task.getGlobalData(string *name*)	Returns the value stored earlier using task.setGlobalData.
task.getProtocolName()	Retrieves the full protocol path (string).

task.Headmode method

Available only for SubProcess (Bravo). The following method enables selection of a subset of the barrels in the pipette head:

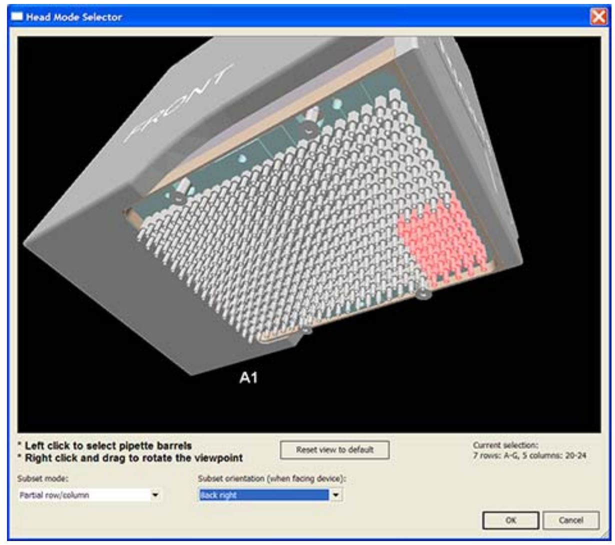
task.Headmode(string *hst,hso,row,col*)

Parameter	Values
hst	Specifies the barrel selection as follows: 0 = all_barrels 1 = column 2 = row 3 = partial_column_row 4 = single_barrel
hso	Specifies the region of the barrel selection, where, 0 = front_right 1 = back_right 2 = back_left 3 = front_left
row	Specifies the row dimensions of the barrel selection, for example, • 96-barrel head, the maximum row value is 8, and the maximum column value is 12. • 384-barrel head, the maximum row value is 16, the maximum column value is 24.
col	Specifies the column dimensions of the barrel selection, for example, • 96-barrel head, the maximum row value is 8, and the maximum column value is 12. • 384-barrel head, the maximum row value is 16, the maximum column value is 24.

The following examples show how to use task.Headmode.

<table>
<tr><th>Barrel selection options</th><th>Example task.Headmode script and comparable view in the Head Mode Selector dialog box</th></tr>
<tr><td>Select partial row/column</td><td>Selecting 35 barrels (7-by-5 region)

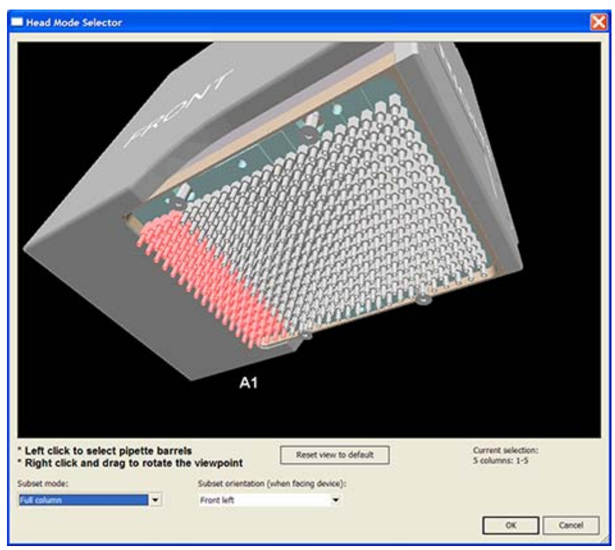
`task.Headmode = “3,1,7,5”`



</td></tr>
<tr><td>Select column or columns</td><td>Selecting 5 columns

Note: Row specified is ignored.

`task.Headmode = “1,2,1,5”`

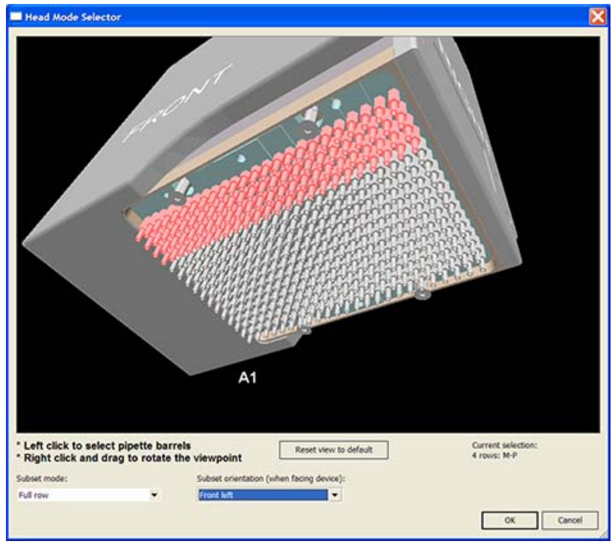


</td></tr>
</table>

<table>
<tr><th>Barrel selection options</th><th>Example task.Headmode script and comparable view in the Head Mode Selector dialog box</th></tr>
<tr><td>Select row or rows</td><td>Selecting 4 rows

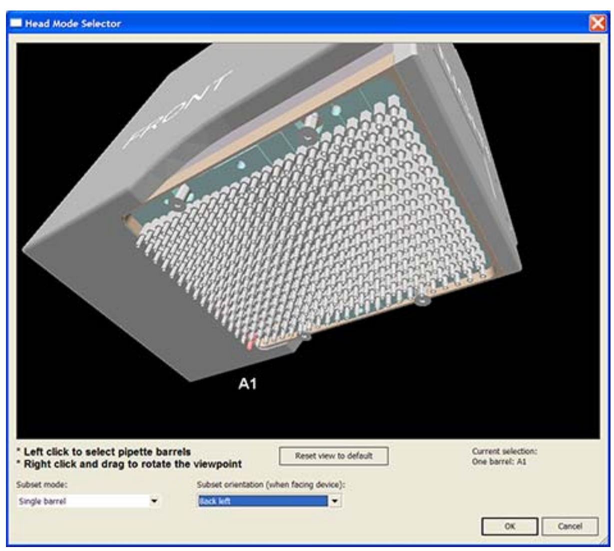
Note: Column specified is ignored.

<code>task.HeadMode = "2,3,4,1"</code>


</td></tr>
<tr><td>Select single barrel</td><td>Selecting single barrel in the back left

Note: Row and column specified are ignored; region is always 1-by-1.

<code>task.HeadMode = "4,2,1,1"</code>


</td></tr>
</table>

Related information

For information about...	See...
Writing JavaScript for the Print and Apply task	"Using JavaScript with the CentrifugeAuto task" on page 108
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined functions	"VWorks-defined functions" on page 87
Other VWorks-defined objects	• "InventoryLabware object" on page 89 • "plate object" on page 91 • "plates[] object" on page 95 • "plateDB object" on page 112 • "runset object" on page 123 • "forms object" on page 126
Using JavaScript utilities	"Using JavaScript utilities" on page 128
JavaScript task	"JavaScript" on page 573
Startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60
Specifying a quadrant pattern in the Well Selection dialog box	• "Aspirate (Bravo, Vertical Pipetting Station)" on page 392 • "Dispense (Bravo, Vertical Pipetting Station)" on page 451 • "Mix (Bravo, Vertical Pipetting Station)" on page 497

Using JavaScript with the CentrifugeAuto task

About this topic

This topic provides two examples of using JavaScript to modify the implementation of the CentrifugeAuto task. For more information about using the CentrifugeAuto task, see “CentrifugeAuto” on page 286.

Counterweight selection

When using the Auto managed counterweight mode, you can use JavaScript to select the counterweight, for example:

```
if(place.instance%2 == 1)
{
task.Counterweight = “CW1”;
}
else
{
task.Counterweight = “CW2”;
}
```

Reduce number of spins for one plate

When running an odd number of microplates with the CentrifugeAuto task in Use 2 protocol plates mode, the second to last plate is spun twice so that the last microplate is paired. Using JavaScript, you can direct the protocol to switch to Auto managed counterweight mode for the last spin, for example:

```
if((plate.instance == task.numberOfRuns) &&
(plate.instance%2 == 1))
{
task.skip();
}
if((plate.instance == task.numberOfRuns) &&
(plate.instance%2 ==1))
{
task.Counterweight = “CW1”
}
else
{
task.skip();
}
```

Related information

For information about...	See...
CentrfiugeAuto task	"CentrifugeAuto" on page 286
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined functions	"VWorks-defined functions" on page 87
Using JavaScript utilities	"Using JavaScript utilities" on page 128

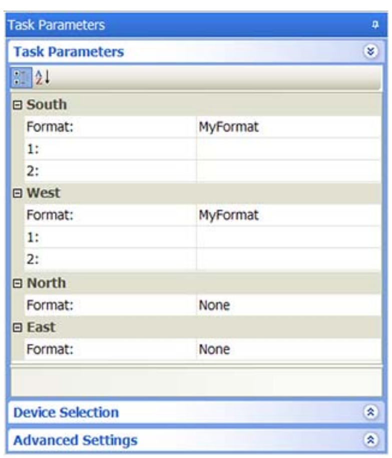
About scripting the Print and Apply task

Print and Apply task parameter exceptions

Unlike other tasks, the Print and Apply task parameters have the following exceptions:

- The number of parameters can vary, because each labeling device has a different number of formats, and each format has a different number of fields. (In the software, the contents of each field are specified in the Field Composer dialog box.)
- The software uses the destination side of the microplate (south, west, north, or east) as a prefix for the parameter names:

  In the following example, MyFormat is specified for the south and west sides of a microplate for fields 1 and 2. The software will append a prefix to each field name to indicate the side. The new field names will be South_1, South_2, West_1, and West_2.



Scripting what to print (label content)

When scripting the task parameters for the Print and Apply task, the field parameter must be assigned to a string that encodes what is to be printed. The following table lists the task parameters and the corresponding string options.

Note: When in doubt about available task parameters, add the following script into the task and run: `for(x in task){print(x + "::" + task[x])}` This script prints all the current task parameters.

Parameter	JavaScript string

Date	`"[DATE:enum1]"`
	Where `enum1`:
	0 – Use System Format
	1 – YYYY/MM/DD
	2 – DD/MM/YYYY
	3 – MM/DD/YYYY
	4 – YY/MM/DD
	5 – DD/MM/YY
	6 – MM/DD/YY
	Example:
	`task.East_1 = "[DATE:0]"`

Time	`"[TIME:enum1]"`
	Where `enum1`:
	0–12 (12-hour time AM/PM)
	1–24 (24-hour time)
	Example:
	`task.East_1 = "[TIME:0]"`

Counter	`"[COUNTER:StartAt:IncBy:TotalNumberOfDigits:IncEveryNPlates:enum1]"`
	Where `enum1`:
	0 – Numeric (0–9)
	1 – Alphanumeric (0–Z)
	Example:
	`task.East_1 = "[COUNTER:0:1:3:2:1]"`

File With Lookup	`"[FILEWLOOKUP:filename:keyCol:ValueCol]"`
	Example:
	`task.East_1 = "[FILEWLOOKUP:c:/temp.csv:1:1]"`

File Start at row/col and increment row	`"[FILE:filename:startRow:startCol]"`
	Example:
	`task.East_1 = "[FILE:c:/temp.csv:1:1]"`

Parameter	JavaScript string

Static Text	`"[STATIC:staticText]"`
	Example:
	`var my_string = "hello world";`
	`task.East_1 = "[STATIC:"+ my_string +"]";`

CopyBarcode from a different side	`task.East_1 = "[BARCODE:copyFromSideEnum1]"`
	Where `copyFromSideEnum1`:
	0 – south
	1 – west
	2 – north
	3 – east
	Example:
	`task.East_1 = "[BARCODE:2]"`

Guidelines for scripting where to print the barcode label

About scripting which field to print to

To print on a single side or more than one side, use the following syntax:

```
task.East_1
```

where, `East` is the side of the labware and `1` is field 1.

About scripting the format

Although it is possible to script the format, you must ensure that the scripted format has the same number of fields as the format specified in the protocol.

```
task.West_Format=3
//script the format

task.West_3 = "[FILEWLOOKUP:C:\\bcf.csv:1:3]"
//script field 3 of format 3
```

Note: To read and write the field task parameters, the values used are pre-evaluation. For example, to print the current time, you would set the task parameter for field 1 to `[TIME:0]`. If you print this from JavaScript using `print(task.East_1)`, the result will be `[TIME:0]`, not as the actual time.

Related information

For information about...	See...
Print and Apply task	"Print and Apply" on page 318
Label formats	*Microplate Barcode Labeler User Guide*
Field contents	"Print and Apply" on page 318
Using JavaScript in the VWorks software	"Using JavaScript" on page 84

plateDB object

About the plateDB object

The VWorks JavaScript interpreter provides the following objects that can be accessed by a script.

- plateDB object
- "InventoryLabware object" on page 89
- "plate object" on page 91
- "plates[ ] object" on page 95
- "task object" on page 98
- "runset object" on page 123
- "forms object" on page 126

This topic describes the plateDB object properties and methods.

plateDB enables control of the database through JavaScript. You can use the plateDB object to create plate groups, add plates to existing plate groups (if they exist in a location group), remove plates from plate groups, and delete plate groups.

Properties

The plateDB object has no properties.

Methods

Methods are JavaScript functions invoked through an object. The plateDB object has the following methods, which are available on systems with the inventory database option.

Syntax:

plateDB.*method( )*

Note: A common set of return codes are used in the applicable methods. However, based on relevance, an applicable method might use only a subset of the return codes.

Method	Description
createPlateGroup(string *plateGroup*)	Creates a new plate group. Returns true on success, or returns false on failure. The method also prints to the log the reason for failure. Example: `plateDB.createPlateGroup("myplate group")`

Method	Description
deletePlateGroup(string *plateGroup*)	Deletes a plate group. Returns true on success, or returns false on failure. The method also prints to the log the reason for failure. Example: `plateDB.deletePlateGroup("myplate group")`
addPlateToGroup(int *side*, string *barcode,* string *locationGroup*, string *plateGroup*)	Adds a labware that exists in a location group to a plate group. Returns true if successful, or returns false upon failure. The method also prints the reason for failure to the log. *Note:* The plate must exist in a locationGroup, and the given barcode must be on the specified side of the plate. Example: `plateDB.addPlateToGroup(WEST, "BC01", "locationGroup", "plateGroup")`
deleteAllPlatesFromGroup(string *plateGroup*)	Deletes all labware from a specified plate group. Returns false upon failure, and prints the reason for failure to the log. *Note:* The plate group must exist. Example: `plateDB.deleteAllPlatesFromGroup ("plateGroupName")`

Method	Description
importCsvToInventory(string *CSVFilepath*, bool *OverwriteInventory*)	Imports the contents of the specified CSV file into the inventory database. You have the option of overwriting the existing data in the database during the import process: true—Overwrite the existing data. false—Skip the current labware. The method returns the following: –1—Unknown error. 0—Method is successful. 1—Location is occupied by another labware. 2—Location is empty. 3—Device does not exist. 4—Cassette does not exist. 5—Slot does not exist. 6—CSV file does not exist in the specified directory. Example: `plateDB.importCsvToInventory ("C:\VWorks Workspace\labware.csv", true)`

Method	Description

importLabwareToInventory(array *Labware*, bool *OverwriteInventory*)	Imports the array of labware data into the inventory database. The InventoryLabware object is passed to this method and is an element in the array.
	Each array element (labware) must have the following information:
	• Device
	• Cassette number
	• Slot number
	• Labware type
	You have the option of overwriting the existing data in the database during the import process:
	true—Overwrite the existing data.
	false—Skip the current labware.
	The method returns the following:
	–1—Unknown error.
	0—Method is successful.
	1—Location is occupied by another labware.
	2—Location is empty.
	3—Device does not exist.
	4—Cassette does not exist.
	5—Slot does not exist.
	6—File does not exist.
	Example:
	`var LabwareArray = new Array();`
	`var Labwareone = new InventoryLabware();`
	`Labwareone.device = "Agilent Plate Hub Carousel";`
	`Labwareone.cassette = "1";`
	`Labwareone.slot = "1";`
	`Labwareone.type = "384 Greiner ";`
	`Labwareone.westbc = "west";`
	`LabwareArray[0] = Labwareone;`
	`plateDB.importLabwareToInventory(LabwareArray,true);`

Method	Description
clearInventory()	Deletes all labware data in the inventory database. The method returns the following: –1—Unknown error. 0—Method is successful. 1—Location is occupied by another labware. 2—Location is empty. 3—Device does not exist. 4—Cassette does not exist. 5—Slot does not exist. 6—File does not exist. Example: `plateDB.clearInventory();`

clearStorageLocation(string *GroupName*)	Deletes labware in the specified location group or plate group.
	The method returns the following:
	–1—Unknown error.
	0—Method is successful.
	1—Location is occupied by another labware.
	2—Location is empty.
	3—Device does not exist.
	4—Cassette does not exist.
	5—Slot does not exist.
	6—File does not exist.
	Example:
	`plateDB.clearStorageLocation("MyPlateGroup");`
	`plateDB.clearStorageLocation("MyLocationGroup");`

Method	Description
clearAbandonedLabware( )	Deletes labware that has become abandoned. Labware becomes abandoned when it moves out of storage and is somewhere in the system when a deadlock occurs or when a run is aborted. The method returns the following: –1—Unknown error. 0—Method is successful. 1—Location is occupied by another labware. 2—Location is empty. 3—Device does not exist. 4—Cassette does not exist. 5—Slot does not exist. 6—File does not exist. Example: `plateDB.clearAbandonedLabware();`
clearAllLabwareBarcodes( )	Deletes all labware barcodes in the inventory database. Other labware information is retained. The method returns the following: –1—Unknown error. 0—Method is successful. 1—Location is occupied by another labware. 2—Location is empty. 3—Device does not exist. 4—Cassette does not exist. 5—Slot does not exist. 6—File does not exist. Example: `plateDB.clearAllLabwareBarcodes();`

Method	Description

addLabwareToLocation(string *DeviceName*, string *CassetteNumber*, string *SlotNumber*, string *Type*, bool *Overwrite*)	Adds labware to the specified storage location in the inventory database.
	The method returns the following:
	–1—Unknown error.
	0—Method is successful.
	1—Location is occupied by another labware.
	2—Location is empty.
	3—Device does not exist.
	4—Cassette does not exist.
	5—Slot does not exist.
	6—File does not exist.
	Example:
	`plateDB.addLabwareToLocation("Agilent Plate Hub Carousel","1","1","384 Greiner",true);`
	`plateDB.addLabwareToLocation("Agilent Plate Hub Carousel","all","1","384 Greiner",true);`
	`plateDB.addLabwareToLocation("Agilent Plate Hub Carousel","1","all","384 Greiner",true);`
	`plateDB.addLabwareToLocation("Agilent Plate Hub Carousel","all","all","384 Greiner",true);`

Method	Description

deleteLabwareAtLocation(string *DeviceName*, string *CassetteNumber*, string *SlotNumber*)	Deletes the labware information at the specified storage location.
	The method returns the following:
	–1—Unknown error.
	0—Method is successful.
	1—Location is occupied by another labware.
	2—Location is empty.
	3—Device does not exist.
	4—Cassette does not exist.
	5—Slot does not exist.
	6—File does not exist.
	Example:
	`plateDB.deleteLabwareAtLocation("Agilent Plate Hub Carousel","1","1");`
	`plateDB.deleteLabwareAtLocation("Agilent Plate Hub Carousel","all","1");`
	`plateDB.deleteLabwareAtLocation("Agilent Plate Hub Carousel","1","all");`
	`plateDB.deleteLabwareAtLocation("Agilent Plate Hub Carousel","all","all");`

getLocationByBarcode(string *Barcode*)	Obtains the location of a labware using the supplied barcode.
	Returns an array of labware (InventoryLabware objects) if the method is successful. Otherwise, the method returns a null array.
	Example:
	`var LabwareArray = new Array();`
	`LabwareArray = plateDB.getLocationByBarcode("barcode");`

Method	Description
exportDatabase(string *FileName*)	Exports the data in the inventory database to a SQL file. The method returns the following: –1—Unknown error. 0—Method is successful. 1—Location is occupied by another labware. 2—Location is empty. 3—Device does not exist. 4—Cassette does not exist. 5—Slot does not exist. 6—File does not exist. Example: `plateDB.exportDatabase("C:/VWorks Workspace\exportedfile.sql")`
importDatabase(string *FileName*)	Imports the data in the specified SQL file into the inventory database. The method returns the following: –1—Unknown error. 0—Method is successful. 1—Location is occupied by another labware. 2—Location is empty. 3—Device does not exist. 4—Cassette does not exist. 5—Slot does not exist. 6—File does not exist. Example: `plateDB.importDatabase("C:/VWorks Workspace\importedfile.sql")`
enumeratePlateGroup(string *plateGroup*)	Returns the plate IDs from the plate database for this plate group. Returns an array of database plate IDs. If the method fails, the reason for failure prints to the log. *Note:* The plate group must exist. Example: `plateDB.enumeratePlateGroup("plateGroupName")` See additional example in getPlateBarcode method, below.

Method	Description

getPlateBarcode(int *side*, array *plateID*)	Returns the plate barcode (string) for the specified plate ID and the side of the plate.
	If the method fails, the reason for failure prints to the log.
	Example:
	`var plateIDs = plateDB.enumeratePlateGroup("plateGroupName")`
	`for(var plateID in plateIDs){`
	`    print(plateDB.getPlateBarcode(WEST, plateIDs[plateID]))`
	`}`

enumerateListOfGroups(int *groupType*)	Returns the list of plate group names for the list type *groupType*, where 0 = both types (plate and location groups) 1 = plate groups 2 = location groups The method returns an array of strings corresponding to the group names. If the method fails, the reason for failure prints to the log. Example: `var group_array = plateDB.enumerateListOfGroups(0)`
getLabwareInfo(string *labwareName*)	Returns information about the specified labware, such as the number of wells (NumberOfWells=N). If the labware specified does not exist in the labware definition, the method returns an empty string. Example: `plateDB.getLabwareInfo("96 Greiner Polystyrene Clear Round Well, Flat Bottom")`

Related information

For information about...	See...
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined functions	"VWorks-defined functions" on page 87

For information about...	See...
Other VWorks-defined objects	• “InventoryLabware object” on page 89 • “plate object” on page 91 • “plates[ ] object” on page 95 • “runset object” on page 123 • “task object” on page 98 • “forms object” on page 126
Using JavaScript utilities	“Using JavaScript utilities” on page 128
JavaScript task	“JavaScript” on page 573
Startup and cleanup protocols	“Setting up Startup and Cleanup Protocol processes” on page 60

runset object

About the runset object

The VWorks JavaScript interpreter provides the following objects that can be accessed by a script.

- runset object
- “plate object” on page 91
- “plates[ ] object” on page 95
- “task object” on page 98
- “InventoryLabware object” on page 89
- “plateDB object” on page 112
- “forms object” on page 126

This topic describes the runset object properties and methods.

runset enables control of the Runset Manager through JavaScript. You can use the runset object to clear the runs from a runset or append protocols to a runset. You can also query the Runset Manager and retrieve various fields of interest, such as barcodes. For more information on runsets, see “Managing runsets” on page 243.

Note: If you are using the VWorks Watcher, the runset object can be used in a script that is called by Watcher to create runsets automatically. For details, see “Setting up and using the Watcher tool” on page 685.

Properties

The runset object has no properties.

Methods

Methods are JavaScript functions invoked through an object. The runset object has the following methods.

Method	Description
clear()	Clears all of the entries in the runset manager, except for currently executing protocols, which are not affected. Example: `runset.clear()`

Method	Description
appendProtocolFileToRunset (string *ProtocolPath*, int *RunTimes*, string *ProtocolNotes*, string *formToUse*)	Appends the contents of the protocol file at *ProtocolPath* to the Runset Manager. Specifies the number of *RunTimes*, and provides any *ProtocolNotes*. Optionally, specifies the form to be associated with the protocol once the protocol starts running. The *formToUse* argument is for use with a scripted pushbutton in a form, where the script contains a call to appendProtocolFileToRunset(). *formToUse* must specify the file name (not the path) of the form that contains the pushbutton. The third argument, *ProtocolNotes*, must be an empty string in this case. Returns true if successful or false upon failure. If a failure occurs, for example, the .pro file is missing, errors are written to the log.
Example: `runset.appendProtocolFileToRunset("c:/some dir/someprotocol.pro", 10,  "", "FormName1.VWForm");`	
appendRunsetFileToRunset (string *runsetPath*, string *formToUse*)	Appends the contents of a runset file, located at *runsetPath*, to the Runset Manager. Optionally, specifies the form to be associated with the runset. *formToUse* is for a scripted pushbutton in a form, and the argument must specify the file name (not the path) of the form that contains the pushbutton. Returns true if successful or false upon failure. If a failure occurs, for example, the .rst file is missing, errors are written to the log.

Examples:
`runset.appendRunsetFileToRunset("c:/runset.rst");`
`runset.appendRunsetFileToRunset("c:/runset.rst", "FormName1.VWForm");`	

openRunsetFile(string *runsetPath*, string *formToUse*)	Opens the runset located at *runsetPath*. Optionally, opens the form to use to run the runset. *formToUse* is for a scripted pushbutton in a form, and the argument must specify the file name (not the path) of the form that contains the pushbutton. Returns true if successful or false upon failure. If a failure occurs, for example, the .rst file is missing, errors are written to the log.

Examples:
`runset.openRunsetFile("c:/temp/my_runset.rst");`
`runset.openRunsetFile("c:/temp/my_runset.rst", "FormName1.VWForm");`	

Method	Description
getRunsetXML()	Returns the XML contents of a runset file (.rst).

Example for getRunsetXML()

```
// Note: In this example, runsetParser is a hypothetical
JavaScript function.
var runsetXML = runset.getRunsetXML()
var runsetParser = new runsetParser(runsetXML)
var numRunningProtocols =
runsetParser.GetNumberOfRunningProtocols()


for(  var running_protocol_idx  = 0;
      running_protocol_idx < numRunningProtocols;
      running_protocol_idx++){


   print(runsetParser.GetRunningProtocol(running_protocol_i
   dx).GetFilename())
   print(runsetParser.GetRunningProtocol(running_protocol_i
   dx).GetRunsetNotes())
   //The runset notes might include items such as barcodes.
}
```

Related information

For information about...	See...
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
VWorks-defined functions	"VWorks-defined functions" on page 87
Other VWorks-defined objects	• "InventoryLabware object" on page 89 • "plate object" on page 91 • "plates[] object" on page 95 • "plateDB object" on page 112 • "task object" on page 98 • "forms object" on page 126
Using JavaScript utilities	"Using JavaScript utilities" on page 128
JavaScript task	"JavaScript" on page 573
Startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60
VWorks Watcher tool	"Setting up and using the Watcher tool" on page 685

For information about...	See...
How to create a runset	"Managing runsets" on page 243

forms object

About the forms object

The VWorks JavaScript interpreter provides the following objects that can be accessed by a script.

- forms object
- "plate object" on page 91
- "plates[ ] object" on page 95
- "task object" on page 98
- "InventoryLabware object" on page 89
- "plateDB object" on page 112
- "runset object" on page 123

This topic describes the forms object usage.

forms provides the same function as the Toggle Full Screen button, which is one of the specialized buttons that you can include on a form that runs a protocol. For more information on forms, see "Creating protocol forms for operators" on page 153.

Properties

The forms object has no properties.

Methods

Methods are JavaScript functions invoked through an object. The forms object has the following method.

Method	Description

setFullScreen(Boolean *True/ False*)	Changes the display of the VWorks window between a full-screen view of a protocol form and a normal tabbed view.
	Example:
	`forms.setFullScreen(true)`
	`forms.setFullScreen(false)`

Related information

For information about...	See...
Designing forms to run protocols	"Creating protocol forms for operators" on page 153
VWorks-defined functions	"VWorks-defined functions" on page 87

Using start and finish protocol scripts

About start and finish scripts

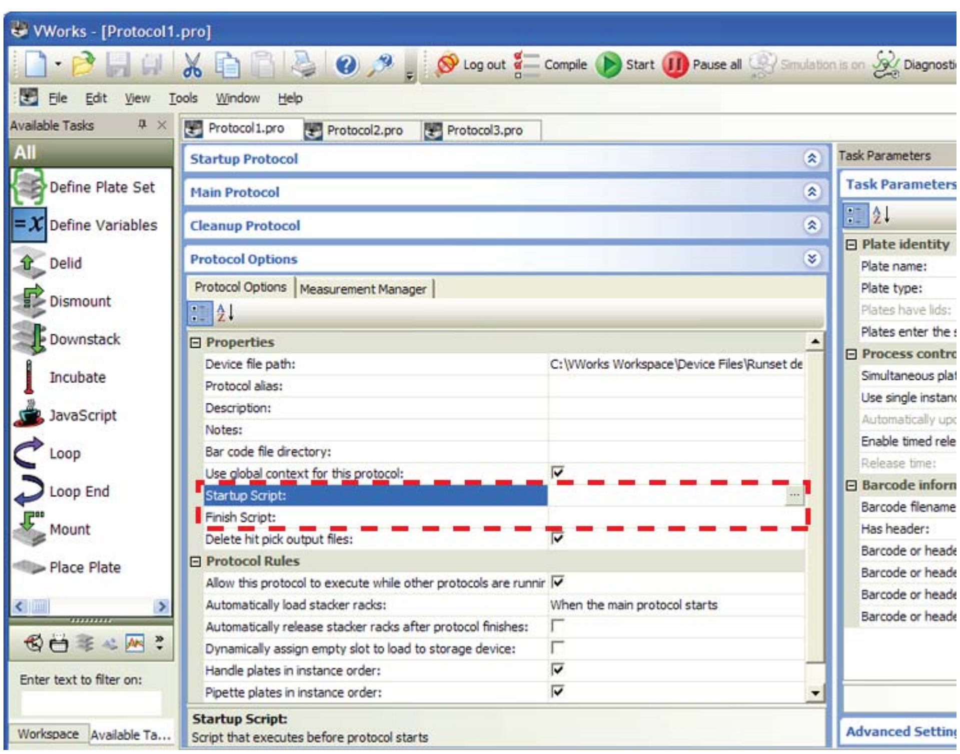
A startup protocol script is a JavaScript that runs before the Startup Protocol begins. A finish protocol script is a JavaScript that runs after the Cleanup Protocol finishes. For example, you can use an open () statement to open and load a file.

Procedure

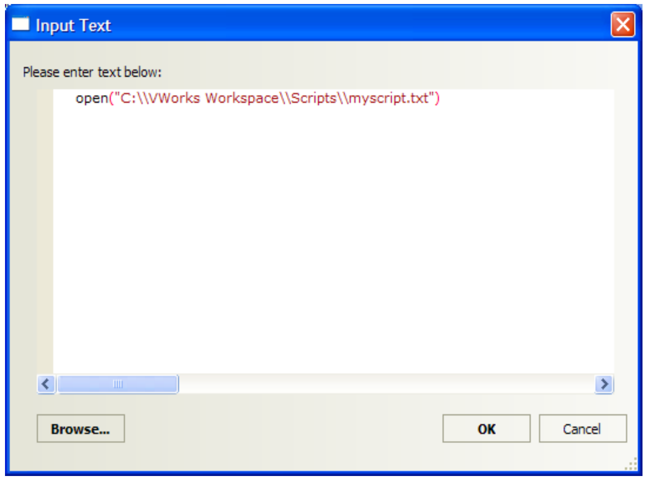
To add start or finish scripts:

1 Open the protocol file, and then click **Protocol Options.**

2 Click the **Startup Script** or the **Finish Script** field, and then click the ... button that appears.



The Input Text dialog box appears.



3 Do one of the following:

- Type the JavaScript code in the box.
- Click **Browse**. In the **Open** dialog box, select a file that contains the JavaScript code, and click **Open**.

4 Click **OK** to save the changes and return to the VWorks window.

Related information

For information about...	See...
Opening protocols	"Opening a protocol" on page 213
Setting protocol options	"Setting protocol options" on page 30
Using script variables directly in task parameters	"Using simple variables" on page 77
Using JavaScript in the VWorks software	"Using JavaScript" on page 84
Startup and cleanup protocols	"Setting up Startup and Cleanup Protocol processes" on page 60

Using JavaScript utilities

Utilities description

Two JavaScript utilities are available:

- *ActiveX Wrapper*. Allows you to create an ActiveX object and use the associated ActiveX methods.

- *File Object*. Allows you to create a file object to read from and write to a file.

You can write JavaScript using these utilities in the VWorks Advanced Settings area. For instructions on how to add the JavaScript task and display the Advanced Settings area, see "Where to write JavaScript" on page 85.

ActiveX Wrapper

ActiveX controls are software components that allow different software products to interact. For example, if you want to use the VWorks software to control a third-party device, you can use the device's ActiveX control to invoke the device's operations.

The ActiveX Wrapper utility in the VWorks software allows you to use another product's ActiveX control to invoke the product's operations. Make sure you install the product's ActiveX control software before you run the JavaScript.

To use the ActiveX utility, you need to:

1. Create an ActiveX object to reference the ActiveX control.
2. Call the associated ActiveX methods to invoke the ActiveX operations.
3. Use the Set and Get methods to access the ActiveX properties.

Create an ActiveX object

To create an ActiveX object:

In the Advanced Settings area, type the JavaScript code to create an ActiveX object.

For example, if the ActiveX control PROG_ID is PlateLocCtrl.2, you can create the object as follows:

```
var ocx
if( ocx == undefined){
ocx = new ActiveX( "PLATELOC.PlateLocCtrl.2");
}
```

The var statement declares a JavaScript variable. In this example, the variable is `ocx`.

The if statement prevents the software from creating the ActiveX multiple times if the script is run repeatedly.

The `ocx = new ActiveX` statement passes the PlateLoc PROG_ID to the ActiveX object generator. Using the ID, the generator calls the CreateInstance API. The resulting ActiveX object is then wrapped in the scripting layer that translates arguments and returns values that are understood by both the PlateLoc Sealer and the VWorks software.

Calling the ActiveX methods

To call the ActiveX methods:

Call the methods using the following syntax:

```
objectname.method
```

For example, if you want to call the AboutBox() method, you can type the following:

```
ocx.AboutBox()
```

To list the available ActiveX methods, use the following JavaScript statements:

```
for( x in ocx.members)
```

```
print( x)
```

Accessing the ActiveX properties

To access the ActiveX properties:

Use the Set or Get method and the following syntax:

- comm.set("property_name", value)
- comm.get("property_name")

For example, if you want to access the ActiveConnection property of an ADO command object, you can type the following:

```
var db = new ActiveX( "ADODB.Connection")
var comm = new ActiveX( "ADODB.Command")
comm.set("ActiveConnection", db)
```

The first `var` statement creates an ADO object and assigns it to a variable named db.

The second `var` statement creates an ADO command object and assigns it to a variable named comm.

The `comm.set` statement sets the ActiveConnection property to the connection object in the db variable.

File Object

The File Object utility allows you to create a file object so that you can read from and write to a file.

To use the File Object utility:

1. Create the file object.
2. Call the desired file object methods:
   - Open()
   - Close()
   - Read()
   - Write()
   - IsOpen()
   - Exists()
   - Delete()

IMPORTANT The JavaScript language is case-sensitive. Make sure you use the correct upper- and lower-case letters when calling the methods.

Creating a file object

To create a file object:

In the Advanced Settings area, type the following JavaScript code:

```
var fileobjectname
if( fileobjectname == undefined){
fileobjectname = new File()
}
```

Note: fileobjectname is the name of the file object you want to create.

Calling the Open() method

To call the Open() method:

Type the following JavaScript code:

```
fileobjectname.Open( "filepath", 0, 0)
```

`fileobjectname` is the name of the file object you created.

`filepath` (the first argument) is the location of the file you are creating. For example, you can type `c:\\fileobjectname.txt`.

`0` (the second argument) specifies how new information will be added to the file. 0 adds new information after the existing information. A non-zero value erases the existing file contents before adding the new information. If you do not specify this argument, the system will use the default value of 0.

`0` (the third argument) specifies how the line endings in binary files will be translated. 0 translates line endings to a carriage return followed by a line feed. 1 does not translate the existing line ending. If you do not specify this argument, the system will use the default value of 0.

Calling the Close() method

To call the Close() method:

Type the following JavaScript code:

```
fileobjectname.Close()
```

`fileobjectname` is the name of the file object you created.

The Close method closes the file and releases any locks on the file so that other software can access it.

Calling the Read() method

To call the Read() method:

Type the following JavaScript code:

```
var result = fileobjectname.Read()
```

The Read() method returns the entire file contents as a string into the variable called result. Although line-by-line reading is not available, you can use built-in JavaScript string methods to parse the file.

If another process is concurrently adding information to the file, later calls to the Read method will read the newly added information.

Calling the Write() method

To call the Write() method:

Type the following JavaScript code:

```
fileobjectname.Write("writeoutput" + "\n")
```

`fileobjectname` is the name of the file object you created.

`writeoutput` is the string you want to add to the file.

`\n` adds a new line at the end of the string.

Calling the IsOpen() method

To call the IsOpen() method:

Type the following JavaScript code:

```
var open = fileobjectname.IsOpen()
```

The `var` statement checks to see if the file opening call was successful.

Calling the Exists() method

To call the Exists() method:

Type the following JavaScript code:

```
var exists = fileobjectname.Exists(“filepath”)
```

`filepath` is the location of the file you are checking.

The `var` statement checks to see if the file exists in the specified folder and returns true if the file is present.

Calling the Delete() method

To call the Delete() method:

Type the following JavaScript code:

```
fileobjectname.Delete("filepath")
```

`filepath` is the location of the file you are deleting. For example, you can type `c:\\fileobjectname.txt`.

Related information

For information about...	See...
JavaScript language	Mozilla Developer Center
Using JavaScript in the VWorks software	“Using JavaScript” on page 84
Using script variables directly in task parameters	“Using simple variables” on page 77
JavaScript task description	“JavaScript” on page 573
Startup and cleanup protocols	“Setting up Startup and Cleanup Protocol processes” on page 60
Setting protocol options	“Setting protocol options” on page 30

Creating new protocol tasks using the JavaScript Wrapper

See the JavaScript Wrapper Application Note.

Note: If you have trouble opening the link, go to http://www.chem.agilent.com/Library/usermanuals/Public/G5415-90059_R_00_JSWrapper_S_EN.pdf

4
Using macros to create protocols

This chapter describes the VWorks macros that you can use to help simplify and expedite the protocol writing process. You must have VWorks Administrator or Technician access to create macros, and you should have an understanding of how to create VWorks protocols.

This chapter contains the following topics:

About protocol macros and the macro library

About this topic

This topic describes the VWorks macros and the macro library that you can create to help simplify and expedite the protocol writing process. To create macros and protocols, you must have VWorks Technician- or Administrator-level access.

Macros defined

A VWorks macro is a collection of protocol tasks and associated task parameters grouped together in an abbreviated form. You can insert the macro easily at other places within the same protocol or throughout other protocols where the same task sequence is required.

A macro eliminates the need to recreate a repeated task sequence within the same protocols or subsequent protocols that you write. Therefore, protocol writing can be easier, faster, and less error-prone.

Suppose you want to create a protocol that has multiple wash routines, where each wash routine consists of a sequence of Tips On, Aspirate, Dispense, and Tips Off tasks. You could do the following using a macro:

1. Create the wash routine task sequence.
2. Highlight the task sequence and add it as a macro to the macro library.
3. Set custom variables at the macro level, such as the volume or number of times to wash, so that the parameters could be scripted before running the protocol.

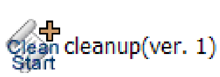
   For instance, the variables could be exposed on a protocol form that the operator is permitted to change at run time.
4. Insert the macro at the other points in the protocol where the wash routine is required.

For the subsequent protocols that you create, which require the wash routine, you can insert the macro from the macro library. For each instance of the macro, you should verify that the parameter values are set correctly, such as Location, plate and Location, location.

Later, if you want to modify the macro, the change can be replicated to all the other instances of the macro in an open protocol.

Types of macros

You can create the following types of macros in the VWorks software:

Macro	Description
Startup or cleanup macro 	Available for use in the Startup Protocol area and the Cleanup Protocol area.

Macro	Description
Main process macro Main(ver. 1)	Available for use in the Main Protocol area only within the main process.
Subprocess macro subprocess(ver. 1)	Available for use in the Main Protocol area only within a subprocess.

Macros have the following requirements:

- All tasks in the macro must be contiguous in the protocol.
- A macro can contain subprocess or process tasks, but not both.
- A macro cannot contain another macro.

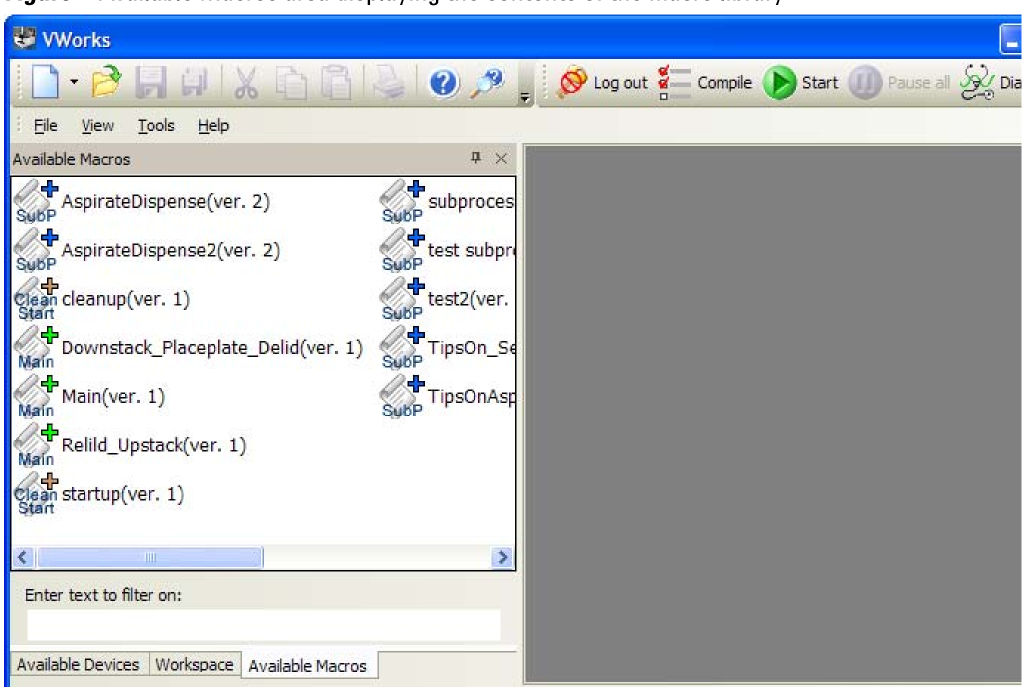
Macro library

The macro definitions are stored in a macro library. Although different versions of a given macro may be in use in the protocols, the software maintains only one official version in the macro library. The VWorks software uses an .mlb file to store the macro library. By default, the file path is as follows:

...\VWorks Workspace\VWorks\MacroLibrary.mlb

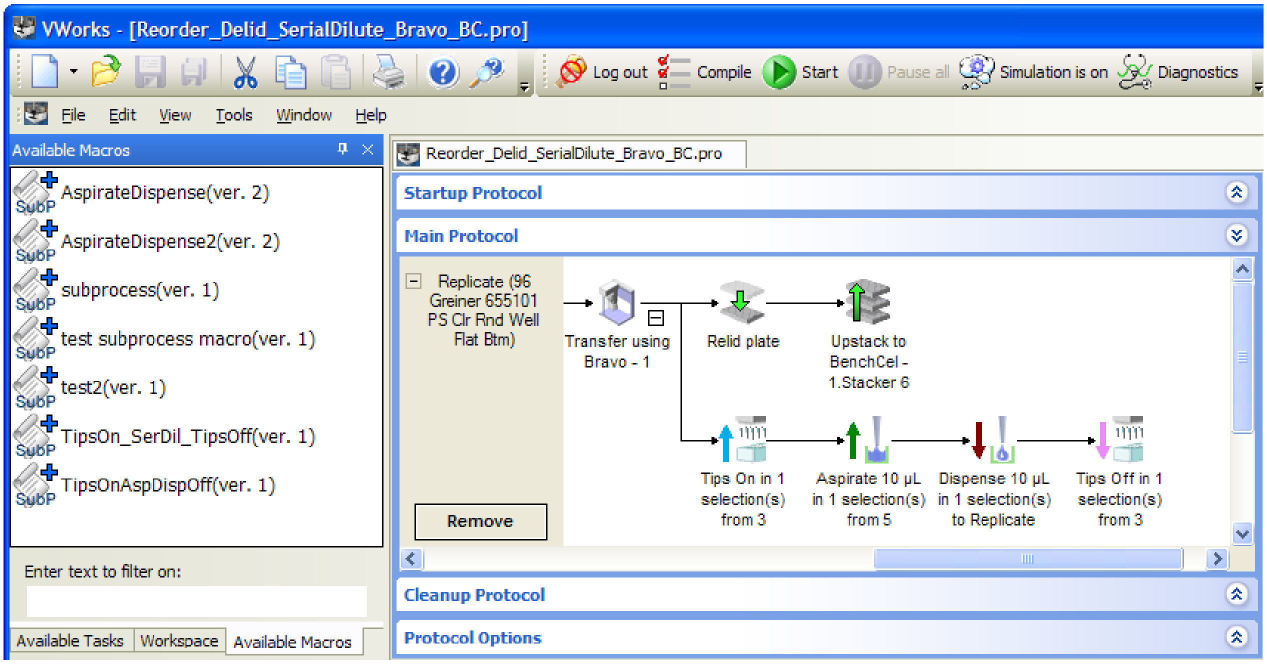
The Available Macros area in the VWorks window displays the contents of the macro library.

Figure Available Macros area displaying the contents of the macro library



The Available Macros area lists the macros alphabetically. If no protocol is open, the macro list includes all types of macros. If a protocol is open, a subset of the macro library appears in the list depending on which area of the protocol tab is active and which devices are used in the protocol. For example, in the following figure only the subprocess macros for the Bravo Platform appear.

Figure Available Macros area displaying only the subprocess macros



Related information

For information about...	See...
Creating macros	"Adding macros to the macro library" on page 137
Editing macros and macro version control	"Editing a macro" on page 142
Using a macro in a protocol	"Inserting macros in protocols" on page 140
Exporting and importing the macro library	"Copying macros to a different computer" on page 151
Exporting and importing protocols	"Exporting and importing protocols and associated components" on page 626
Creating a basic protocol	"Creating a protocol: basic procedure" on page 13
Working with JavaScript	"Creating a protocol: advanced topics" on page 73

Adding macros to the macro library

About this topic

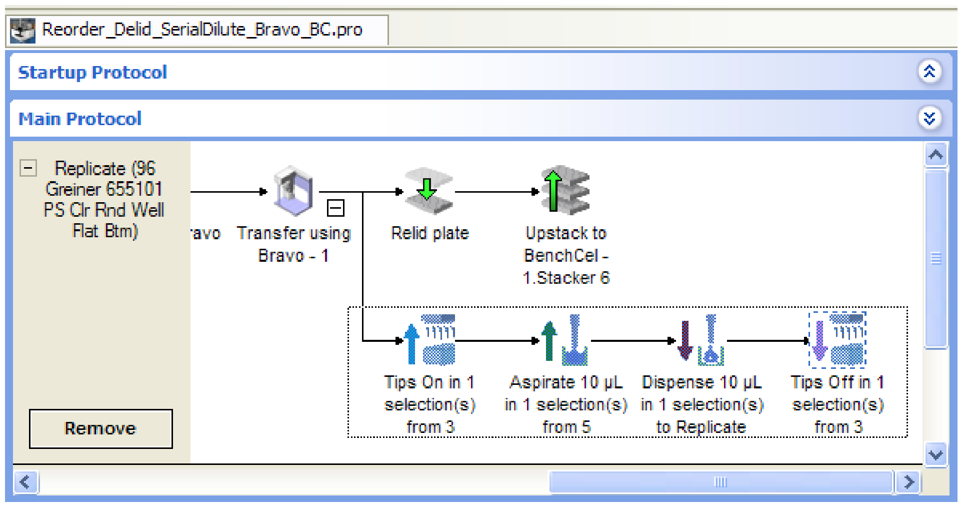
This topic describes how to create macros and add them to the macro library and how to delete macros.

The procedures in this topic assume that a protocol task sequence that you want to replicate as a macro is already created. For details on how to set up tasks and create a protocol, see "Workflow for creating a basic protocol" on page 18.

Creating a macro and adding it to the macro library

To create a macro and add it to the macro library:

1 In the protocol area, drag the pointer around the task or series of tasks that you want to include in the macro. Make sure that only the desired tasks are highlighted.



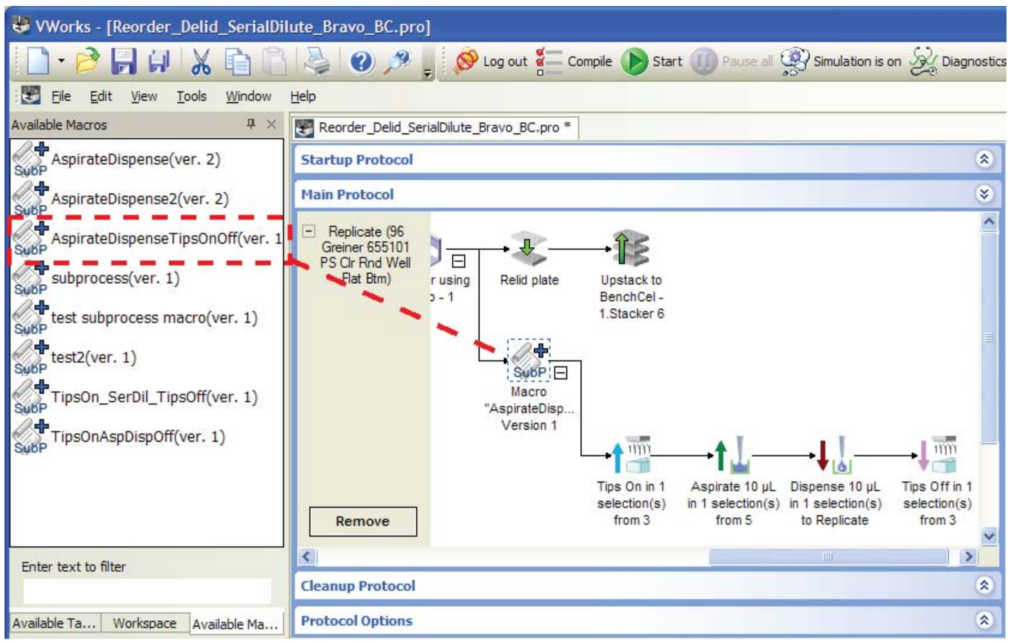
2 Right-click the selection, and then choose **Add to macro library** from the shortcut menu. The Add to Macro Library dialog box opens.

3 Type a name for the macro, and click **OK**. A descriptive name can be more helpful for other macro users than a generic name. A version number is appended to the macro name, starting with version 1.

The corresponding macro icon appears in the Available Macros area and in the protocol. Any future edits of the macro will increment the version number by 1.

IMPORTANT To ensure version control, you should not change the macro name after the macro has been used in a protocol. Although you can rename a macro in the macro library, doing so removes any association to instances of the macro with the former name that are already in use in a protocol.

In the protocol area, expand the macro icon to view and to verify the macro contents.



Note: To associate a custom icon or custom parameters with the macro, see "Editing a macro" on page 142.

Adding a copy of a macro or saving the macro by a different name

You can create a copy of a macro to add to the macro library. For example, you might want to copy a macro and use the copy as the starting point for creating a similar but slightly different macro.

To create a copy of a macro from the Available Macros tab:

1 In the **Available Macros** tab, right-click the macro icon, and then choose **Make a copy** from the shortcut menu.

2 In the **Create Macro Copy** dialog box, type a new name for the copy, and click **OK**. The new macro icon appears in the Available Macros area.

To save a copy of a macro from a protocol:

1 In the protocol, right-click the macro icon, and then choose **Save macro as** from the shortcut menu.

2 In the **Add to Macro Library** dialog box, type a new name for the copy, and click **OK**. The new macro icon appears in the Available Macros area.

Deleting a macro from the macro library or a protocol

You can delete a macro from the macro library or a macro instance from a protocol:

- *Deleting a macro from the macro library*. In this case, instances of the deleted macro can still remain in your protocols. However, the instances of the deleted macro no longer have version control or any association with other instances.
- *Deleting an instance of a macro from a protocol*. In this case, only the selected instance is deleted from the protocol. The macro remains in the macro library and in other protocols where it is used.

  Note: In a protocol where you no longer want a macro, yet you still want to retain a macro's contents (task sequence), you can revert the macro. See "Reverting an instance of a macro to an ungrouped task sequence" on page 142.

To delete a macro from the macro library:

In the **Available Macros** area, right-click the macro icon, and then choose **Delete** from the shortcut menu. The macro icon disappears from the Available Macros list.

To delete an instance of a macro from a protocol:

In the protocol, right-click the macro icon, and then choose **Delete** from the shortcut menu. The macro icon and macro task contents disappear from the protocol.

Related information

For information about...	See...
Editing macros and understanding macro version control	"Editing a macro" on page 142
Adding a custom icon to the macro	"Associating a custom icon with the macro" on page 145
How to expose task parameter variables at the macro level	"Adding parameter variables and scripting to a macro" on page 148
Using a macro in a protocol	"Inserting macros in protocols" on page 140
Creating a basic protocol	"Creating a protocol: basic procedure" on page 13

Inserting macros in protocols

About this topic

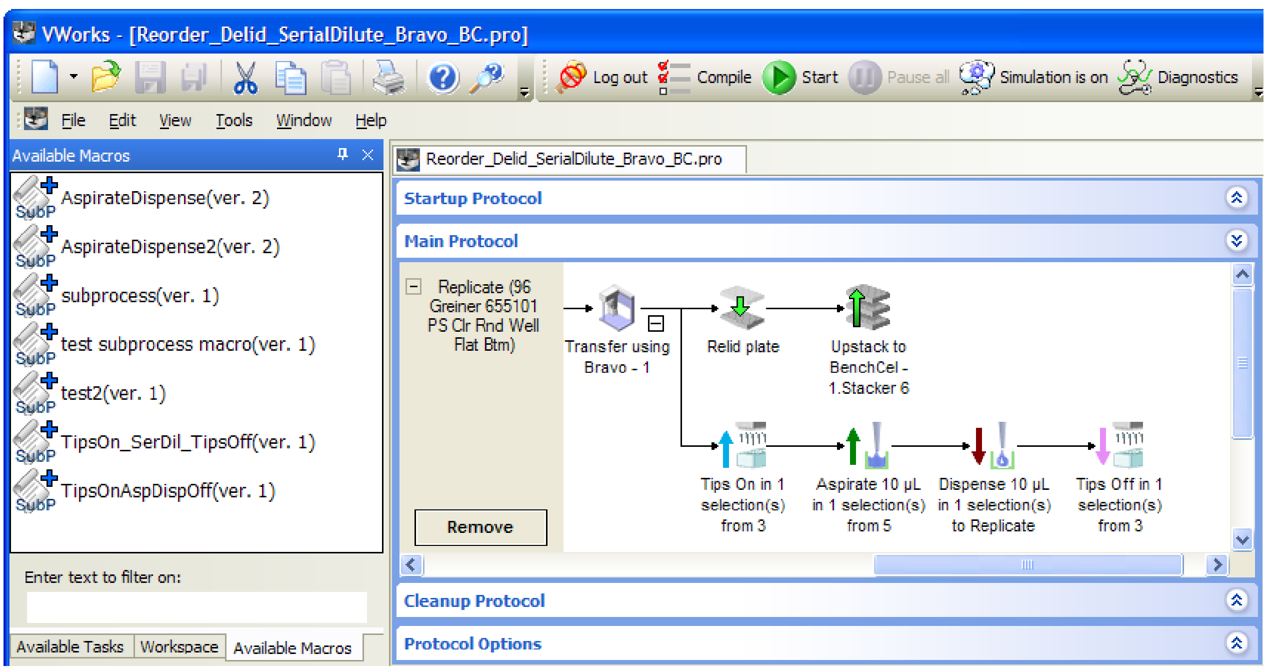
This topic describes how to insert macros into the protocols that you are creating. You must have VWorks Technician- or Administrator-level access to create protocols and work with macros.

Viewing and filtering the list of macros

In the VWorks window, the Available Macros area lists only the macros from the macro library that are relevant for the protocol type (startup, cleanup, main process, or subprocess) and associated devices selected in the protocol tab.

In the following figure, a Bravo subprocess appears in the Main Protocol area and the Available Macros area lists only subprocess macros for the Bravo Platform.

Figure Available Macros area displaying subprocess macros for the Bravo Platform



You can filter the Available Macros list even further.

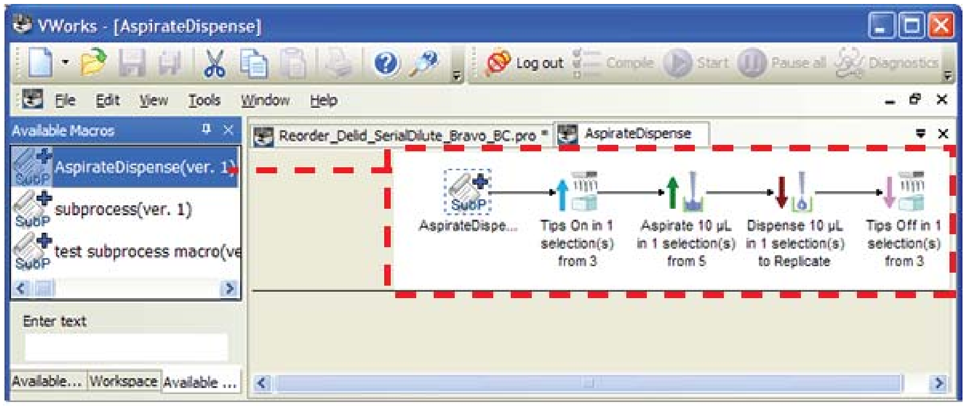
To view and filter the list of macros in the macro library:

1 In the **VWorks** window, choose **View > Available Macros** to display the Available Macros area if not already displayed.

2 In the **Available Macros** area, type the text in the **Enter text to filter on** box that you want to use to filter the list of macro names. As you type each character, the Available Macros list changes to meet the filter requirement.

For example, if you type *wash*, only the macros that include the name *wash* remain in the Available Macros list.

Note: To preview a macro's contents, right-click the macro icon in the **Available Macros** tab, and then choose **Edit** from the shortcut menu. A macro editor tab appears in protocol area, as the following figure shows. To close the macro editor tab, click the corresponding **x** button.



Inserting a macro in a protocol

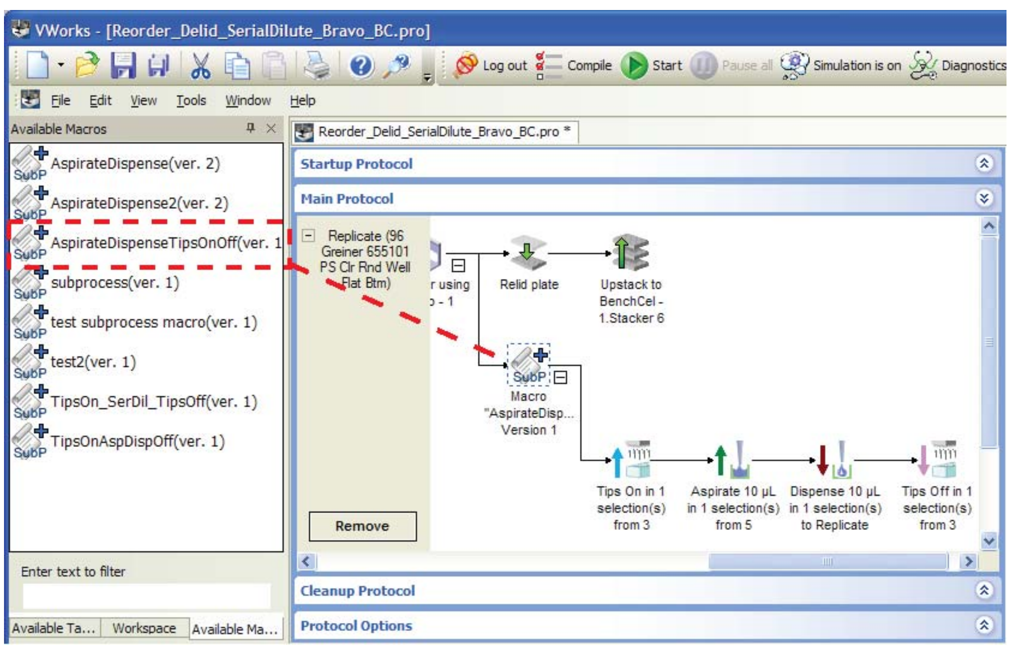
To insert a macro in a protocol:

1 Click the position in the protocol where you want to insert the macro.

2 In the **Available Macros** area, double-click the macro icon. The selected icon appears in the protocol at the location you selected.

Alternatively, you can drag the macro from the **Available Macros** area into position in the protocol.

3 To view the contents of the macro within the protocol, expand the macro icon.



4 Verify that the task parameter settings are correct for the new instance of the macro, including Location, plate and Location, location.

Reverting an instance of a macro to an ungrouped task sequence

If you have an instance in a protocol where you no longer want a macro, yet you still want to retain a macro's contents (task sequence), you can revert the macro.

To revert an instance of a macro:

In the protocol area, right-click the macro icon, and then choose **Ungroup macro**.

The macro icon disappears from the protocol, and the tasks from the macro remain inserted in the protocol position where the macro formerly resided.

Related information

For information about...	See...
Adding macros and removing macros from the macro library	"Adding macros to the macro library" on page 137
How to edit macros and macro version control	"Editing a macro" on page 142
Exporting the macro library	"Copying macros to a different computer" on page 151
Creating a basic protocol	"Creating a protocol: basic procedure" on page 13
Using JavaScript	"Creating a protocol: advanced topics" on page 73

Editing a macro

About this topic

To edit a macro, you must have VWorks Technician- or Administrator-level access.

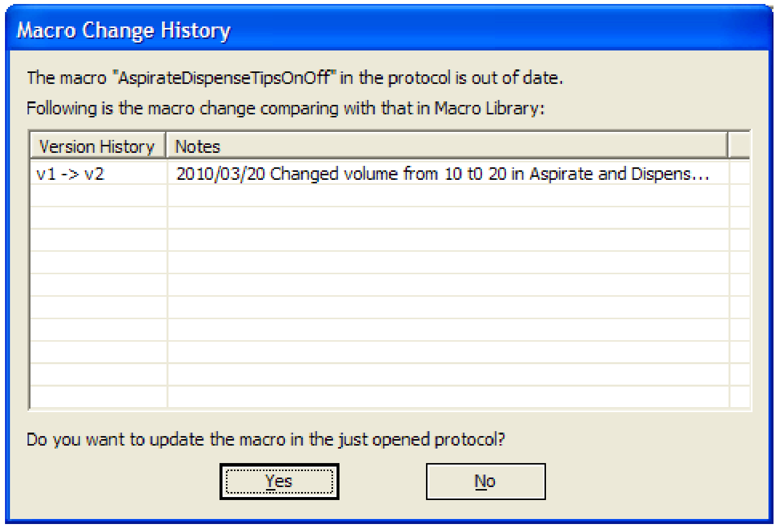
Editing options and macro version control

The macro library stores the latest version of a macro that has been edited in the macro editor tab. You can use different versions of a given macro in your protocols, but the different versions reside only in the protocols.

You can edit the VWorks macros in the following ways:

- *Editing a macro from the macro library in the macro editor tab.* Replaces the version of the same name in the macro library, incrementing the version number by 1.

Later, if you open a protocol that has an earlier macro version than the macro in the macro library, the following Macro Change History dialog box opens. You can choose whether to update the macros in the protocol to match the definition in the macro library.



- *Editing a macro directly in the protocol tab*. Updates the macro in the protocol only. Macros in the macro library remain unchanged. Instead, you have the following options:
  - Keep a derivative of the definition from the macro library in the selected protocol.
  - Replace the modified instances in the protocol with the definition from the macro library.
  - Add the modified definition under a new name in the macro library.

  The software does not maintain a change history of the macros that are edited in the protocol tab. However, you can easily update all instances of a macro in a protocol to match the version of the same name in the macro library. See "Updating instances of a macro to match the version in the macro library" on page 147.

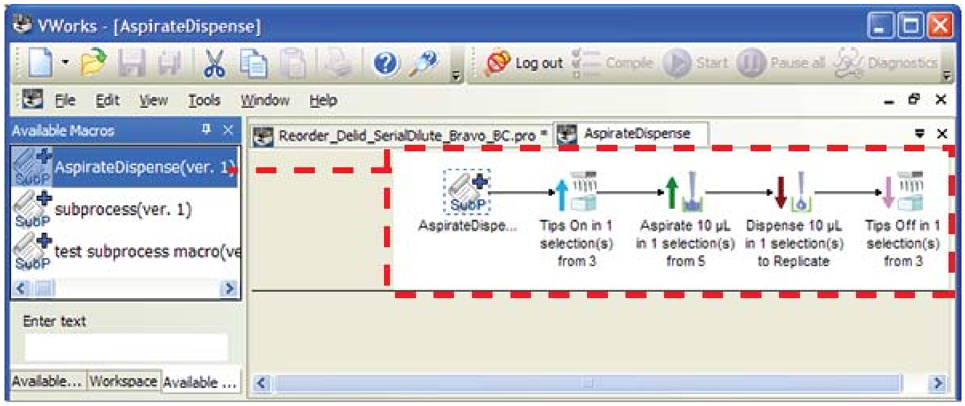
Editing a macro from the library in the macro editor tab

To edit a macro from the library in the macro editor tab:

1. In the **Available Macros** tab, right-click the macro icon, and then choose **Edit** from the shortcut menu.

   A macro editor tab opens in protocol area and displays the contents of the macro as the following figure shows.

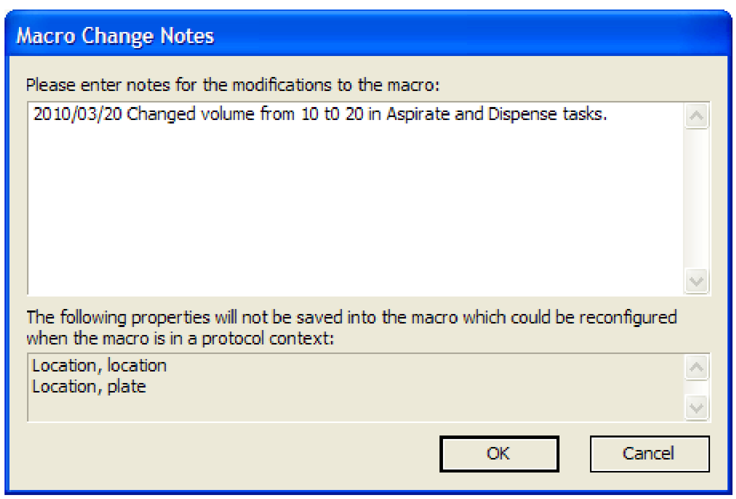
Figure Macro editor tab



2 To add or delete a task:

- *Add a task*. Either drag the task icon from the **Available Tasks** area to the macro editor tab, or double-click the task icon in the **Available Tasks** area to add the task to the macro editor tab.
- *Delete a task*. In the macro editor tab, right-click the task icon, and then choose **Delete** from the shortcut menu.
- *Change a task's parameter values*. Click the task icon in the macro editor tab. In the **Task Parameters** area, edit the task parameters as you typically would when creating a protocol.
- Associate a custom icon with the macro. See "Associating a custom icon with the macro" on page 145.
- Associate custom parameters with the macro. See "Adding parameter variables and scripting to a macro" on page 148.

3 Choose **File > Save**. The Macro Change Notes dialog box opens:



4 Type a description of your edits, and then click **OK**.

Later, if you open a protocol containing a different version of this macro, the software displays this change history for reference.

5 When the message appears, asking if you want to update the macro in the protocol:

- Click **Yes** to update all instances of the macro with the same name in the specified protocol, regardless of version number. The software increments the macro version number by 1 in the macro library.
- Click **No** to increment the macro version number by 1 in the macro library only. Any previous versions of the macro in the protocol with the same macro name remain in the protocol intact.

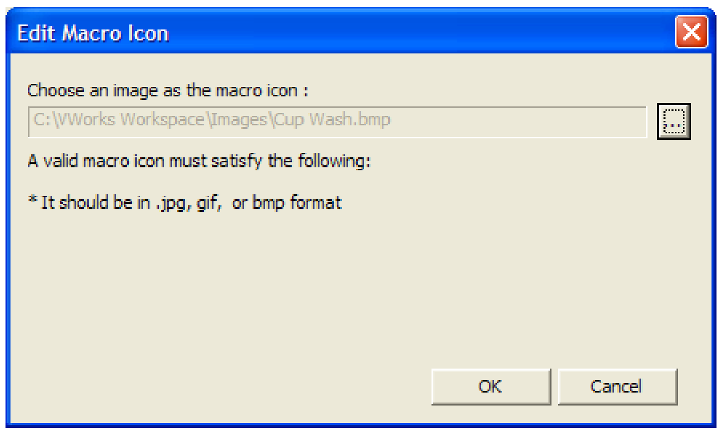
If any unopened protocols also contain the previous version of the macro, you have the option to update the macros when you open each protocol.

Associating a custom icon with the macro

To associate a custom icon with the macro:

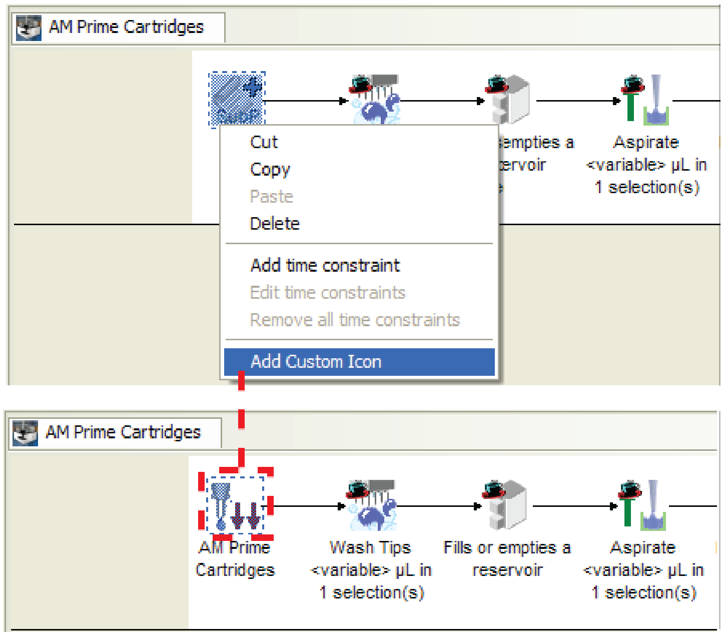
1 In the macro editor tab, right-click the macro icon, and then choose **Add Custom Icon** from the shortcut menu. The Edit Macro Icon dialog box opens.

2 Click the ... button, select the image file (.jpg, .bmp, or .gif), and then click **Open**.



3 Click **OK**. The selected image replaces the macro icon in the macro editor tab.

Figure Macro before (top) and after (bottom) adding custom icon in macro editor tab



Editing a macro directly in the protocol tab

Macros in the macro library remain unchanged when you edit an instance of a macro in the protocol.

To edit a macro in the protocol tab:

1 In the protocol, expand the macro icon so that you can view the macro task contents.

2 Edit the tasks as desired:

- *Add a task*. Drag the task icon from the **Available Tasks** area to the macro in the protocol, or double-click the task icon in the **Available Tasks** area.

  Alternatively, you can copy a task icon from elsewhere in the protocol. To do this, right-click the task icon, and then choose **Copy**. Click the target location in the macro, right-click, and then choose **Paste**.

- *Change task parameter values*. Click the task icon. Edit the values in the **Task Parameters** area as you typically would when creating a protocol.

- *Delete a task*. Right-click the task icon, and then choose **Delete** from the shortcut menu.

- Associate a custom icon with the macro. See "Associating a custom icon with the macro" on page 145.

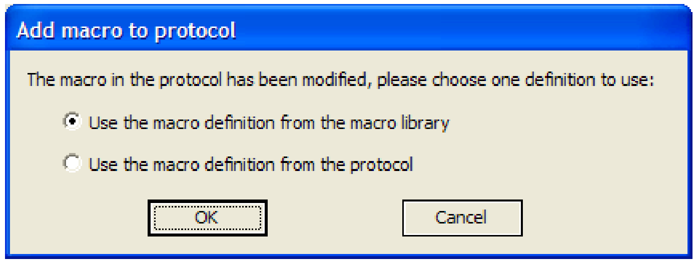
- Add custom parameters and scripting to the macro. See "Adding parameter variables and scripting to a macro" on page 148.

3 Do one of the following:

- *To rename the edited macro and add it to the macro library*. In the protocol, right-click the icon of the edited macro, and then choose **Save macro as**. The Add to Macro Library dialog box opens.

- *To keep the edited macro definition in the protocol only.* Choose **File > Save**. The changes are applied to all instances in the protocol that have the same name and version number. The name of the edited instance changes to "*macro_name* Derived from Version n".

  Note: If you subsequently add an additional macro of the same name from the macro library to this protocol, the Add macro to protocol dialog box opens. You can choose which definition to use for all the macros in the protocol that have the same name.



Updating instances of a macro to match the version in the macro library

If a protocol contains a derivative version of a macro or a different version than one of the same name in the macro library, you can choose to update all instances in the open protocol to the definition in the macro library or to keep the definition from the protocol.

To update a macro in the protocol tab to the current version in the macro library:

In the protocol tab, right-click the macro icon, and then choose one of the following:

- **Update this macro.** Replaces the selected macro instance in the open protocol with the definition from the macro library.
- **Update all revisions of this macro**. Replaces all macro instances of the same name with the definition from the macro library, regardless of the version number of the other macro instances in the protocol.

Renaming a macro in the macro library

CAUTION Renaming a macro in the macro library removes any association to the macro instances of the former name that are in the protocols.

To rename a macro in the macro library:

1 In the **Available Macros** tab, right-click the macro icon, and then choose **Rename** from the shortcut menu.
2 In the **Rename Macro** dialog box, type the new name and click **OK**.

Related information

For information about...	See...
Adding macros and removing macros from the macro library	"Adding macros to the macro library" on page 137

For information about...	See...
Using macros in protocols	“Inserting macros in protocols” on page 140
Exporting the macro library	“Copying macros to a different computer” on page 151
Creating a basic protocol	“Creating a protocol: basic procedure” on page 13
Using JavaScript	“Creating a protocol: advanced topics” on page 73

Adding parameter variables and scripting to a macro

About this topic

This topic describes the following:

- “Adding macro parameter variables” on page 148
- “Filtering the macro parameters” on page 149
- “Adding JavaScript to a macro” on page 150

This topic assumes that you have an understanding of how to create VWorks protocols, including how to do the following:

JavaScript tasks	For details, see...
Define JavaScript variables in the VWorks software	“Using simple variables” on page 77
Write programs in JavaScript in the VWorks software	“Using JavaScript” on page 84

Adding macro parameter variables

You can expose parameters for individual tasks at the task macro level. For example, you might want to edit the parameter values of tasks contained in the macro without having to open the macro and then open each task to access the task parameters.

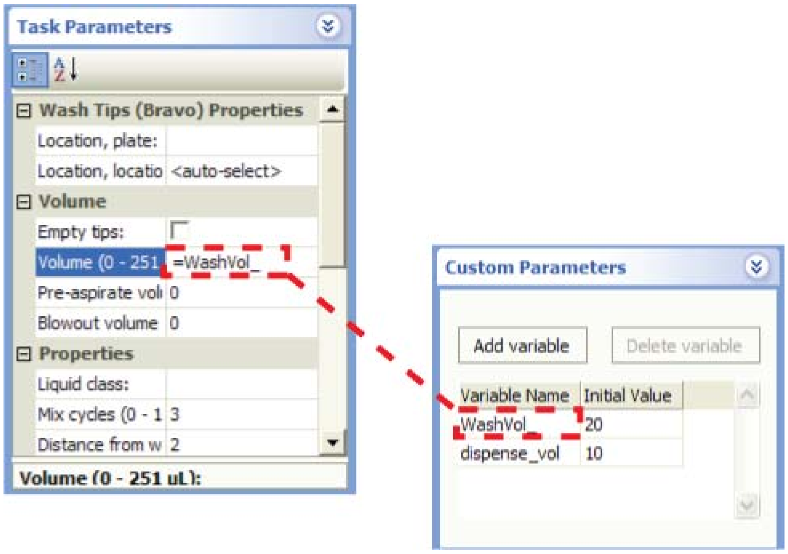
To add task parameter variables to the macro:

1. In the **Available Macros** tab, right-click the macro icon, and then choose **Edit** from the shortcut menu.

   A macro editor tab opens in protocol area and displays the contents of the macro.
2. In the macro editor tab, click the macro icon. The Custom Parameters area replaces the Task Parameters area next to the protocol area.
3. Click **Add Variable**. A row appears in the Variable Name table.

4 In the new row, click the **Variable Name** field, and then type the exact same JavaScript variable that appears in the Task Parameters area for the given task parameter.

For example, a macro contains multiple Wash Tips tasks for which you are using the same volume. In the corresponding Task Parameters for each task, you define the Volume parameter as a variable, such as `=WashVol_`. You use the same variable, `WashVol_` as the variable name in the macro's custom parameters. You can change the volume specified for both tasks by editing the variable value in the macro's Custom Parameters area.



5 In the **Initial Value** field of the **Custom Parameters** area, type the value for variable.

For more details on using JavaScript variables for task parameters, see "Using simple variables" on page 77.

Filtering the macro parameters

To show all macro parameters:

In the **Custom Parameters** area, right-click the variables table, and select **Show all**.

To filter the macro parameter list:

1 In the **Custom Parameters** area, right-click the variables table, and select **Show all**.

2 Right-click a cell in the table, and select from the available filtering options:

View	Description
Last filter	Returns the display to when the last filter was applied.
Filter by row	Displays the parameters that have the same value as the parameter in the selected row. For example, if you choose to filter the list based on the initial value of a given parameter. Only the parameters that have the same value as the parameter in the selected row are displayed.

View	Description
Filter by column	Displays the parameter names that are an exact match of the selected column.

Adding JavaScript to a macro

You can write JavaScript for a macro in a similar way that you can for a task. For example, you could enable a macro for one set of conditions and disable the macro for all other conditions using the `task.skip()` method.

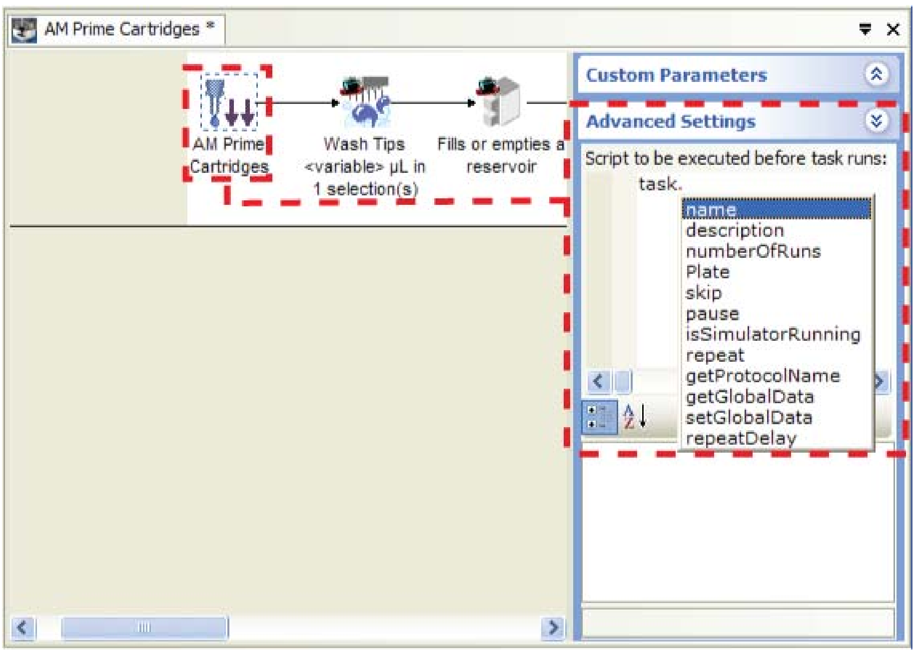
At the macro level, you can use only the shared methods of the task object. For a description of the task object's shared methods, see "task object" on page 98.

To add JavaScript to a macro:

1 In the **Available Macros** tab, right-click the macro icon, and then choose **Edit** from the shortcut menu.

A macro editor tab opens in protocol area and displays the contents of the macro.

Figure Macro editor tab and the Advanced Settings area



2 In the macro editor tab, click the macro icon. The Custom Parameters area replaces the Task Parameters area next to the protocol area.

3 Click **Advanced Settings**.

4 In the **Script to be executed before task runs** box, type your script.

Note: To view the list of available properties for the task object, type `task.` You can select a property from the list that appears.

Related information

For information about...	See...
Adding macros to the macro library	“Adding macros to the macro library” on page 137
Macro version control	“Editing a macro” on page 142
VWorks JavaScript variables	“Using simple variables” on page 77
Advanced Parameters area	“task object” on page 98

Copying macros to a different computer

About this topic

To work with macros, you must have VWorks Technician- or Administrator-level access. You can copy macros to a different computer in the following ways:

- *Copy the entire macro library*. This topic describes how to copy the entire macro library.
- *Export and import the macros that are part of a protocol*. If you export a protocol (.vzp file), the macros in the protocol are included in the exported .vzp file. You also have the option of exporting the protocol macros of the same name from the macro library as well as the protocol. In this case, the macros appear in the Available Macros tab as well as in the protocol after you import the protocol. For details, see “Exporting and importing protocols and associated components” on page 626.

Copying the macro library and verifying the file path

To copy the macro library to a different computer:

1 Copy and paste the library file (.mlb) from one computer to another.

  By default, the file path is as follows:

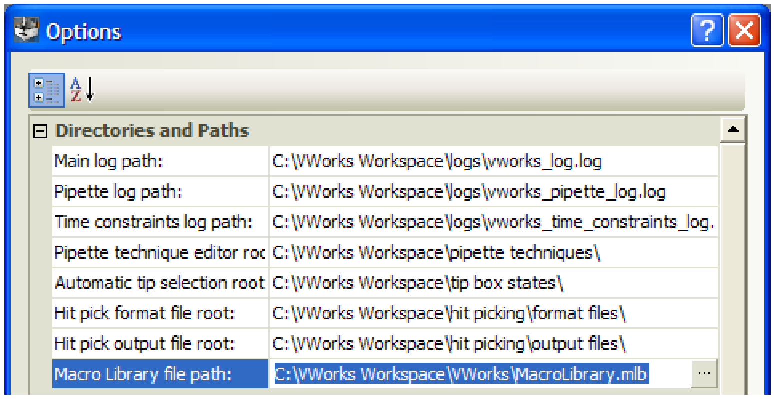
  ...\VWorks Workspace\VWorks\MacroLibrary.mlb

2 On each computer, verify that the file path for the macro library is set in the VWorks software.

To verify or change the file path for the macro library:

1 In the **VWorks** window, choose **Tools > Options**. The Options dialog box opens.

2 Under **Directories and Paths**, verify that the file path in the **Macro Library file path** field is correct.



3 If the file path is not displayed or is not correct:

a Click the **Macro Library file path** field, and then click the ... button that appears.

b In the **Open** dialog box, locate and select the .mlb file (for example, MacroLibrary.mlb), and then click **Open**. The new file path appears in the **Macro Library file path** field.

Related information

For information about...	See...
Exporting and importing macros	"Exporting and importing protocols and associated components" on page 626
Macros and the macro library	"About protocol macros and the macro library" on page 134
Creating macros	"Adding macros to the macro library" on page 137
Using macros to help create protocols	"Inserting macros in protocols" on page 140
Creating a basic protocol	"Creating a protocol: basic procedure" on page 13
Working with JavaScript	"Creating a protocol: advanced topics" on page 73

5
Creating protocol forms for operators

You can create forms with custom user interfaces for the operators who run your protocols. A form provides a simplified interface, which can be helpful for users who have limited or infrequent experience with the VWorks software.

You must have VWorks Administrator or Technician access to create protocol forms, and you should also have an understanding of how to create VWorks protocols.

This chapter contains the following topics:

About forms for running protocols

About this topic

To create or edit forms, you must have VWorks Technician- or Administrator-level access. You should also have an understanding of how to create protocols in the VWorks software.

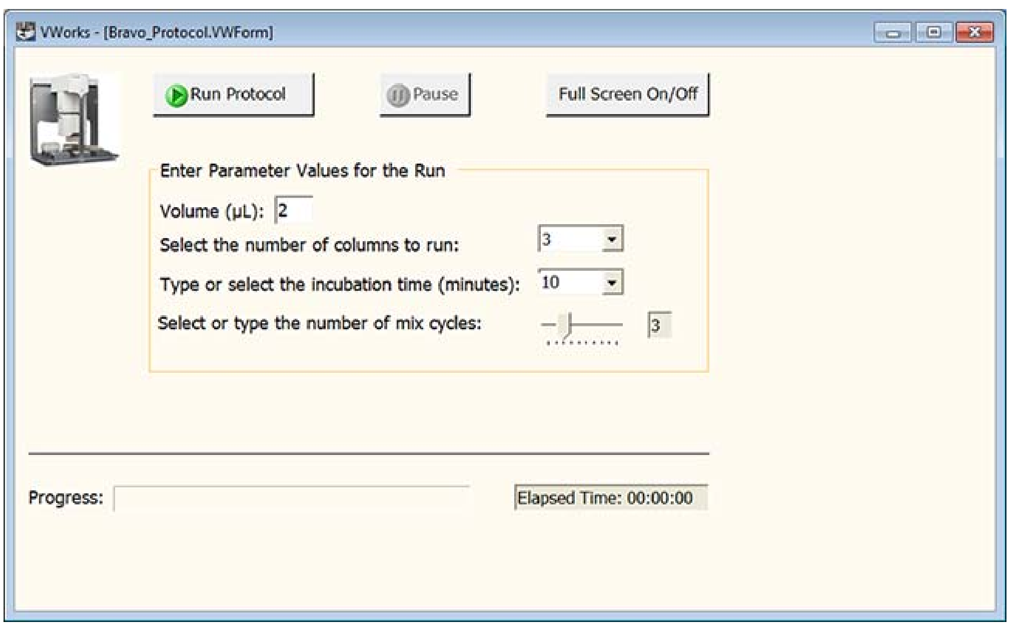
This topic describes the VWorks forms and the Form Designer window.

Forms defined

A form is a customized user interface that enables operators to run an associated VWorks protocol. A VWorks user who creates protocols (Technician or Administrator access) can design forms to simplify the operator duties when it comes time to run the associated protocols.

The form can be simple or relatively complex, depending on the requirements of a target group of operators. At a minimum, the form should have buttons for starting and pausing the associated protocol. You may also want to include the Full Screen On/Off button (default) to enable the form users to toggle the VWorks window between a full screen form view and the normal tabbed view. The form design can allow the operator to enter task parameter values for the associated protocol or limit the operator to a read-only display of task parameter values, progress, and other protocol data that the form's author wants to display.

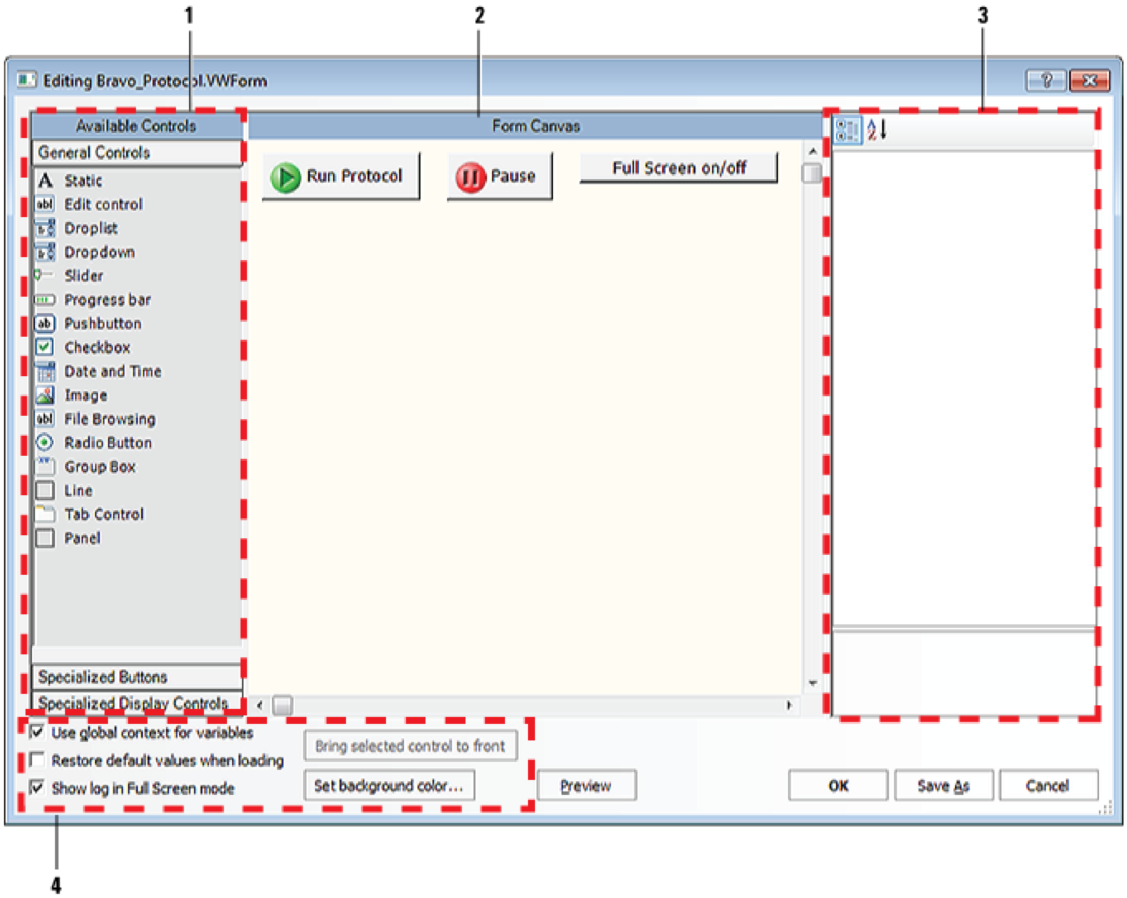
Figure Example of VWorks window displaying a form in full screen view without the logs



Form Designer overview

The following figure shows four primary areas in the Form Designer window, and the table describes these four areas.

Figure Form Designer window



Item	Area	Description
1	Available Controls	The control palette, which includes three categories from which to select controls and indicators.
2	Form Canvas	The work area where the form's author arranges the form controls while viewing a representation of the form. Three buttons are included in this area by default: Run Protocol, Pause, and Full Screen on/off. As you drag controls into position in the Form Canvas, invisible grid lines assist with the alignment based on the controls that are already in the form. *Note:* To disable the autogrid, press ALT while dragging the controls into place.
3	Control Properties	The properties associated with a control that is currently selected in the Form Canvas. The figure shows the Run Protocol Properties because the Run Protocol button is selected.
4	Form properties	The properties that apply to the entire form, such as the background color and the context for JavaScript variables that are used in the form.

Available controls overview

The Available Controls area of the Form Designer window contains the following areas, each of which provides different types of controls that you can use in your form design:

- "General Controls area" on page 156
- "Specialized Buttons area" on page 157
- "Specialized Display Controls area" on page 158

General Controls area

Most of the controls in the General Controls area can be assigned a JavaScript variable. By assigning a JavaScript variable, you can allow form users to edit task parameter values or enable the display of runtime data. The JavaScript variable must be assigned in both the control properties and in the corresponding task parameter in the protocol.

The following table provides an overview of the general controls. For more details about the general controls and how to configure them, see "Adding form controls that allow editing or runtime data display" on page 176.

Control	Description
Static (text)	Displays read-only text in the form. The Static control can be used to: • Provide a caption for another control. • Display runtime data if a JavaScript variable is assigned in the control properties and in the corresponding task parameter in the protocol.
Edit control	Allows the form users to enter or edit data for a task parameter.
Droplist	Presents a drop-down list of choices for a task parameter.
Dropdown	Allows the form users to type arbitrary text into the edit field or choose from a drop-down list of choices for a task parameter.
Slider	Lets users set a value on a continuous range of possible values for a task parameter.
Progress bar	Provides a visual indication of progress for a task.
Pushbutton	Allows you to write your own script for the action the button will perform for the form users.
Checkbox	Allows users to enable or disable the corresponding actions for a task parameter.
Date and Time	Displays the preset date and time or a date that the user can set for a task.
Image	Displays a static image.
File Browsing	Enables form users to select a file.

Control	Description
Radio Button	Allows you to present options for a task parameter.
Group Box	Displays a rectangle that can be resized and labeled to visually group other controls together.
Line	Provides a vertical or horizontal line that you can use to visually separate items on the form.

Specialized Buttons area

The Specialized Buttons area contains button controls that are already programmed to perform specific commands, which are currently available elsewhere in the VWorks window.

The following table provides an overview of the specialized buttons. For more details about the buttons and how to configure them, see "Configuring a run button and other specialized buttons in a form" on page 164.

Button	Description
Toggle Full Screen	Changes the display of the VWorks window from a normal view to a full-screen view and back again. In the normal view, the form is a tab within the protocol area of the VWorks window. In the full-screen view, the VWorks window displays only the form, and optionally the logs.
Save Data Entry values	Saves all data entry values associated with the form's controls in the form file (.VWForm).
Reset Data Entry values	Resets all data entry values, which are associated with the controls in the form, to the default values.
Print Data Entry values	Prints all data entry values associated with the controls in the form.
Initialize All Devices	Opens the device file (.dev) specified by the button, and establishes communication with the corresponding devices. The following run buttons also initialize the devices for the corresponding protocol if simulation mode is off.
Run Protocol Run Specified Protocol Run Runset	Provide a choice of preprogrammed buttons for starting the protocol or runset.
Pause Run	Pauses the run and opens the Scheduler Paused dialog box, which is the same function as the Pause all button on the toolbar in the VWorks window.

Button	Description
Menu Action	Performs a command that the form designer selects from a list of commands, which are also available in the Tools menu of the VWorks window. This button control can enable a user with VWorks Operator-level access the ability to perform a command that normally requires VWorks Technician-level access.

Specialized Display Controls area

The controls in the Specialized Display Controls area display data that is not drawn from a JavaScript variable.

The following table provides an overview of the specialized display controls. For more details about the specialized display controls and how to configure them, see "Adding indicators for elapsed time and progress to a form" on page 173.

Control	Description
Elapsed Time	Displays the duration (hours:minutes:seconds) of the corresponding protocol or runset that the form is running.
Overall Progress	Displays a visual progress indicator for the corresponding protocol or runset that the form is running.

Related information

For information about...	See...
Creating or editing a form	"Workflow for creating or editing a form" on page 160
Controls under General Controls in the Form Designer window	"Adding form controls that allow editing or runtime data display" on page 176
Controls under Specialized Buttons in the Form Designer window	"Configuring a run button and other specialized buttons in a form" on page 164
Controls under Specialized Display Controls in the Form Designer window	"Adding indicators for elapsed time and progress to a form" on page 173
Form properties in the Form Designer window	"Setting the form properties" on page 201
Creating a basic protocol	"Workflow for creating or editing a form" on page 160

For information about...	See...
Working with JavaScript	"Creating a protocol: advanced topics" on page 73

Workflow for creating or editing a form

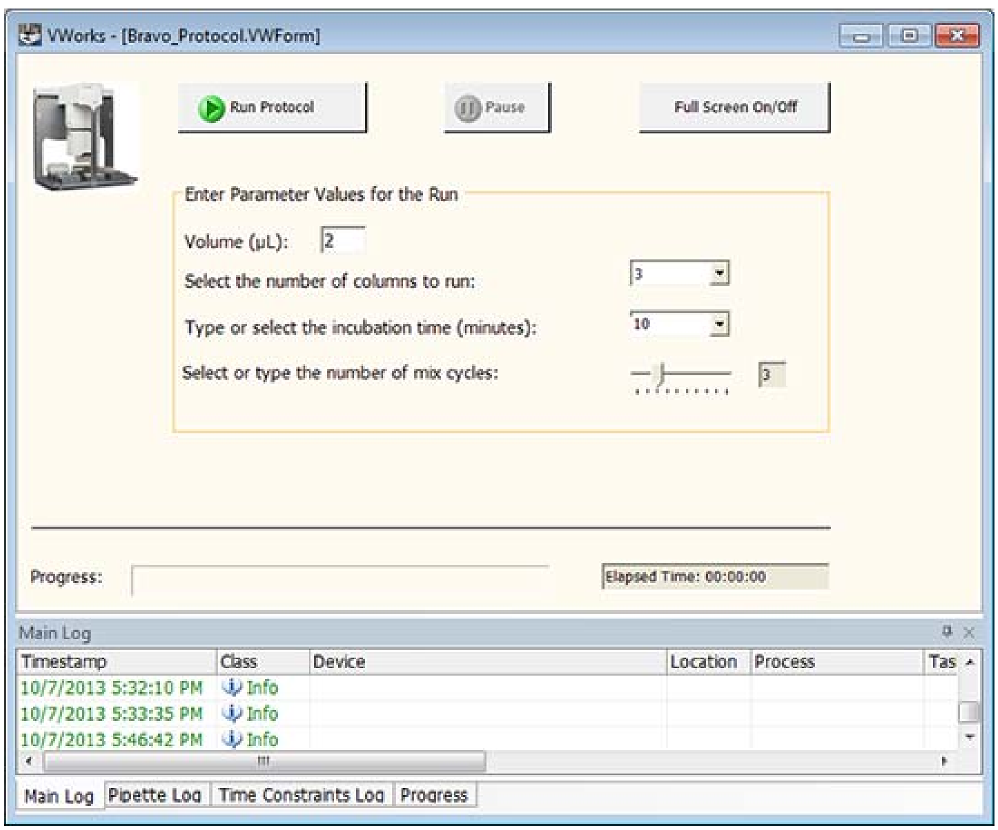
About this topic

To create a form, you must have VWorks Technician- or Administrator-level access and an understanding of how to create protocols in the VWorks software.

Before you start

- Create the protocol (.pro) or runset (.rst) that you want the form to run.
- Determine how you want the form to function. For example, will users simply press a run button to start the associated protocol, or will they be required to enter task parameter values before the run?
- If the form will include controls that display runtime data for a given task or require a user to enter values, you must have a JavaScript variable associated with the control and the corresponding task parameter. For details on JavaScript variables, see "Using simple variables" on page 77.
- Determine whether the form users will require documentation specific to your form.

Figure Example VWorks form in full screen view with logs



Workflow

Step	For this task...	See...
1	Open the Form Designer.	"Opening the Form Designer" on page 162
2	Select an appropriate button to run the protocol or runset.	"Configuring a run button and other specialized buttons in a form" on page 164
3	Determine whether to include the Pause and Full Screen on/off buttons (default), and add any other specialized buttons that you want to use.	"Configuring a run button and other specialized buttons in a form" on page 164
4	Add indicators for the elapsed time and progress of the protocol or runset that the form is running.	"Adding indicators for elapsed time and progress to a form" on page 173
5	Add controls that display or allow editing of the run data.	"Adding form controls that allow editing or runtime data display" on page 176
6	Set the form's parameters and save the form.	"Setting the form properties" on page 201

Related information

For information about...	See...
Creating or editing a form	"Workflow for creating or editing a form" on page 160
Creating a basic protocol	"Creating a protocol: basic procedure" on page 13
Working with JavaScript	"Creating a protocol: advanced topics" on page 73
JavaScript context and form design	"Understanding JavaScript context in form design" on page 204

Opening the Form Designer

About this topic

To open the Form Designer, you must have VWorks Technician- or Administrator-level access.

This topic describes how to open the Form Designer and how to move and resize the window.

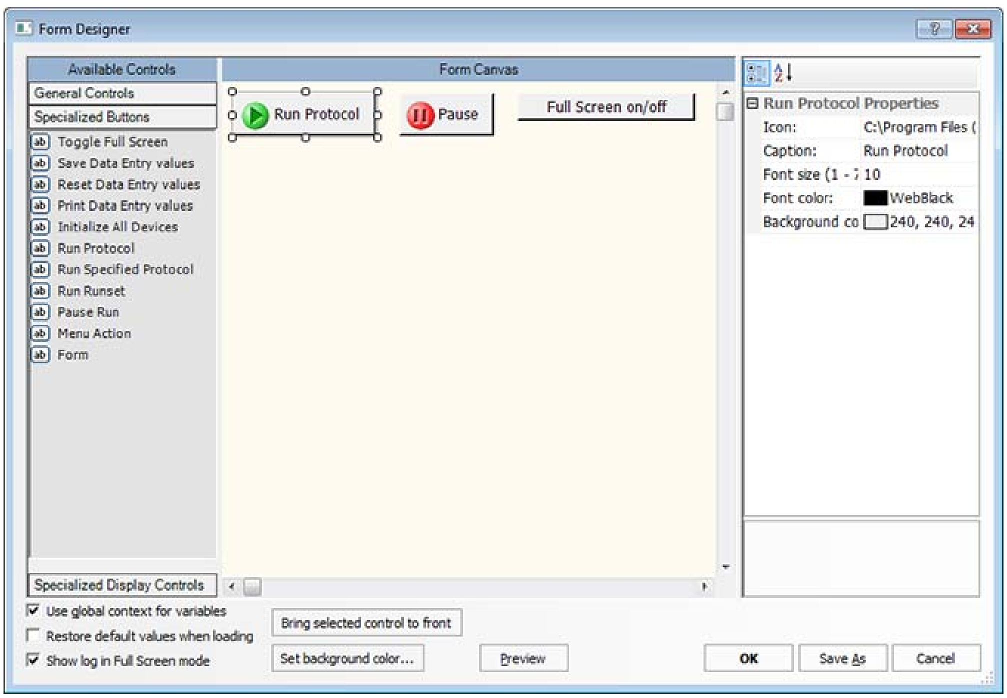
Procedures

To open the Form Designer and create a new form:

1 In the **VWorks** window, choose **File > New > Form**. The Form Designer window opens, and three buttons appear by default in the Form Canvas area.

2 In the **Form Designer** window, click **Save As**. In the **Save As** dialog box, specify the file name and storage location. The file is saved with the .VWForm file extension.

Figure Form Designer window with the three default buttons



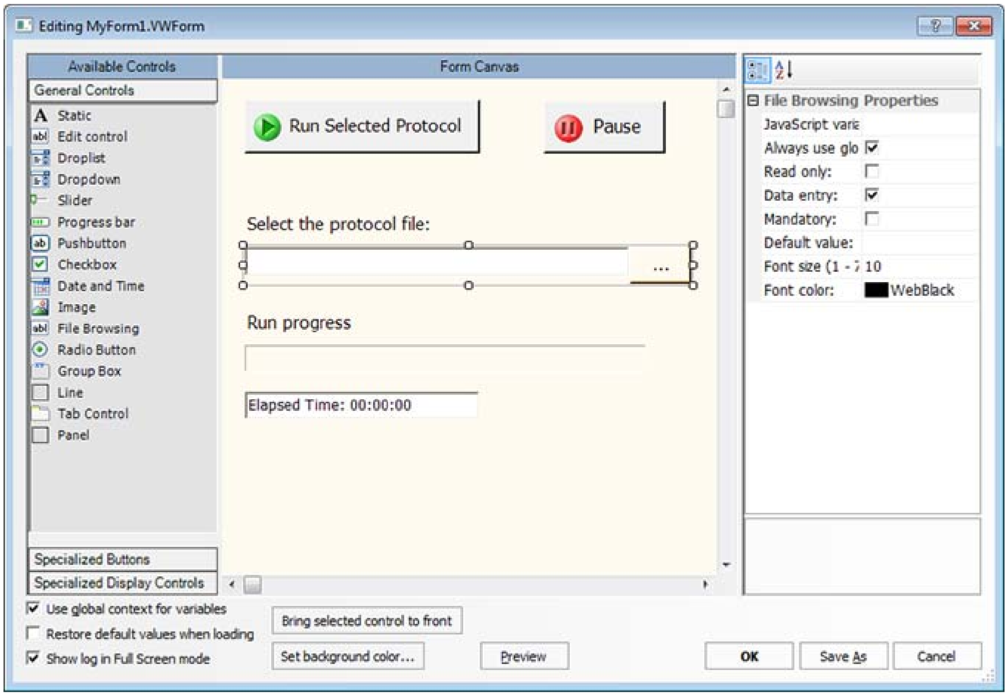
To open the Form Designer and edit a form:

1 In the **VWorks** window, choose **File > Open**. In the **Open** dialog box, select the .VWForm file that you want to edit, and then click **Open**.

The form tab appears in the protocol area of the VWorks window.

2 Choose **Tools > Edit Form**. The selected form appears in the Editing .VWForm window.

Figure Editing .VWForm window with a form to be edited



Moving and resizing the window

You can move the Form Designer window to the side of the screen so that you can continue to access the protocol task parameters while designing your form.

To move the Form Designer window:

Drag the Form Designer window to the new location.

To resize the Form Designer window:

Move the pointer to an edge of the window. When the pointer turns into a double-sided arrow, drag the edge of the window to the new size.

To resize the areas inside Form Designer window:

Move the pointer over the vertical splitter bar that separates each area. When the pointer turns into a double-sided arrow, drag the splitter to the new position.

Related information

For information about...	See...
Workflow for creating or editing a form	"Workflow for creating or editing a form" on page 160
Forms and Form Designer overview	"About forms for running protocols" on page 154
Creating a basic protocol	"Workflow for creating or editing a form" on page 160

For information about...	See...
Using JavaScript	"Creating a protocol: advanced topics" on page 73

Configuring a run button and other specialized buttons in a form

About this topic

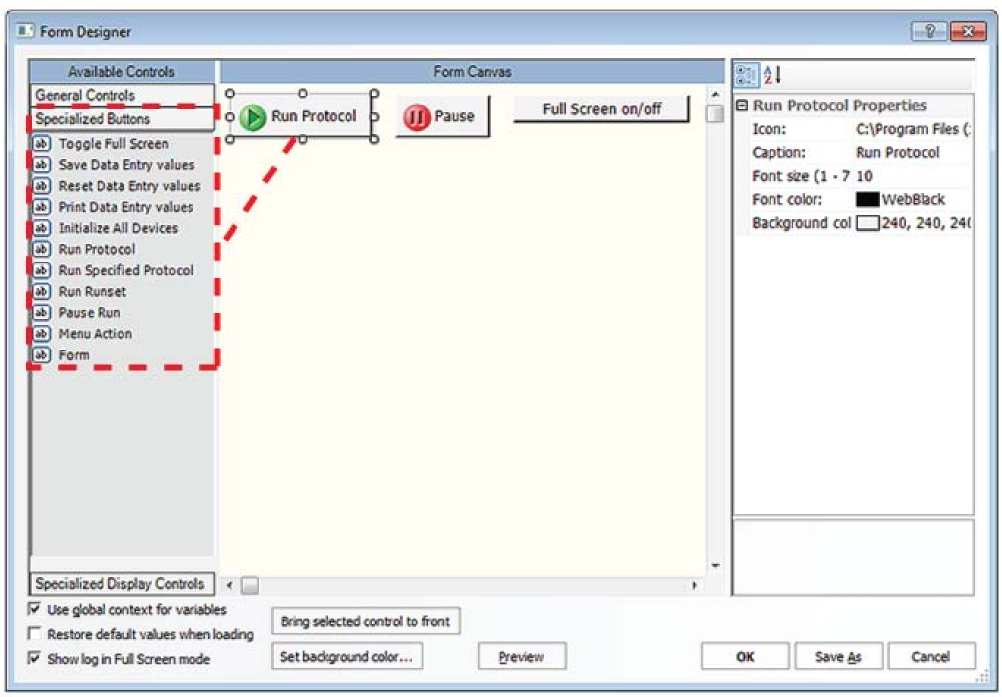
To create or edit forms, you must have VWorks Technician- or Administrator-level access. You should also have an understanding of how to create protocols in the VWorks software.

This topic describes the following:

- "Determining which run button to include" on page 165
- "About including the Pause and Full Screen on/off buttons" on page 165
- "About including other specialized buttons" on page 167
- "Configuring the specialized buttons" on page 169

The Form Designer window contains buttons that perform specific commands, which are available elsewhere in the VWorks window. You can find these buttons under Specialized Buttons in the Form Designer window.

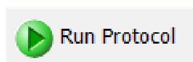
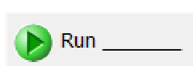
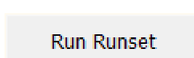
Figure Form Designer window Specialized Buttons area and the default buttons



For more details on creating forms, see "Workflow for creating or editing a form" on page 160.

Determining which run button to include

The following table describes four different options that you can use as the run button on your form. To configure these run buttons, except the Pushbutton control, see "Configuring the specialized buttons" on page 169 in this section.

Run buttons	Description
Pushbutton control	Provides the most flexible way to run a protocol using a form, because you can allow the form users to choose which protocol to run and the number of run times. The Pushbutton control is under the General Controls area. For details on how to configure the Pushbutton control, see "Adding form controls that allow editing or runtime data display" on page 176 and "Example: Creating a scripted Pushbutton control in a form" on page 197.
Run Protocol (default) 	Opens the Run Configuration Wizard for the protocol, so that the user can enter the number of times to run, when to start the run, and the starting barcode. When the user clicks Finish in the wizard, the protocol starts, or is scheduled to start in the future. The Run Protocol button is available to the form user only if the protocol is open and no run is currently in progress. The form can be used to start different protocols, but only one instance of the form can be open and running a given protocol at a time. *Setup requirements*. In the Protocol Options area of the VWorks window, you must specify the form to use (.VWForm). You can also choose to automatically load the form file, so that the form opens automatically when the protocol is opened. If you export a protocol that has a specified form to use, the form file can be included in the exported files.
Run Specified Protocol 	Starts the run for the specified protocol immediately. The Run Configuration Wizard does not appear in this case. *Setup requirements*. The properties for the button must specify the protocol file (.pro) and the number of times to run it.
Run Runset 	Starts the specified runset. The Run Configuration Wizard does not appear in this case. If any runs are scheduled to start as soon as possible, a warning message asks the operator to make sure that the system is ready for the runs to start. *Setup requirements*. The properties for button must specify the runset file (.rst), and you must select the Use global context for variables check box for the form.

About including the Pause and Full Screen on/off buttons

IMPORTANT Be sure to include the Toggle Full Screen button in your form if you want the user to view the form in the full screen view.

When you open the Form Designer to create a form, the Pause and Full Screen on/off buttons are included in the form by default. If you accidentally remove these buttons, you can reselect the Pause Run and Toggle Full Screen controls from the Specialized Buttons area of the Form Designer window.

Buttons	Description
Pause Run (default)  	Provides the same function as the Pause all button on the toolbar: pauses the run and opens the Scheduler Paused dialog box. For details, see “Pausing the run” on page 261. Include the Pause button in your form to give the form users a way to pause, continue, or abort the run when the form is in full-screen view and the toolbar is hidden.
Toggle Full Screen (default)  	Changes the VWorks window between the normal view and full-screen view of the form: • *Full screen on*. Displays only the form in the VWorks window. Optionally, the view can include the Main Log, Pipette Log, Time Constraints Log, and Progress tabs. All other controls and areas, such as menus, toolbars, and the work area are hidden. • *Full screen off*. Displays the form as a tab in the protocol area of the VWorks window. **IMPORTANT** The full-screen view is an option only if the form includes the Toggle Full Screen button. **IMPORTANT** If you click the  button in either view, you will exit the VWorks software.

The following figures show examples of the full screen on and off.

Figure VWorks window with example form in full screen view

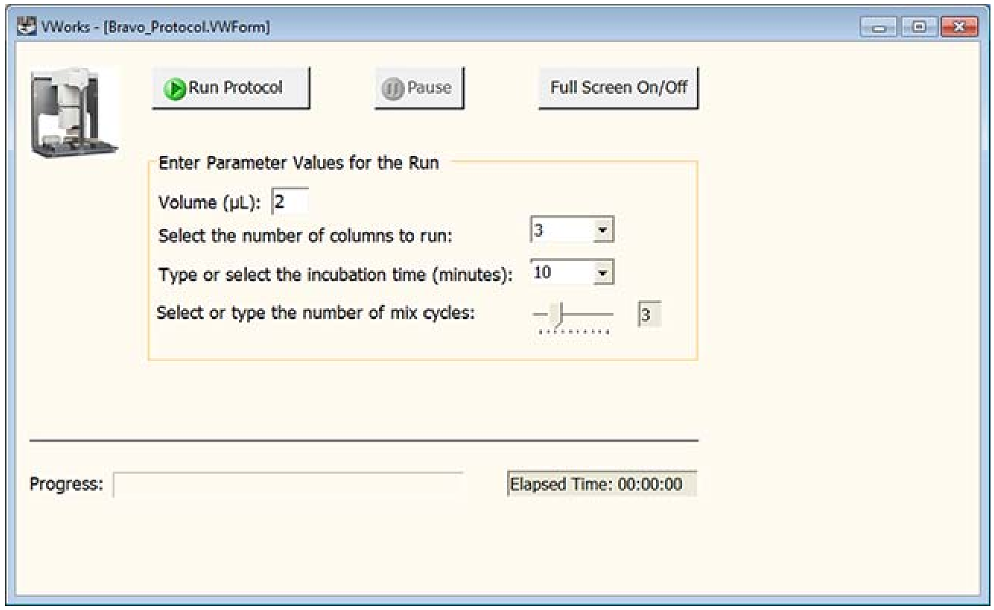
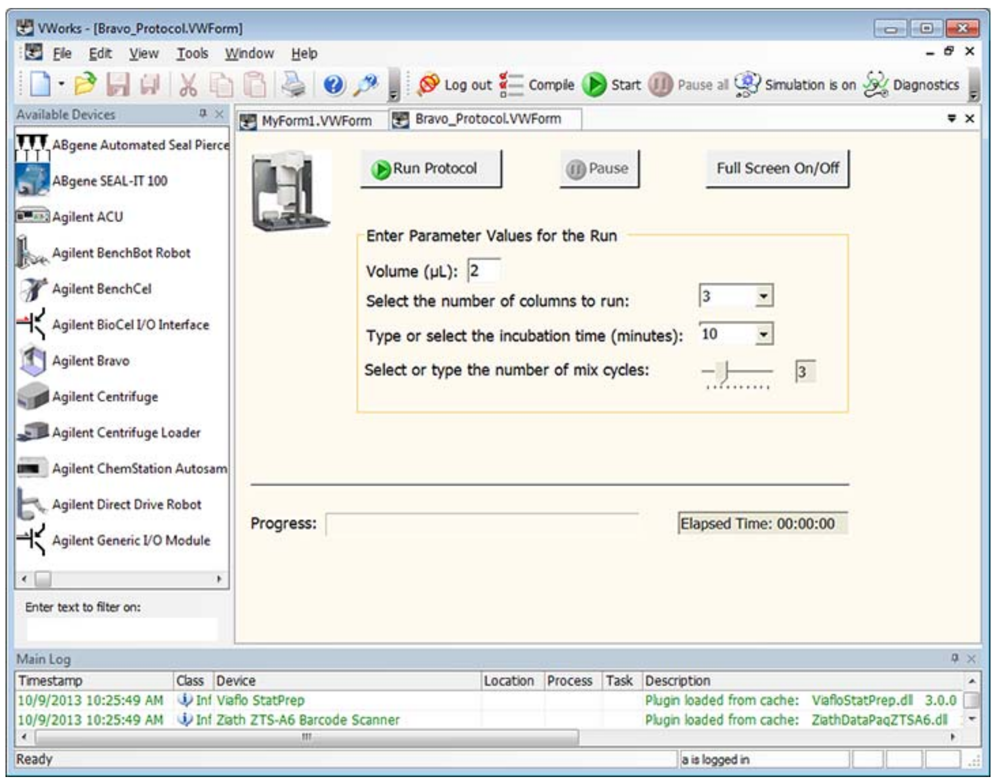


Figure VWorks window with example form in tab view (full screen off)



About including other specialized buttons

In addition to the run buttons and the Pause and Full Screen on/off buttons, the Form Designer provides the following specialized buttons that you can include in your form:

- Initialize All Devices button

- Buttons for data entry values (print, reset, and save)
- Menu Action button

Initialize All Devices button

If the simulation mode is turned off when you open a protocol, the software automatically loads the device file associated with the protocol and asks you whether you want to initialize the devices. If you want the form user to be able to initialize the devices manually while the form is displayed in full-screen view, you can include the Initialize All Devices button. This button is available under Specialized Buttons in the Form Designer window, and the button performs the same function as the Initialize all devices button in the Devices tab of the VWorks window.

Buttons for data entry values

The Data entry property, which enables form users to enter or edit values, is available for many of the Form Designer controls found under General Controls. The following buttons manage the data entry values that are entered by a form's user.

Button	Description
Print Data Entry values	Prints all data entry values associated with the controls in the form, including values entered by the form user and default values that were not modified.
Reset Data Entry values	Resets all data entry values associated with the controls in the form to the default values. Alternatively, values can be reset automatically when the form is reopened if the Restore default values when loading check box is selected in the Form Designer window when creating the form.
Save Data Entry values	Saves the data entry values associated with the controls in the form so that the values persist until they are explicitly reset.

Menu Action button

IMPORTANT The Menu Action button gives a user who has VWorks Operator-level access the ability to perform actions that normally require Technician-level access.

You can use the Menu Action button to perform the following commands.

Menu Action button options	For details, see...
• Display Gantt Charts • Labware Editor • Liquid Library Editor • Pipette Technique Editor • System State Editor • Tip State Editor • Configure Labware • Backup, Restore (Backup Manager) • Log Management • Migrate All Files in a Folder • Error Library • Inventory Editor • Global Options (Options command in the Tools menu)	These commands are also available in the Tools menu of the VWorks window. For a description of the Tools menu commands, see "Tools menu" on page 666. **IMPORTANT** These commands normally require VWorks Technician-level access.
• Export (exports the form and all associated files)	The Export command is also available in the File menu. For details on the Export command, see "Exporting and importing protocols and associated components" on page 626.
• Hit Picking Format wizard–Bravo • Hit Picking Format wizard–Labcyte Echo • Toggle Watcher	These commands are also available in the Tools menu of the VWorks window. For a description of the Tools menu commands, see "Tools menu" on page 666. **IMPORTANT** These commands normally require VWorks Technician-level access.

Configuring the specialized buttons

To configure the specialized buttons:

1. If the button you are configuring is already in the **Form Canvas** area of the Form Designer window, go to step 4.

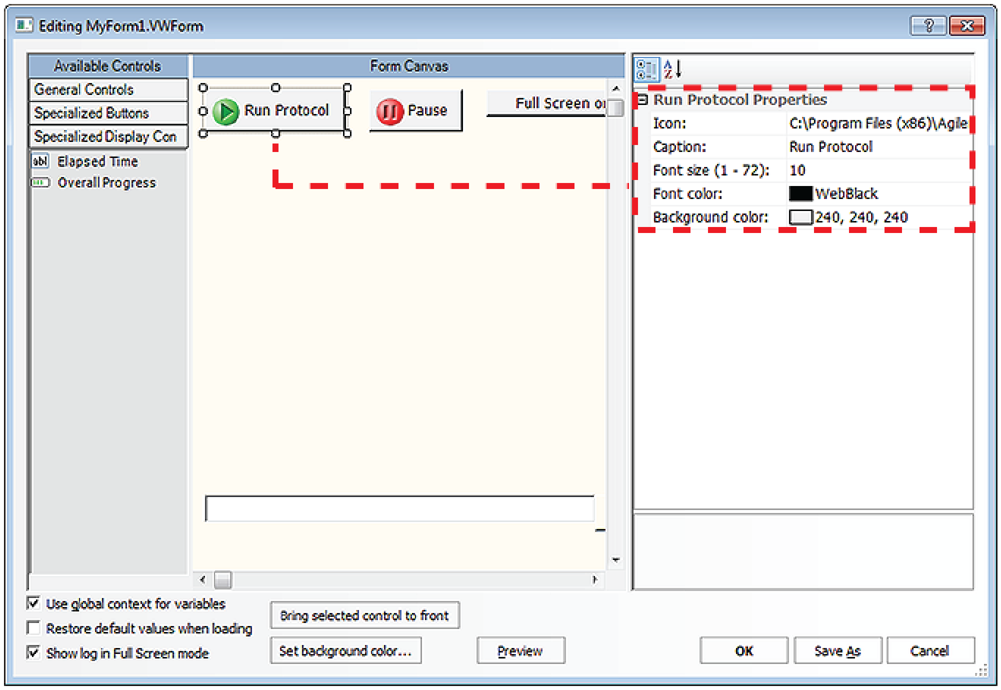
   If the desired button is not yet in the form, click **Specialized buttons**, and drag the button to the **Form Canvas** area.

   An invisible autogrid assists you in aligning the control in the form. To turn off the autogrid, press ALT while dragging the control into position.

2. In the **Form Canvas** area, click the button that you are configuring. A resizing box appears around the button's border. To resize the button, drag one of the sizing handles.

   Note that the corresponding properties appear for the selected button as the following figure shows.

Figure Form Designer window displaying properties for the selected button



Note: Click **Preview** to display the image that the form's user will see.

3 Edit the **Properties** of the button as required. The following table describes all the properties, but a given button may use only a subset of these properties.

Properties	Description
Icon	*Optional.* To select an image to use as an icon in addition to the button caption, click the field, and then click the ... button that appears. In the Open dialog box, locate and select the image file (.jpg, .png, .bmp, or .ico), and then click Open. *Note:* Icons for the run and pause buttons are installed in the following folder: ...\Program Files\Agilent Technologies\VWorks\clipart
Caption	To change the label that appears on the button, type the new text in the field.
Font size	*Optional.* To change the font size for the selected control and subsequently created controls, type a new number in the field. Default: 10

Properties	Description
Font color, Background color (button)	*Optional.* To change the font color for the selected control and subsequently created controls, click the field, and then click the ▼ that appears. In the palette list, select the color, or click Custom Color to open the Select Color dialog box and create a custom color. Default font color: Web Black Default background color: 240, 240, 240
Run Specified Protocol button only	
Protocol file	To select the protocol that this form will run, click the field, and then click the ... button that appears. In the Open dialog box, locate and select the .pro file, and then click Open.
Number of times to run	Type the number of times the protocol will run after the form user clicks the run button.
Run Runset button only	
Runset file	To select the runset that this form will run, click the field, and then click the ... button that appears. In the Open dialog box, locate and select the .rst file, and then click Open.
Menu Action button only	
Action	Click the field, and then click the ▼ that appears. Click the command that you want the button to perform. **IMPORTANT** The menu action button gives a user who has Operator-level access the ability to perform the specified action, which may normally require Technician-level access.
Initialize All Devices button only	
Device file	To select the device file that contains the devices for the corresponding protocol, click the field, and then click the ... button that appears. In the Open dialog box, locate and select the .dev file, and then click Open.

4 If you are configuring the **Run Protocol** button Run Protocol:

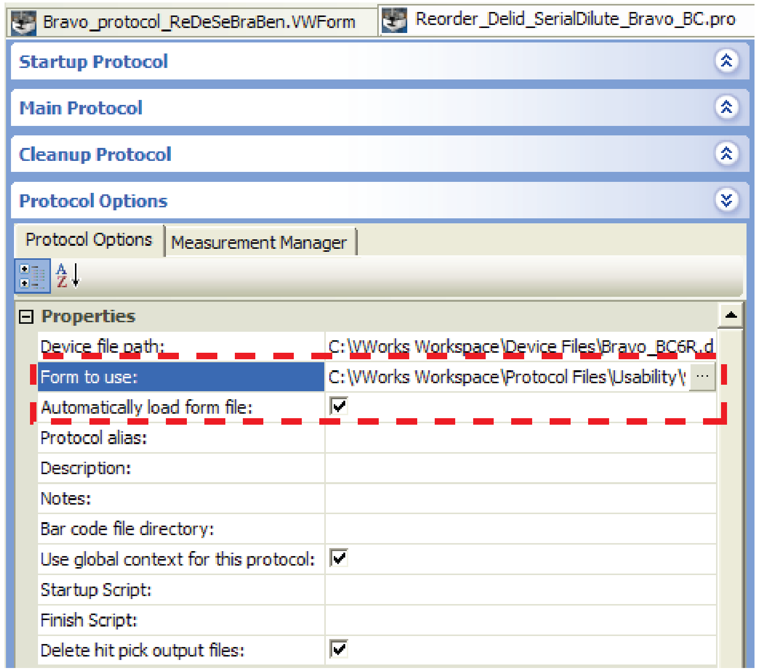
a In the **VWorks** window protocol tab, under **Protocol Options > Properties**, click the **Form to use** field, and then click the ... button that appears.

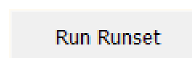
b In the **Open** dialog box, locate and select the form file that you are creating. (.VWForm), and then click **Open**.

c In the **Automatically load form file** field:

- Ensure the check box is selected (default) if the form should open automatically anytime the protocol is opened.

- Clear the check box if the form should not open automatically when the protocol is opened.



5 If you are configuring the **Run Runset** button , make sure you select the **Use global context for variables** check box in the **Form Designer** window.

If you want the JavaScript variables used by the form's controls to interact with variables in the protocols, ensure that the **Use global context for this protocol** check box is selected in the **Protocol Options** for each protocol.

If one of the protocols uses its own private context (the **Use global context for this protocol** check box is not selected), you can include a `getGlobalObject()` JavaScript call in each case where a variable in the protocol must read from, or write to, one of the variables used by the form. For details, see "VWorks-defined functions" on page 87.

Note: If you want one control to overlap another control on the form, select the control that you want on top, and then click **Bring selected control to front**.

Related information

For information about...	See...
Workflow for creating or editing a form	"Workflow for creating or editing a form" on page 160
Forms and Form Designer overview	"About forms for running protocols" on page 154
Controls under General Controls in the Form Designer window	"Adding form controls that allow editing or runtime data display" on page 176

For information about...	See...
Controls under Specialized Display Controls in the Form Designer window	"Adding indicators for elapsed time and progress to a form" on page 173
Form properties in the Form Designer window	"Setting the form properties" on page 201
Creating a basic protocol	"Workflow for creating or editing a form" on page 160

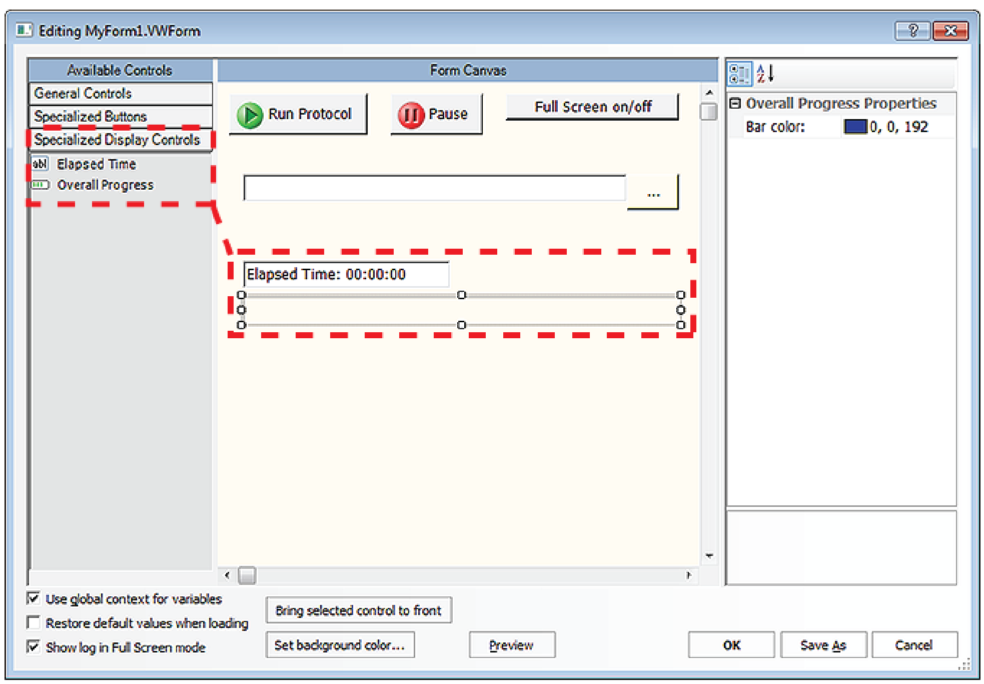
Adding indicators for elapsed time and progress to a form

About this topic

To create or edit forms, you must have VWorks Technician- or Administrator-level access. You should also have an understanding of how to create protocols in the VWorks software.

This topic describes the controls in the Specialized Display Controls area of the Form Designer window and how to configure them.

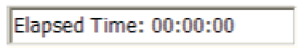
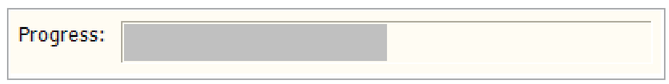
Figure Form Designer window and the Specialized Display Controls



For more details on creating forms, see "Workflow for creating or editing a form" on page 160.

About the specialized display controls

The Form Designer has two controls that display standard data for the protocol or runset. The data for these controls are not drawn from JavaScript variables.

Specialized display controls	Description
Elapsed Time	Displays the hours:minutes:seconds of duration for corresponding protocol or runset that the form is running. 
Overall Progress	Provides a visual indication of the percent complete for the corresponding protocol or runset that the form is running. The following figure shows a Progress bar with a Static text control for the *Progress* caption. In this example, a gray bar shows the real-time progress. 

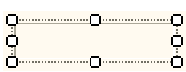
Procedure

To add indicators for elapsed time and progress:

1. In the **Form Designer** window under **Available Controls**, click **Specialized display controls**, and then drag the desired control to the **Form Canvas** area.

   An invisible autogrid assists you in aligning the control in the form. To turn off the autogrid, press ALT while dragging the control into position.

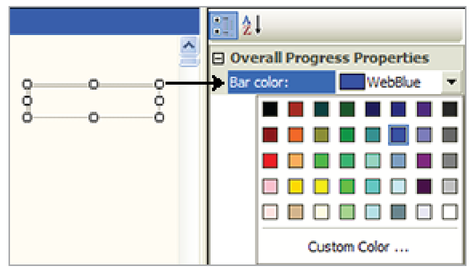
2. In the **Form Canvas** area, click the control so that a resizing box appears around the border.

   To resize the control, drag one of the sizing handles.

3. Edit the control **Properties** as required. The following table describes the properties.

Properties	Description
Font size	*Elapsed time only*. To change the font size, type a new number in the field. Any controls that you create subsequently will use the new font size. Default: 10

Properties	Description
Font color or Bar color	*Optional*. To change the color for the selected control and subsequently created controls, click the field, and then click the that appears. Click the color in the palette list, or click **Custom Color** to open the Select Color dialog box and create a custom color. 

Note: To create a caption for the Overall Progress control, you can combine the Overall Progress control with the Static text control. See "About the General Controls" on page 176.

Note: To have one control overlap another control on the form, select the control that you want on top, and then click **Bring selected control to front**.

Related information

For information about...	See...
Workflow for creating or editing a form	"Workflow for creating or editing a form" on page 160
Controls under General Controls in the Form Designer window	"Adding form controls that allow editing or runtime data display" on page 176
Controls under Specialized Buttons in the Form Designer window	"Configuring a run button and other specialized buttons in a form" on page 164
Form properties in the Form Designer window	"Setting the form properties" on page 201
Creating a basic protocol	"Workflow for creating or editing a form" on page 160

Adding form controls that allow editing or runtime data display

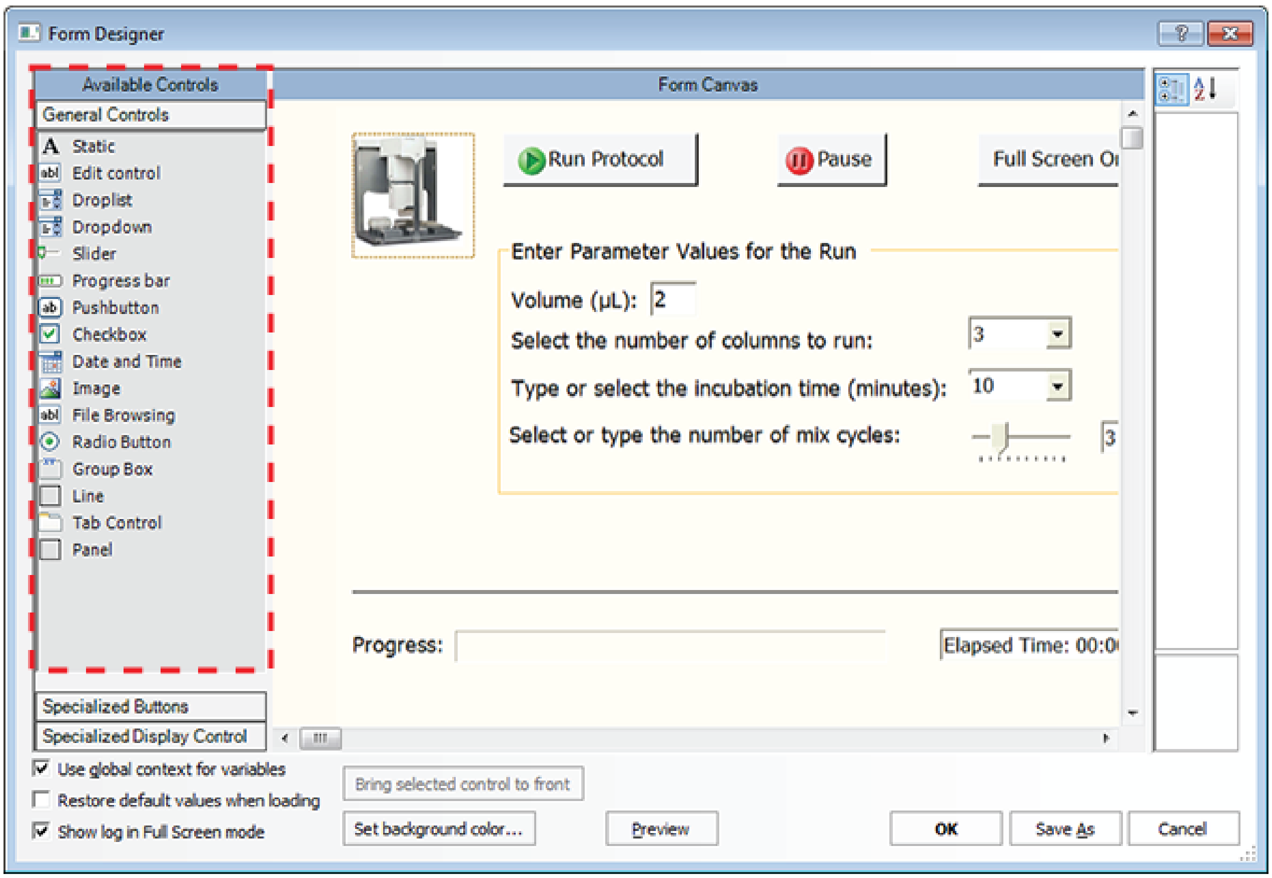
About this topic

To create or edit forms, you must have VWorks Technician- or Administrator-level access. You should also have an understanding of how to create protocols and define JavaScript variables in the VWorks software.

This topic describes the General Controls in the Form Designer window and how to configure them:

- "About the General Controls" on page 176
- "Configuring the General Controls" on page 181

Figure Form Designer window General Controls and example form

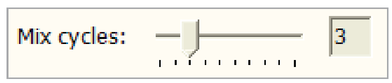
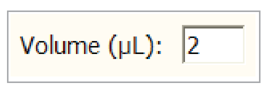
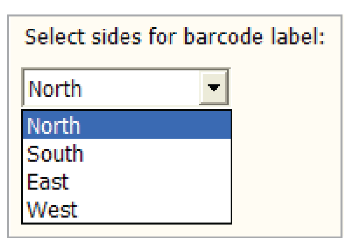


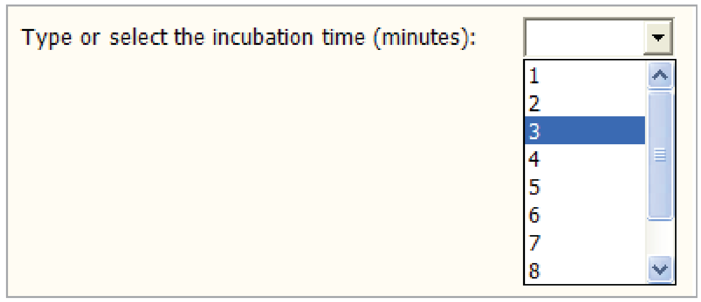
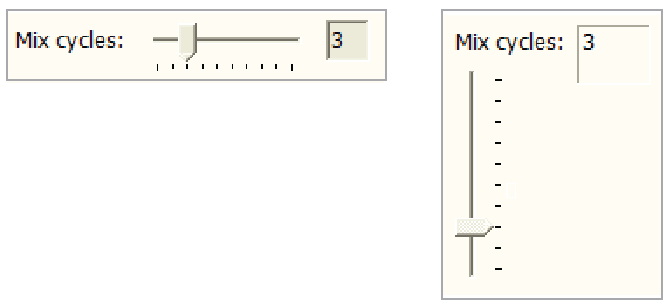
For more details on creating forms, see "Workflow for creating or editing a form" on page 160.

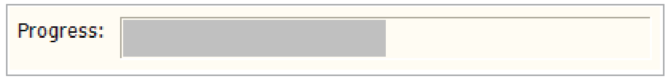
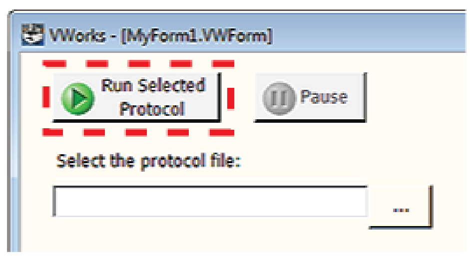
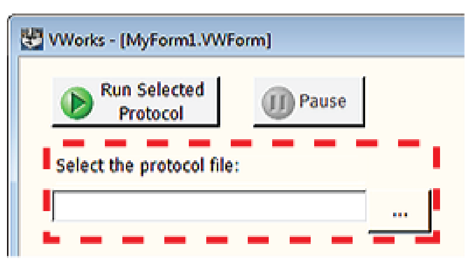
About the General Controls

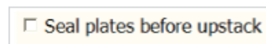
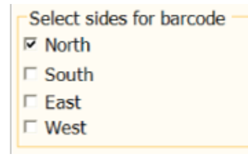
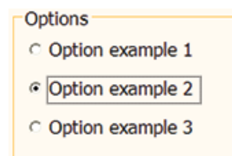
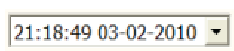
The controls available under General Controls can be assigned JavaScript variables to allow users to edit data or to display runtime data on the form. For example, you might want the form to display the volume dispensed or to allow the form's user to enter the number of mix cycles. The JavaScript variable assigned to the control must also be assigned to the given task parameter value.

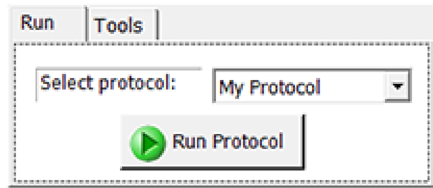
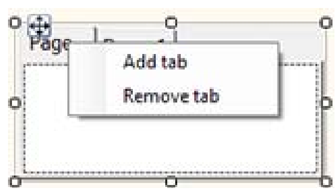
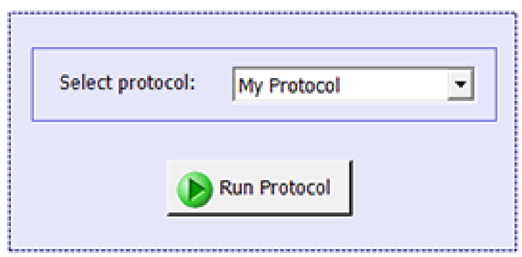
The general controls include the following:

Control	Description
Static text	Displays read-only text. Use this control to create: • Captions for other controls on the form, such as drop-down lists, progress bars, sliders, and edit controls. • Read-only displays for runtime data. The JavaScript variable must be assigned in the Static control properties and in the corresponding task parameter in the protocol. The following figure shows an example that consists of two Static text controls in combination with a Slider control. A Static text control (left) provides the caption, *Mix cycles,* and a Static text control (right) displays the value of the current slider setting. 
Edit control	Allows form users to enter or edit data. The JavaScript variable must be assigned in the Edit control properties and in the corresponding task parameter in the protocol. The following figure shows an Edit control (right) that allows the form users to edit the volume. This control has an assigned JavaScript variable (`vol`), which is also assigned in the corresponding task parameter in the protocol.  *Note:* A Static text control (left) provides the caption, *Volume (uL).*
Droplist	Provides a drop-down list of choices. In the following figure the Droplist control has an assigned JavaScript variable (`BCside`) that is also specified in the protocol task parameter for defining the barcode side of the labware.  *Note:* A Static text control provides the caption for the Droplist control.

Control	Description
Dropdown controls	Provides a drop-down list of choices and allows users to type arbitrary text into the edit field. The Dropdown control in the figure has an assigned JavaScript variable (`time`) that is also specified in the protocol task parameter for defining the incubation time.  *Note:* A Static text control provides the caption for the Dropdown control.
Slider	Enables users to set a value on a continuous range of possible values. The following figure shows a horizontal and vertical slider, where the position of the slider is coordinated with a Static text control (left) or Edit control (right).  In the horizontal example (left), the user moves the slider to set the Mix task parameter value, while the value updates in the display. In the vertical example (right), the user can type a number in the box or move the slider to set the Mix task parameter value.

Control	Description
Progress bar	Provides a visual indication of task progress. The JavaScript variable must be assigned in the control properties and in the corresponding task parameter in the protocol. In the following figure, the gray bar shows how much of the task has been completed.  *Note:* A Static text control provides the caption, *Progress*. *Note:* To display the progress for the entire protocol or runset that the form is running, use the Overall Progress bar under Specialized Display Controls.
Pushbutton	Allows you to write your own script for the action that the button will perform. The following figure shows a form with a custom pushbutton for running a protocol that the form's user selects in a File Browsing control. The script specified in the Pushbutton Properties calls the file path defined as the JavaScript variable for the File Browsing control.  *Note:* For more details on how to set up this example, see "Example: Creating a scripted Pushbutton control in a form" on page 197.
File Browsing	Allows users to locate and select the specified file type. In the following figure, the File Browsing control has an assigned JavaScript variable for the file path (full file name and type), which is called by the script associated with the custom pushbutton that runs the selected protocol.  *Note:* A Static text control provides the caption, *Select the protocol file*.

Control	Description
Checkbox	Allows users to enable or disable the corresponding actions. The JavaScript variable must be assigned in the Checkbox control properties and in the corresponding task parameter in the protocol. The following figure shows two examples.  If the user selects the check box, the assigned JavaScript variable is set to 1. If the user clears the check box, the assigned JavaScript variable is set to 0.
Radio button	Presents choices from among several items. Each Radio button control has a JavaScript variable that is also assigned in the corresponding protocol task parameter.  If a user selects an option, the assigned JavaScript variable is set to 1. If the user clears the option, the assigned JavaScript variable is set to 0.
Date and Time	Displays a preset date and time or a date that the user can set. To correspond with a given task, the JavaScript variable must be assigned in the control properties and in the corresponding task parameter in the protocol.  *Note:* You can use the Elapsed Time indicator for the entire run. See "Adding indicators for elapsed time and progress to a form" on page 173.
Group box	Displays a rectangle that can be resized and labeled to visually group other controls together. To correspond with a given task, the JavaScript variable must be assigned in the control properties and in the corresponding task parameter in the protocol.
Image	Allows you to place a static image on the form. To correspond with a given task, the JavaScript variable must be assigned in the control properties and in the corresponding task parameter in the protocol.
Line	Provides a vertical or horizontal line that you can use to visually separate items on the form.

Control	Description
Tab Control	Displays tabbed pages in the form. You can add different controls and information in each tabbed page. ***Figure*** Tabbed pages  By default, when you drag a Tab Control onto the Form Canvas, the software displays two tabbed pages. To add another tabbed page, right-click the tab area, and select **Add tab**. To remove a tabbed page, right-click the tab, and then select **Remove tab**. To change the tab names, see “Setting Tab Control properties” on page 195. ***Figure*** Adding or removing tabbed pages 
Panel	Displays a container in which you can add controls and information. You can add one or more panels in a form to group or categorize different controls. ***Figure*** Panel containing different controls 

Configuring the General Controls

To add a control to the form:

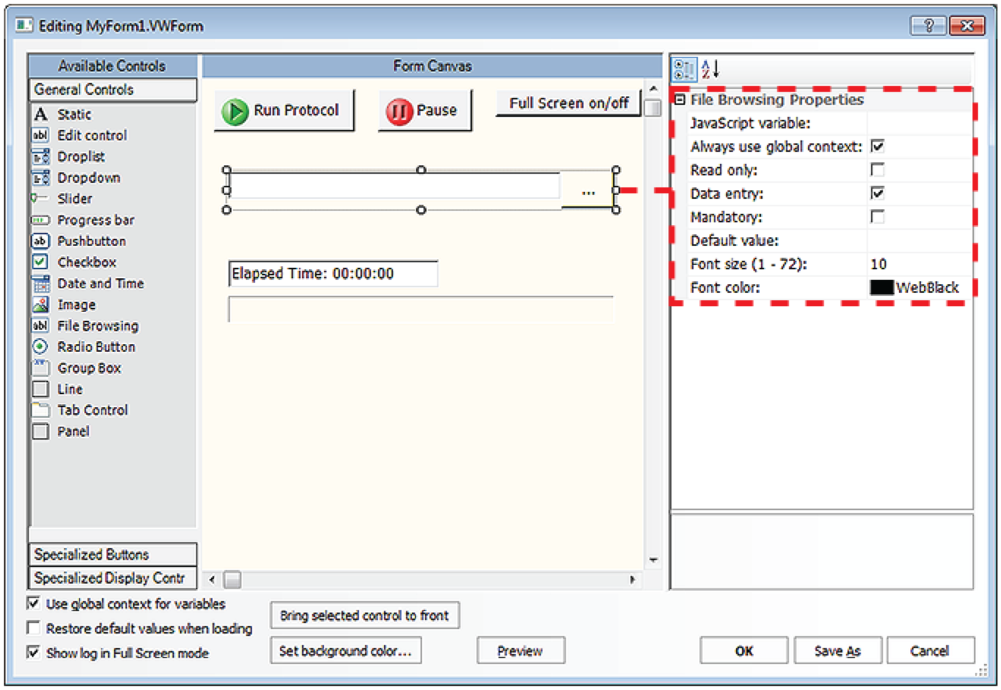
1 Under **Available Controls**, click **General Controls**, and then drag the desired control to the **Form Canvas** area.

An invisible autogrid assists you in aligning the control in the form. To turn off the autogrid, press ALT while dragging the control into position.

2 In the **Form Canvas** area, click the control that you want to set up, so that a resizing box appears around the border. To resize the control, drag one of the sizing handles.

Note that the corresponding properties appear for the selected control as the following figure shows.

Figure Form Designer window displaying properties for the selected control



Note: Click **Preview** to display the image that the form's user will see.

3 Edit the **Properties** for the control. The properties can include a combination of the following depending on the control:

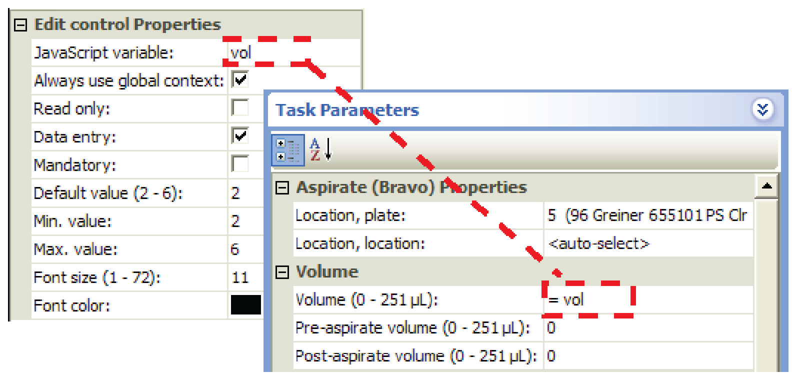
- "Setting properties common to the general controls" on page 183
- "Setting Static text control properties" on page 187
- "Setting Droplist and Dropdown properties" on page 188
- "Setting Slider properties" on page 188
- "Setting Progress bar properties" on page 190
- "Setting Pushbutton control properties" on page 191
- "Setting File Browsing properties" on page 192
- "Setting Checkbox properties" on page 193
- "Setting Radio Button properties" on page 193
- "Setting Image (static) control properties" on page 194
- "Setting Date and Time properties" on page 194
- "Setting Group Box properties" on page 195
- "Setting Tab Control properties" on page 195
- "Setting Panel properties" on page 196

The Line control does have any properties that you can set.

Note: If you want one control to overlap another control on the form, select the control that you want on top, and then click **Bring selected control to front**.

Setting properties common to the general controls

The following table lists the properties for all controls. A given control might use only a subset of the properties listed.

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td>JavaScript variable</td><td>
The JavaScript variable that will be associated with the focused control.
Type any string that qualifies as valid syntax for a JavaScript variable name.
In the protocol, type the variable name for the corresponding Task Parameters property.
For example, if you want the form's user to enter the aspirate volume for a given task, you could use an Edit control that has a JavaScript variable of <code>vol</code>. In the protocol, you would specify the Aspirate Properties for Volume as = <code>vol</code>.


For more details on using JavaScript variables in the Task Parameters area, see “Using simple variables” on page 77.
</td></tr>
</table>

<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td>Always use global context</td><td>If the form’s Use global context for variables check box is selected, you can ignore this setting. If the form’s Use global context for variables check box is not selected, specify which context to use for the control’s JavaScript variable:

Global context. Select this check box.
For example, if the variable in question will be acted on before the protocol is running or is scheduled to run, use the global context.

IMPORTANT Any protocol that uses the global context and has JavaScript variables of the same name will be affected by changes you make to this variable, even if this form’s Use global context for variables check box is not selected.
Context of the running protocol. Clear this check box.
For example, if the variable in question will communicate with the protocol once the protocol is running or is scheduled to run, you may not need to use the global context.

For more details on JavaScript context, see “Understanding JavaScript context in form design” on page 204.</td></tr>
<tr><td>Read only</td><td>The state that determines whether the form's user can edit the contents of the control.

Select the check box to prevent users from editing the control contents.
Clear the check box to allow users to edit the control contents.
</td></tr>
<tr><td>BorderStyle</td><td>The border surrounding the control.
You can specify one of the following border styles:

FixedSingle. Adds a single line that encloses the control.
Fixed3D. Adds a 3-dimensional effect to the border’s appearance.
</td></tr>
<tr><td>BackgroundImage</td><td>The image displayed as the background for the control. The selected image is displayed on top of the selected background color.
Click the browse button to locate and select the desired image.</td></tr>
</table>

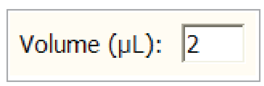
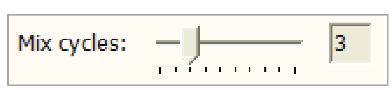
Properties	Description
BackgroundImageLayout	How the selected image is displayed. Select one of the following display methods: • *Center*. Center-aligns the image on the control. • *Stretch*. Stretches the image to fit the size of the control. The dimensional ratio of the image is not maintained. • *Tile*. Uses multiple copies of the image to cover the entire control area. The images are in their original sizes and are placed adjacent to one another. • *Zoom*. Magnifies the image such that the width or length covers the width or length of the control. The dimensional ratio of the image is maintained.
Background_Color	The color of the control. Click the field, and then select the desired color.
Data entry	The option to allow the user to edit the value in the control when the protocol is paused. • Select the check box to allow form users to edit the value when a protocol is not running and when a running protocol is paused. • Clear the check box to allow users to edit the value only when a protocol is not running. For the data entry controls, you can also provide the Print, Reset, and Save data entry buttons from the Specialized Buttons area. For details, see "Configuring a run button and other specialized buttons in a form" on page 164. *Note:* The Data entry property is not available if the Read only check box is selected.
Mandatory	The option to require the form's user to enter a value for this field before starting the associated protocol or runset. Select the check box to require the user to enter a value for the field. Clear the check box if the user can start the protocol or runset without entering a value for the field. Available if the Data entry check box is selected.
Default value	The value the form should display when the user resets the field. Type the default value in the box. Special conditions apply for some controls, such as check boxes and radio buttons. For details, see the property description for the specific control.

Properties	Description
Min value, Max value	A range of permitted values. The value saved to the associated JavaScript variable will be a number. Type the minimum and maximum value to define the range of values. If the form's user types a value outside this range, an error message appears. To allow the user to type any value in an Edit control, leave the Min and Max value properties blank. The value saved to the associated JavaScript variable will be a number, unless the value appears to be something other than a number. In which case, the value will be saved as a string. Special conditions apply for some controls, such as sliders. For details, see the property description for the specific control.
Dock	The option to anchor the control to a position on the Form Canvas. Select one of the following positions: • *None*. You can move the control freely on the canvas. Select None if you want to manually position the control, and you prefer to set the size of the control using the Width and Height properties. • *Top*. The top side of the control is adjacent to the top of the Form Canvas. The control's width is increased to match the width of the canvas. Although the width is fixed, you can change the Height value. • *Left*. The left side of the control is adjacent to the left side of the Form Canvas. The control's height is increased to match the height of the canvas. Although the height is fixed, you can change the Width value. • *Bottom*. The bottom side of the control is adjacent to the bottom of the Form Canvas. The control's width is increased to match the width of the canvas. Although the width is fixed, you can change the Height value. • *Right*. The right side of the control is adjacent to the right side of the Form Canvas. The control's height is increased to match the height of the canvas. Although the height is fixed, you can change the Width value. • *Fill*. The control's size is increased so that it completely covers the Form Canvas area. The width and height of the control are fixed.
Font size	The font size of the text that appears in the control. To change the font size, type a new number in the field. Any controls that you create subsequently will use the new font size. Default: 10

Properties	Description
Font color	The color of the text that appears in the control. To change the color for the selected control and subsequently created controls, click the field, and then click the that appears. Click the color in the palette list, or click Custom Color to open the Select Color dialog box and create a custom color. Default font color: Web Black Default background color: 240, 240, 240

Setting Static text control properties

The Static text control has the following properties in addition to the properties common to the other general controls.

Properties	Description
Display string	Type the text that you want to appear on the form, for example, a caption for another control on your form. The following figure shows an example of a Static text control with a display string that provides the label, *Volume (uL)* for an Edit control.  *Note:* Display string is enabled only if the JavaScript variable property is blank. *Note:* To use the symbol µ, copy the characters to the Microsoft Clipboard from other software. Right-click the field and choose Paste to insert the character into the field.
Border	To add a border around the control, ensure the check box is selected (default). For example, a border is useful if you want to provide a display of runtime data. To remove the border, clear the check box. For example, if you are creating a caption for one of the other controls, a border is unnecessary. The following figure shows an example of two Static text controls combined with a Slider control. The caption on the left is a Static control with no border. The display on the right is a Static control with a border. 

To coordinate the Static text control with a protocol task parameter, both the control and the task parameter must use the same JavaScript variable.

To coordinate the Static text control with another control, for example, the slider, both controls must have the same assigned JavaScript variable and both controls must specify the global JavaScript context so that they can communicate with each other.

Setting Droplist and Dropdown properties

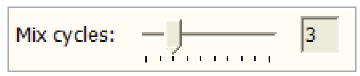
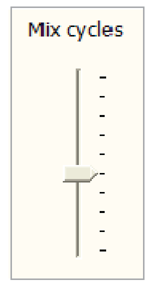
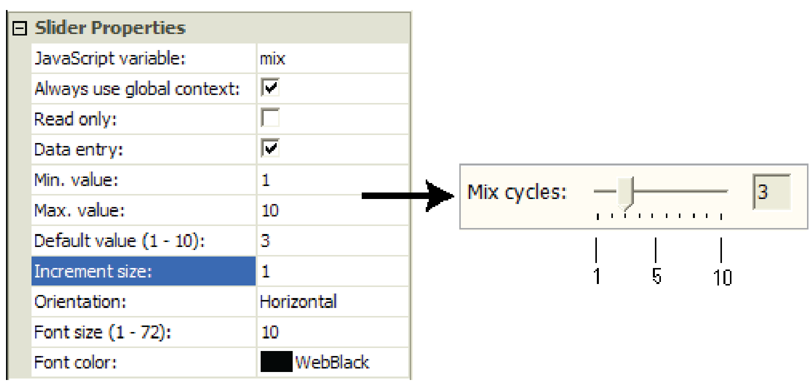
The Droplist and Dropdown controls have the following properties in addition to the properties common to the other general controls.

Note: You might want to use these controls in combination with a Static text control to display a caption for your drop-down list or drop-down combo box.

Properties	Description
Items	Type the list of items that will appear in the list of the Droplist or Dropdown control, where each item is separated by a semicolon with no spaces. For example, type 3;6;9;12 to provide the form's user with the choices 3, 6, 9, 12.
Default value	Type the item from Items list that will display in the form by default. **IMPORTANT** Make sure that the default value is also in the Items list. Otherwise, the Droplist control will display the first list item by default. The Dropdown control will display the default value even if the value is not included in the Items list.

Setting Slider properties

The Slider control has the following properties in addition to the properties common to the other general controls.

Properties	Description
Orientation	Click the field to display the ▾, and then click one of the following: • *Horizontal*. The scale reads from left to right, Min value to Max value. • *Vertical*. The scale reads from bottom to top, Min value to Max value. The figure shows an example of a horizontal and vertical slider. The horizontal slider is combined with an Edit control that displays the changing numeric value as the slider is moved. 
Min value, Max value	To specify the range of values that the slider represents, type a minimum and maximum numeric value. The slider has a tick mark for each numeric value in the range. For example, a slider with a range of 1–10 has 10 tick marks, and a slider with a range of 1–20 has 20 tick marks.  When the form's user moves the slider, the resulting value is written to the control's JavaScript variable as soon as the slider is released.

Properties	Description
Increment size	Type a positive number to use as the increment that the numeric value changes as the slider is moved. For example, if the slider's range is 1–10 and the increment size is 1, the value will change by 1 each time you move the slider. If the increment is 2, the value will change by 2 each time you move the slider. The increment size should be in relation to the slider's range of values. For example, a slider that has a narrow range of values typically would not have a large increment size. Likewise, a small increment size may be insufficient for a slider that has a very wide range of values.

You can use the Slider control in combination with the:

- Static text control to display a caption for the slider or display the current value of the slider's setting.
- Edit control to display the current value of the slider's setting or allow users to enter a value instead of using the slider.

To coordinate the slider setting with a Static text control or Edit control and the protocol task parameter setting:

- Both the Slider and Static or Edit controls must have the same JavaScript variable assigned (for example, `mix`).
- The variable assigned in the Slider and Static or Edit controls must be in the global context so that the two controls can communicate.
- The corresponding Mix task parameter in the protocol must also have the same JavaScript variable assigned (for example, `mix`).

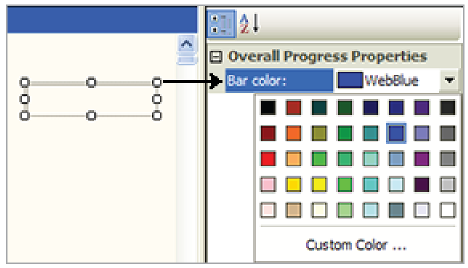
If the slider's Data Entry property is not selected, the slider will move according to the current value of the underlying JavaScript variable. The behavior of the slider is undefined if the JavaScript variable does not have a numeric value, or if the variable's value is outside the slider's range.

Setting Progress bar properties

The Progress bar control has the following properties in addition to the properties common to the other general controls.

Note: You might want to use this control in combination with a Static text control to create the progress bar caption.

Properties	Description
JavaScript variable	Must have a numeric value that is a positive number and the value must not be greater than the Max value setting.
Max value	Type a numeric value larger than 0. The value represented by the left end of the progress bar is always 0. The Max value determines the value represented by the right end of the progress bar.

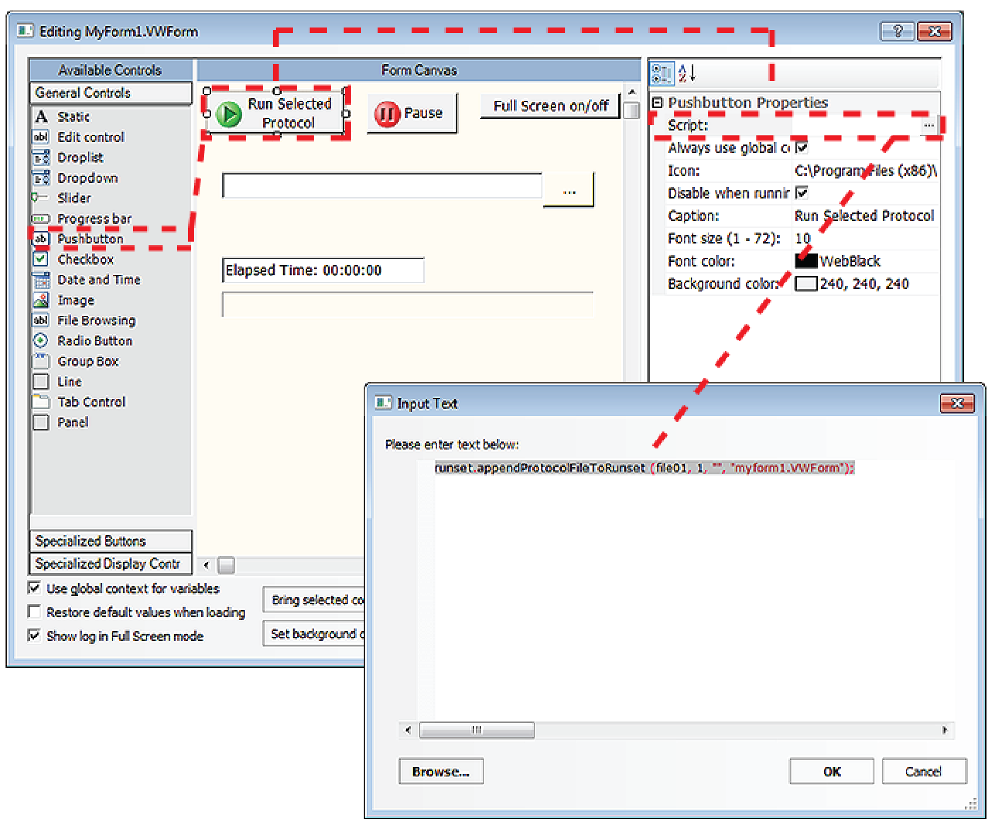
Properties	Description
Bar color	*Optional*. To change the color, click the field, and then click the ▼ that appears. Click the color in the palette list, or click Custom Color to open the Select Color dialog box and create a custom color. 

The progress bar moves according to the current value of the underlying JavaScript variable. All progress bars are horizontal.

Setting Pushbutton control properties

The Pushbutton control can execute any JavaScript code that you specify. You provide the script in the Pushbutton Properties, as the following figure shows.

Figure Workflow to create a pushbutton



In addition to the properties common to the general controls, the Pushbutton control has the following properties. For an example of how to set up a pushbutton using a script, see "Example: Creating a scripted Pushbutton control in a form" on page 197.

Properties	Description
Script	Click the field, and then click the ... button that appears. In the Input Text dialog box, type the JavaScript code to be executed when the pushbutton is pressed, or click Browse to select the file (.txt) that contains the script. *Note:* If the script specifies a file on your computer, and you later export the form with this pushbutton, you must specify the file in the Additional Files page of the Export Wizard.
Always use global context	Specify the context in which to execute the script: • *Global context*. Select the check box to use the global context. For example, if the button is to be pressed while no protocol is running, you must use the global context. **IMPORTANT** Any protocol that uses the global context and contains JavaScript variables of the same name will be affected by changes you make to this variable, even if this form's **Use global context for variables** check box is not selected. • *Context of the running protocol*. Clear the check box to use the protocol's private context.
Disable when running	To make the button unavailable (grayed) whenever the form is running a protocol or runset, select this check box (default). To make the button enabled at all times, clear this check box.
Icon	*Optional*. To select an image to use as an icon in addition to the button caption, click the field, and then click the ... button that appears. In the Open dialog box, locate and select the image file (.jpg, .png, .bmp, or .ico), and then click Open. *Note:* Icons for the run and pause buttons are installed in the following folder: ...\Program Files\Agilent Technologies\VWorks\clipart
Caption	To change the label that appears on the button, type the new text in the field.

Setting File Browsing properties

The File Browsing control has the properties that are common to the other general controls. No additional properties are provided.

For an example of how to set up a File Browsing control, see "Example: Creating a scripted Pushbutton control in a form" on page 197.

Setting Checkbox properties

The Checkbox control has the following properties in addition to the properties common to the general controls.

Properties	Description
Caption	Type the string to be displayed to the right of the check box.
Default value	Type one of the following: 0 — indicates the check box is cleared by default 1 — indicates the check box is selected by default *Note:* Any value other than zero indicates that the check box is selected.

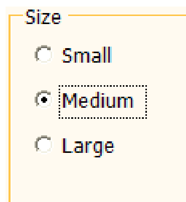
The Checkbox control is binary. When the check box on the form is selected, a 1 is written to the associated JavaScript variable. When the check box is cleared (unchecked), a 0 is written to the associated JavaScript variable.

Note: If the JavaScript variable's numeric value is anything other than 0, the check box will display as checked.

Setting Radio Button properties

The Radio Button control has the following properties in addition to the properties common to the other general controls.

Properties	Description
Default value	Type one of the following: 0 — indicates the option is cleared by default 1 — indicates the option is selected by default *Note:* Any value other than zero indicates that the check box is selected.
Caption	Type the string to be displayed to the right of the radio button.

Properties	Description
Group ID	Type a string to use as the ID of the radio button group. Only one button in the group can be selected at a time. When the form's user selects a button in the group, all other radio buttons with the same Group ID are cleared (not selected) automatically. For example, if you had a grouping of options for size, you could use the string, `size`, as the Group ID.  *Note:* You can use a Group box to visually group a collection of radio button options.

The Radio Button control is binary. When the button on the form is selected, a 1 is written to the associated JavaScript variable. When the button is cleared, a 0 is written to the associated JavaScript variable.

Note: If the JavaScript variable's numeric value is anything other than 0, the option will display as selected.

Setting Image (static) control properties

The Image control has the following properties in addition to the properties common to the other general controls.

Properties	Description
Image file	To select an image to display on the form, click the field, and then click the ... button that appears. In the Open dialog box, locate and select the image file (.jpg, .png, or .bmp), and then click Open.

Setting Date and Time properties

The Date and Time control has the properties common to the other general controls. However, if you provide a default value, it must be in the following format. If you do not provide a default value, the software uses the current date and time from the computer.

Properties	Description
Default value	nn:nn:nn nn-nn-nnnn where, nn:nn:nn is the hour:minutes:seconds in a 24-hour clock and nn-nn-nnnn is the month:day:year

If you select the Data entry property for this control, the user can enter:

- *Time*. The user can edit the default values.
- *Date*. The user can click the drop-down list to select the date from a calendar.

Setting Group Box properties

The Group Box control has the following properties in addition to the properties common to the other general controls.

Properties	Description
Group name	Type the string to display as the caption for the group box.

Controls that you drag inside the group box become part of the group box. The group box may need to be resized to ensure that all the controls within the group box borders are clearly visible.

Setting Tab Control properties

The Tab Control has the following properties in addition to the properties common to the other general controls.

Tab Control properties	Description
Alignment	Specifies the position of the tabs: Top, Left, Right, or Bottom.
SizeMode	Specifies the tab size: • *Normal*. Each tab is sized to fit its name. • *Fixed*. The longest tab name determines the size of all the tabs. • *FillToRight*. The longest tab name determines the size of all the tabs, and the tabs are flush with the right and left margins of the control.
Width	Specifies the width of the tab control.
Height	Specifies the height of the tab control.

IMPORTANT The Dock selection overrides the values you specify for the Width and Height properties. For example, if you dock the control at the top of the canvas, the software increases the width of the control and updates the Width value. To return to the previously set Width and Height values, set the Dock property to None. For more information about docking tabbed pages, see "Setting properties common to the general controls" on page 183.

Each tabbed page contains the following properties in addition to the properties common to the other general controls.

Tabbed page properties	Description
Name	The text that is displayed on the tab. Provide a name that describes the function of the tab contents.

Setting Panel properties

The Panel control has properties that are common to the other general controls. See "Setting properties common to the general controls" on page 183.

Related information

For information about...	See...
Workflow for creating or editing a form	"Workflow for creating or editing a form" on page 160
Pushbutton and File Browsing controls	"Example: Creating a scripted Pushbutton control in a form" on page 197
Controls under Specialized Buttons in the Form Designer window	"Configuring a run button and other specialized buttons in a form" on page 164
Controls under Specialized Display Controls in the Form Designer window	"Adding indicators for elapsed time and progress to a form" on page 173
Form properties in the Form Designer window	"Setting the form properties" on page 201
JavaScript context and form design	"Understanding JavaScript context in form design" on page 204
Creating a basic protocol	"Workflow for creating or editing a form" on page 160
Using JavaScript	"Creating a protocol: advanced topics" on page 73

Example: Creating a scripted Pushbutton control in a form

About this topic

To create or edit forms, you must have VWorks Technician- or Administrator-level access. You should also have an understanding of how to create protocols in the VWorks software.

This topic provides an example of the Pushbutton and File Browsing controls in the Form Designer. For more details, see "Adding form controls that allow editing or runtime data display" on page 176.

About the script for the example

The example presented in this topic uses the following script:

```
runset.appendProtocolFileToRunset (file01, numRunTimes, "",
"myform1.VWForm");
```

This script appends the contents of a protocol file to the Runset Manager, specifies the number of times the protocol should run, and passes the file name of the form (.VWForm) to be associated with the protocol.

The following table describes the arguments for the `appendProtocolFileToRunset` method used in the example script.

Argument	Example	Descriptions
string *ProtocolPath*	`file01`	The protocol file path. In this example, the variable `file01` is used for the file path.
int *RunTimes*	`numRunTimes`	The number of times (integer) to run the protocol. In this example, we use the variable `numRunTimes` so that the form users can edit this value.
string *ProtocolNotes*	`""`	An empty string. Although the third argument is not used in this case, the empty string is required as a placeholder.
string *formToUse*	`"myform1.VWForm"`	The file name of the form to be associated with the protocol once the protocol starts running. The argument must specify the file name (not the path) of the form that contains the pushbutton.

For details on other runset object methods that can be used with forms, see "runset object" on page 123.

Creating the example pushbutton

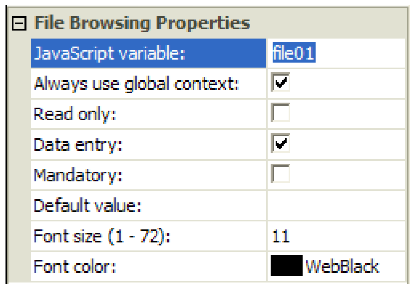
The following example describes how to create a pushbutton that enables the form users to select which protocol to run.

To create the example pushbutton:

1 In the **VWorks** window, choose **File > New > Form**.

  In the **Form Designer** window, click **Save As** and save the form with the name `myform1.VWForm`.

2 To configure the File Browsing control, drag **File Browsing** from **General Controls** to the **Form Canvas**. Enter the following settings under **File Browsing Properties:**

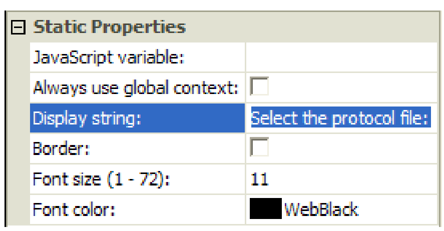


  a In the **JavaScript variable** field, type `file01`.

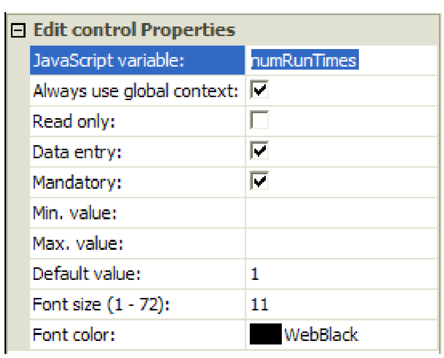
  b Select **Always use global context** and **Data entry**.

3 *Optional*. To provide a caption for the File Browsing control, drag **Static** from **General Controls** to the **Form Canvas**.

  Under **Static Properties**, in the **Display string** field, type the following caption: `Select the protocol file:`



4 To configure an Edit control that allows users to enter the number of runs, drag **Edit control** from **General Controls** to the **Form Canvas**. Enter the following settings under **Edit control Properties:**



  a In the **JavaScript variable** field type `numRunTimes`.

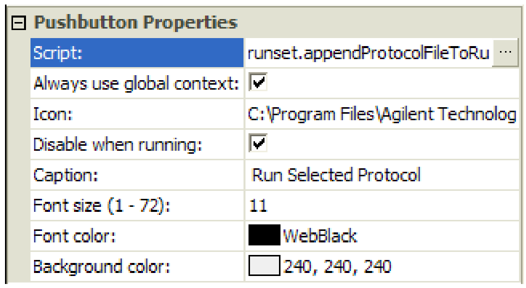
b Select **Always use global context** and **Data entry**.

c Select **Mandatory**. If the users forget to enter the number of runs, the software will display an error message after the user selects the protocol and clicks the pushbutton.

You can also enter a **Default value** that the software will use unless the form user changes the value.

5 *Optional*. Repeat step 3 to create the following caption for the number-of-runs control: `Number of times to run:`

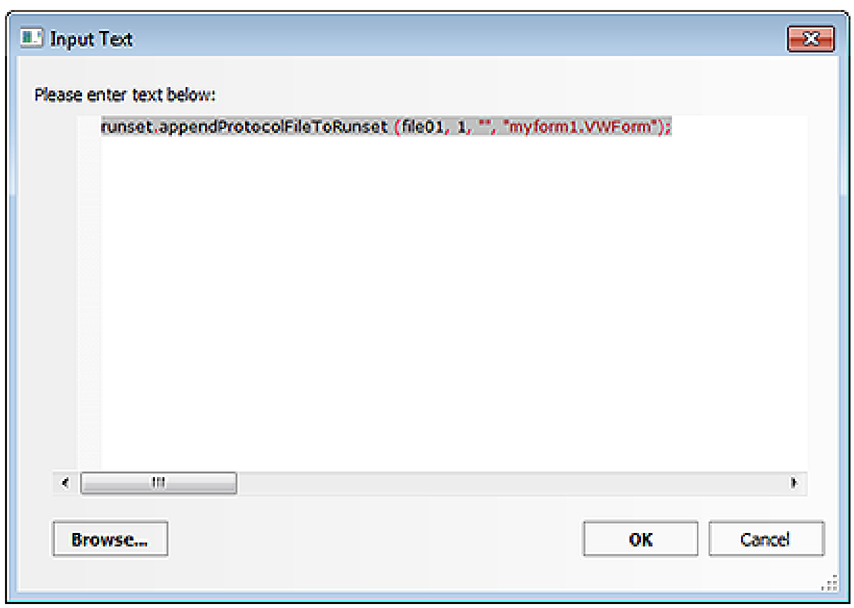
6 To configure the pushbutton, drag **Pushbutton** from **General Controls** to the **Form Canvas**. Enter the following settings under **Pushbutton Properties**:



a Click the **Script** field, and then click the ... button that appears. In the **Input Text** dialog box, type the following script:

```
runset.appendProtocolFileToRunset (file01,
numRunTimes, "", "myform1.VWForm");
```

Note: If you copy this example, make sure to use straight quotation marks instead of curly quotation marks or smart quotes.



b Select **Always use global context** and **Disable when running**.

IMPORTANT The global context is required for the variables in the Pushbutton script, the File Browsing control, and the Edit control because at the time the button is pressed the form is not yet associated with a protocol. Without a protocol association, there is no protocol from which to get an associated protocol's context.

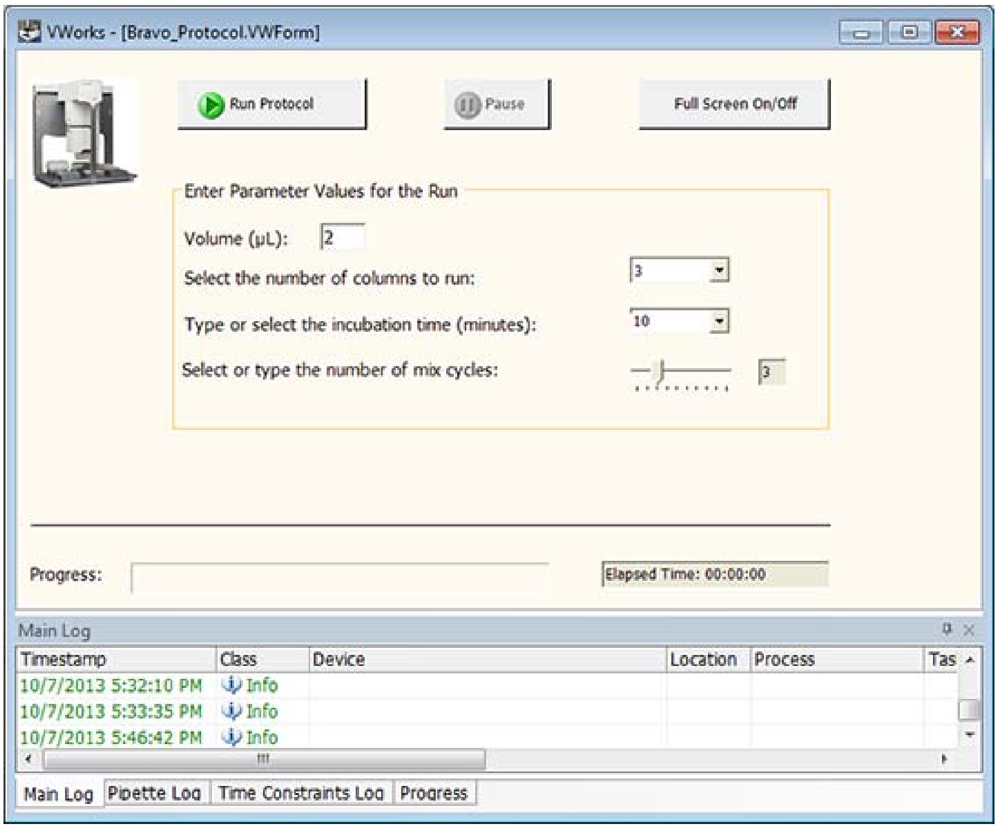
c Type the **Caption** that will appear on the pushbutton, for example, Run Selected Protocol.

d Click the **Image** field, and then click the ... button that appears. In the **Open** dialog box, select the following:

...\Program Files\Agilent Technologies\VWorks\clipart\24b_start_icon_up.ico

7 Click **OK** to close the Form Designer and save the changes.

Figure Example form in full screen view with the logs included



Testing the example pushbutton

To test the example pushbutton:

1 Ensure *myform1.VWForm* is open in the **VWorks** window.

2 In the form, click ... and then select the protocol file (.pro).

3 In the **Number of times to run** box, type a value.

4 Click the custom pushbutton that you created, for example, **Run Selected Protocol**.

Related information

For information about...	See...
Workflow for creating or editing a form	"Workflow for creating or editing a form" on page 160
Controls under General Controls in the Form Designer window	"Adding form controls that allow editing or runtime data display" on page 176
Controls under Specialized Buttons in the Form Designer window	"Configuring a run button and other specialized buttons in a form" on page 164
Controls under Specialized Display Controls in the Form Designer window	"Adding indicators for elapsed time and progress to a form" on page 173
Form properties in the Form Designer window	"Setting the form properties" on page 201
Creating a basic protocol	"Workflow for creating or editing a form" on page 160
Using JavaScript	"Creating a protocol: advanced topics" on page 73

Setting the form properties

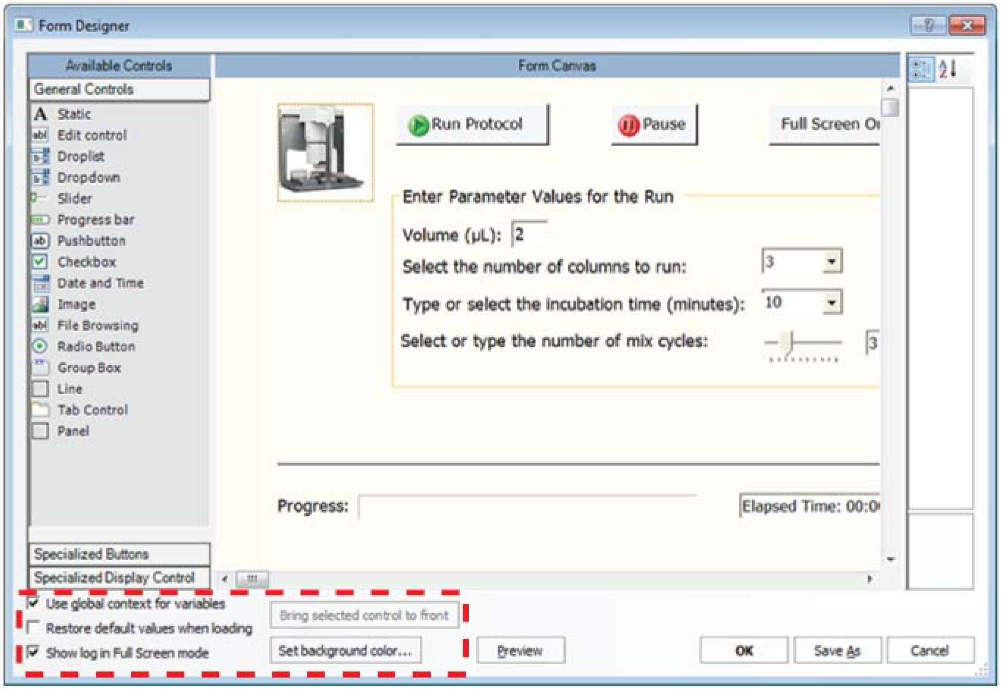
About this topic

This topic describes the properties that appear in the bottom corner of the Form Designer window and the Editing .VWForm windows.

To create or edit forms, you must have VWorks Technician- or Administrator-level access. You should also have an understanding of how to create protocols in the VWorks software.

This topic describes the form properties that appear at the bottom of the Form Designer windows.

Figure Form properties in the Form Designer and Editing .VWForm windows



For more details on creating forms, see "Workflow for creating or editing a form" on page 160.

Procedure

To set the form's properties:

1 In the **Form Designer** window or **Editing .VWForm** window, specify the context for the JavaScript variables associated with all the controls in the form:

- *Context of the running protocol*. Clear the **Use global context for variables** check box. JavaScript variables with the same name in different protocols will not be affected by the properties you set for the controls in this form.

  Individual controls in the form can still specify the global context.

- *Global context for all variables in the form*. Select the **Use global context for variables** check box. For example, if the form is for a runset, you must use the global context.

  All controls in the form will use the global context for their JavaScript variables and scripts, regardless of the settings for the individual control properties.

  Any protocol that uses the global context and contains JavaScript variables of the same name will be affected by the changes you make to the properties for the controls in this form.

  CAUTION Ensure that the variable values you want to use globally are applicable to all other protocols, which also specify the global context. Otherwise, variables with the same name in different protocols will interfere with each other.

For more details on JavaScript context, see "Understanding JavaScript context in form design" on page 204.

2 To ensure that the default values associated with the controls in the form appear each time a user opens the form, select **Restore default values when loading**.

3 If the form includes the Toggle Full Screen button, specify whether to show or hide the Main Log, Pipette Log, Time Constraints Log, and Progress tabs in the full screen view:

- Select the **Show log in Full Screen mode** check box to include the logs.
- Clear the **Show log in Full Screen mode** check box to hide the logs.

4 To change the background color of the form, click **Set background color**, and then select the color from the **Color** dialog box.

5 Click **OK**. If this is the first time to save the form, the Save As dialog box opens. Select a storage location, type a name for the form, and click **Save**. The software saves the form as file type .VWForm.

If you are editing a previously saved form, click **File > Save** after the Form Designer closes.

Related information

For information about...	See...
Workflow for creating or editing a form	"Workflow for creating or editing a form" on page 160
Controls under General Controls in the Form Designer window	"Adding form controls that allow editing or runtime data display" on page 176
Controls under Specialized Buttons in the Form Designer window	"Configuring a run button and other specialized buttons in a form" on page 164
Controls under Specialized Display Controls in the Form Designer window	"Adding indicators for elapsed time and progress to a form" on page 173
JavaScript context and form design	"Understanding JavaScript context in form design" on page 204
Creating a basic protocol	"Workflow for creating or editing a form" on page 160
Using JavaScript	"Creating a protocol: advanced topics" on page 73

Understanding JavaScript context in form design

About this topic

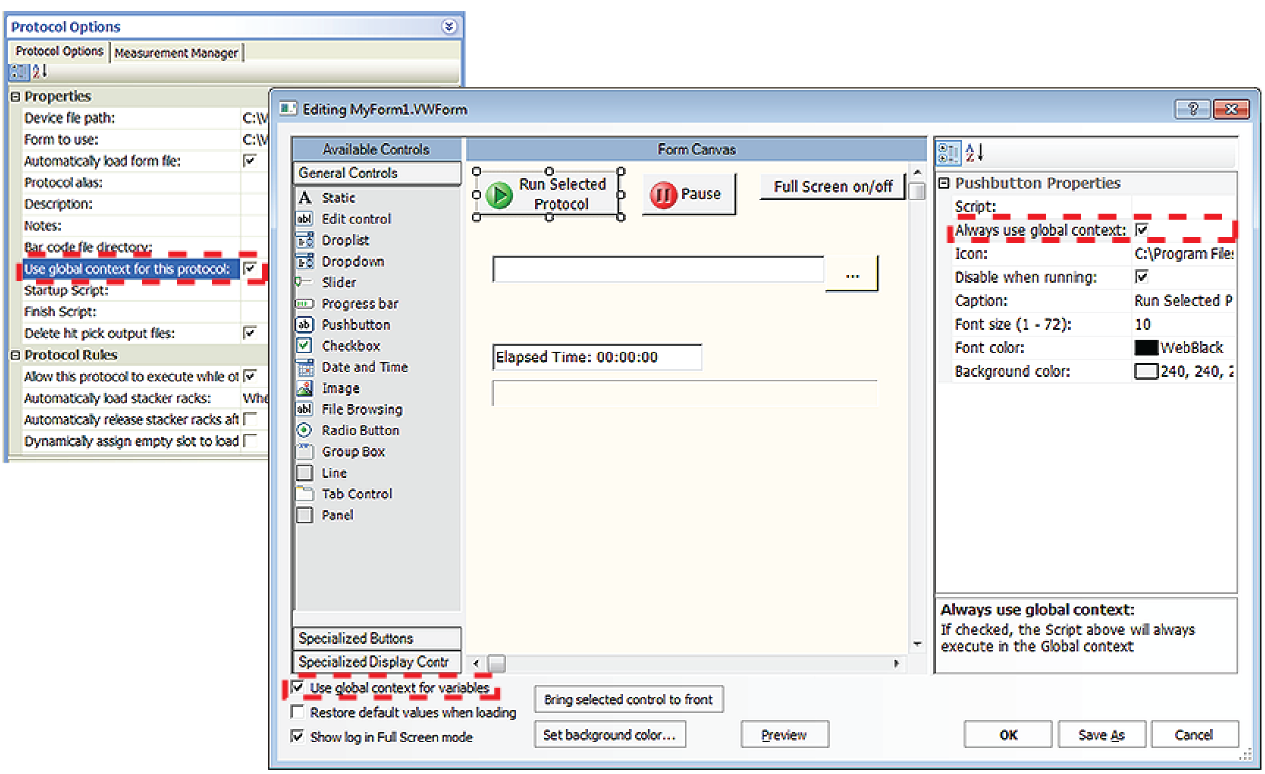
This topic assumes a basic understanding of how to use JavaScript variables and create protocols in the VWorks software.

About JavaScript context and form design

The JavaScript variables used in protocols and in the forms that run the protocols can be in either the global context or the context of the running protocol.

When designing a form, you specify the JavaScript context in three locations:

1. *Protocol*. In the **Protocol Options** area, you select or clear the **Use global context for this protocol** check box to specify the context for the variables in the protocol.
2. *Form*. In the Form Designer window, you select or clear the **Use global context for variables** check box to specify the context for the variables in the form.
3. *Individual controls in the form*. If the form's **Use global context for variables** check box is selected, the context setting for all the controls that use JavaScript is already set. If the form's **Use global context for variables** check box is not selected, you can select the **Always use global context** check box in the control properties for individual controls that use JavaScript.



Protocol's JavaScript context

For each protocol, you specify the context in the Protocol Options area:

- *Global context*. Variables of the same name in the given protocol will be available across all protocols that also specify the global context.
- *Context of the running protocol*. Prevents the variables in a given protocol from being impacted by changes to variables of the same name in other protocols.

Form's JavaScript context

In the Form Designer window, you select or clear the **Use global context for variables** check box to specify the context for the form.

- *Global context*. All variables throughout the entire form will be in the global context.
- *Context of the running protocol*. The variables in each individual control in the form will be in the context of the associated (running) protocol, unless the control specifies the global context. When no protocol association exists (the protocol is not running or scheduled to run), the individual controls that do not specify global context will retain their most recent previous value.

  The global context may not be necessary if the variable in question will communicate with the protocol after the protocol is running or is scheduled to run.

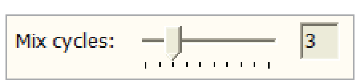
No JavaScript objects that exist in a context other than the global context and the protocol's context will be accessible from either kind of form. For example, plate objects, plates[] array objects, and task objects will not be accessible. If you want the data from any of these objects to be shown on the form, the protocol must contain JavaScript statements at appropriate points to copy the relevant data into JavaScript variables that you devise.

Individual control's JavaScript context

In the Form Designer window, when you set the properties for a control that has an assigned JavaScript variable, you select or clear the **Always use global context** check box to specify the context.

- *Global context*. The variable in question will be available to the protocol and to other controls that also use the global context before the protocol is running or is scheduled to run.

  If two controls on the form must communicate with each other, the global context is required for each control. For example, the following figure shows a Slider control and a display (Static text control). As the slider moves, the value in the display updates to match the slider's position.

- *Context of the running protocol*. The variable in question will be available to the protocol only after the protocol is running or is scheduled to run.

Runsets and the global context

If you have a set of protocols designed to be a runset that will be launched by a form, you use the global context for both the protocols and the form.

Note: Alternatively, you can use the JavaScript function, GetGlobalObject, to access the global context for any variable within the protocol that should interact with the form. See "VWorks-defined functions" on page 87.

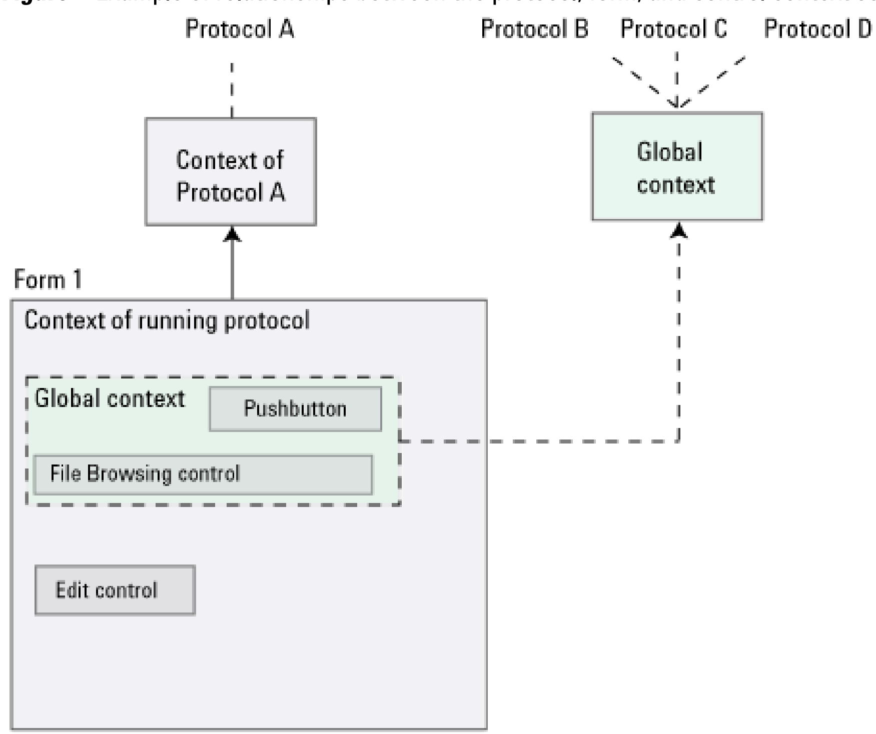
IMPORTANT Multiple protocols designed to run together in the same runset should not reuse the same JavaScript variable names unless the variables are intended to share data.

Context scenarios

The following figure shows four protocols (A, B, C, and D). Form 1 runs protocol A, and protocols B, C, and D run without using a form.

- Protocol A uses its own context for variables.
- Protocols B, C, and D use the global context for variables.
- Form 1 uses the context of the running protocol, except for two of its controls, a pushbutton and a file browsing control, which use the global context.

Figure Example of relationships between the protocol, form, and control context settings



Suppose that protocol A does not use the global context because it uses a variable `x` to count the number of times a loop has executed. But, protocol D also uses a variable called `x` for something completely different. Even though both protocols may run at the same time, their JavaScript variables will not interfere with each other because protocol A uses its own private context.

Suppose that protocols B, C, and D must use the global context because each of these protocols keeps track of the total volume dispensed by all protocols in a variable called `totVol`. Whenever any of these protocols executes a Dispense task, it also executes the JavaScript: "`totVol += task.volume;`". When all three protocols are done, the total volume dispensed in all three protocols can be accessed in a single variable.

In Form 1, the File Browsing control and Pushbutton control are in the global context so that so that the two controls can communicate with each other. The Pushbutton control executes a script that uses a JavaScript variable of `fileName,` which must also be specified by the File Browsing control. Now imagine that the user has typed, `C:\myfile.txt` in this control. When the user presses the Pushbutton control, whatever string happens to be showing in its File Browsing control at that time is assigned to the variable `fileName` in the global context. Now, not only does the script in the Form 1 Pushbutton control recognize the variable `fileName` as having a value of `C:\myfile.txt`, so would all the scripts throughout protocols B, C, and D.

Related information

For information about...	See...
Creating or editing a form	“Workflow for creating or editing a form” on page 160
Controls under General Controls in the Form Designer window	“Adding form controls that allow editing or runtime data display” on page 176
Form properties in the Form Designer window	“Setting the form properties” on page 201
Creating a basic protocol	“Workflow for creating or editing a form” on page 160
Working with JavaScript	“Creating a protocol: advanced topics” on page 73

6
Running a protocol

This chapter contains the following topics:

Workflow for running a protocol

Step	For this task...	See...
1	Turn on the devices that will be used in the protocol and prepare them for operation.	Device user guide
2	Log in to the VWorks software.	"Logging in, logging out, and changing passwords" on page 22
3	Open a form or a protocol.	One of the following: • "Opening a form" on page 211 • "Opening a protocol" on page 213
4	*Optional*. Temporarily disable a device.	"Disabling and enabling a device in the device file" on page 215
5	*Optional*. Set log file directories.	"Setting log file directories" on page 222
6	*Optional*. Set general and view options.	"Setting general and view options" on page 224
7	*Optional*. Set error-handling options.	"Setting error-handling options" on page 229
8	*Optional*. Set up email notification.	"Setting up email notification" on page 232
9	Start the protocol run.	"Starting the protocol run" on page 237
10	*Optional*. Create and manage runsets.	"Managing runsets" on page 243
11	Monitoring the run.	"Monitoring the overall run progress" on page 254
12	*Optional*. Pause the run.	"Pausing the run" on page 261
13	*Optional*. Stop the run.	"Stopping the run" on page 268

Related information

For information about...	See...
Preparing devices for a run	Device user guide
Creating protocols	• "Creating a protocol: basic procedure" on page 13 • "Creating a protocol: advanced topics" on page 73

Opening a form

About this topic

The instructions in this topic assume that you have designed the form and you want to open it and verify that it works before distributing it for use in your lab.

Procedure

To open a form:

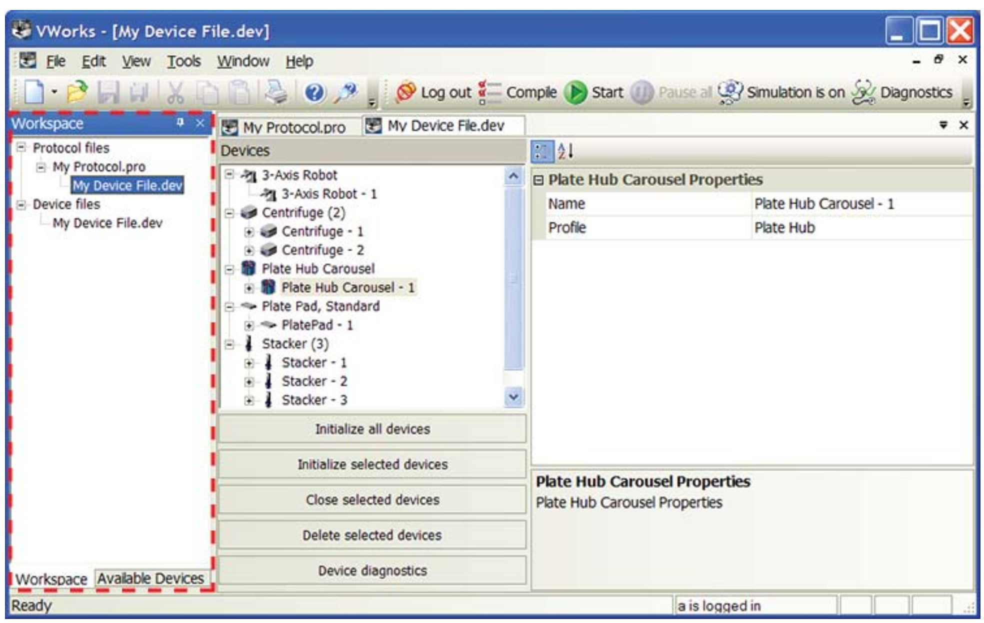
1 Log in to the VWorks software. For instructions, see "Logging in, logging out, and changing passwords" on page 22.

2 Select **File > Open.** The Open dialog box opens.

3 Locate and select the form (.VWForm) file, and then click **Open.**

The form opens in the VWorks window.

Note: A form file is not associated with a protocol or device file until you start a run. If you need to view or update the inventory database before starting the run, you must first open the related protocol or device file. If you want to display the device file, click the **Workspace** tab in the **Available tasks** area, and then click the device file name.



Related information

For information about...	See...
Creating forms	"Creating protocol forms for operators" on page 153

For information about...	See...
Opening a protocol	"Opening a protocol" on page 213

Opening a protocol

About this topic

The instructions in this topic assume that the protocol is associated with the correct device file, profiles, teachpoint files, labware definitions, liquid classes, and other relevant files and databases. For details about how these components relate and impact each other, see “Relationship of VWorks components” on page 4. For instructions on creating or selecting different device files, profiles, and other relevant files, see “Workflow for creating a basic protocol” on page 18.

Procedure

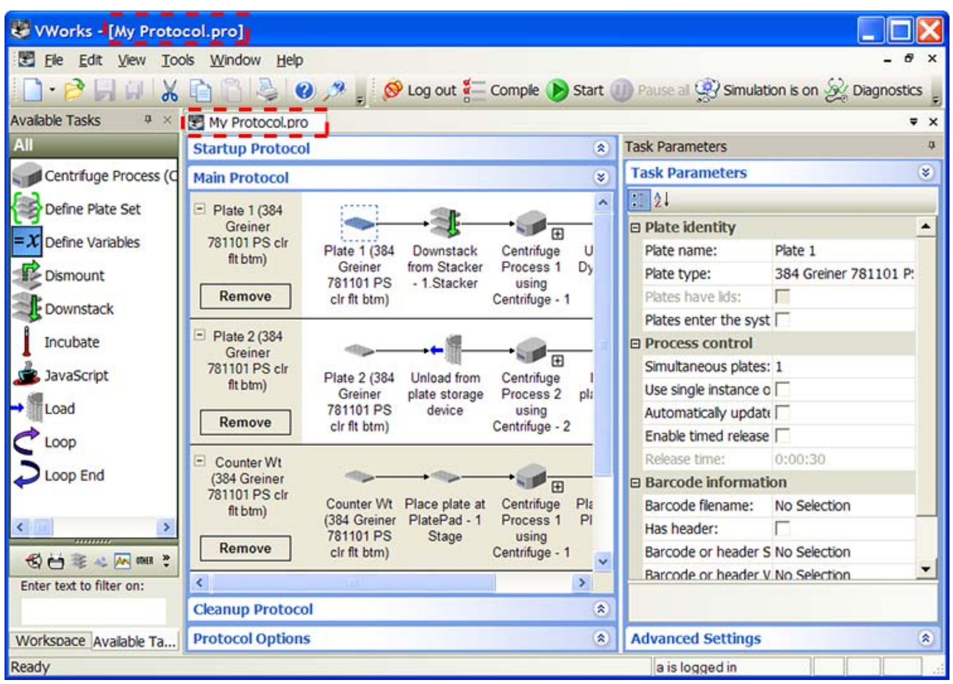
To open a protocol:

1 Log in to the VWorks software. For instructions, see “Logging in, logging out, and changing passwords” on page 22.

2 Select **File > Open.** The Open dialog box opens.

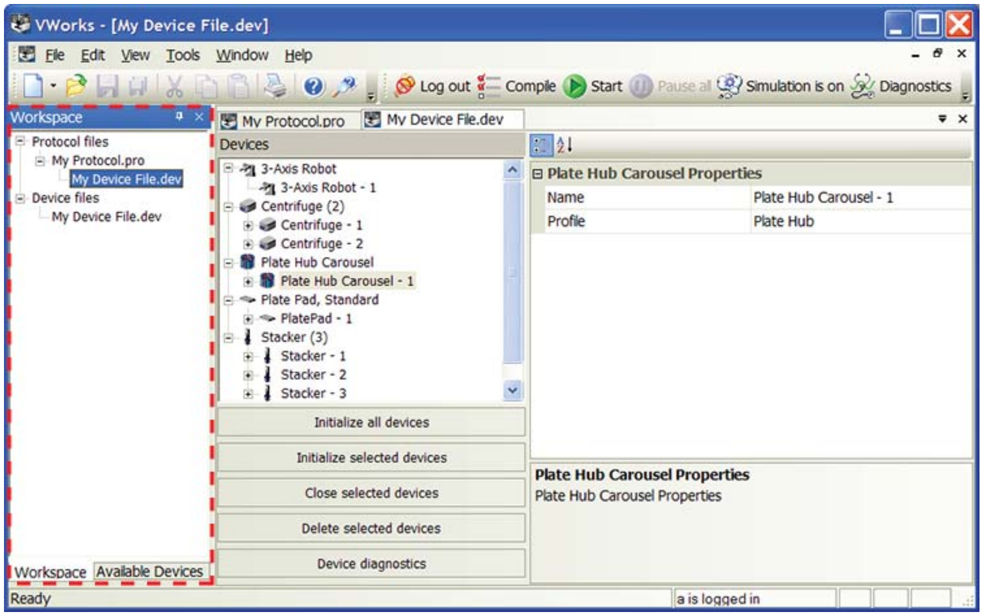
3 Locate and select the protocol (.pro) file, and then click **Open.**

The protocol information is displayed in the VWorks window. Notice that the name of the protocol appears in the title bar and on the tab in the protocol area.



You can open and display multiple protocols. Repeat the steps in this section to open other protocols. To view a particular protocol, in the Protocol area, click the tab that displays the protocol name.

Note: Every protocol requires a device file to run. When you open a protocol file, the device file associated with it is automatically loaded. However, the device file is not displayed. If you want to display the device file, click the **Workspace** tab in the **Available tasks** area, and then click the device file name.



Related information

For information about...	See...
VWorks software components	"Relationship of VWorks components" on page 4
Preparing devices for a run	Device user guide
Creating protocols	• "Creating a protocol: basic procedure" on page 13 • "Creating a protocol: advanced topics" on page 73
Setting the log file directories	"Setting log file directories" on page 222
Setting error-handling options	"Setting error-handling options" on page 229
Setting up email notification	"Setting up email notification" on page 232
Adding an alarm	"Adding an alarm" on page 37
Running a protocol	"Workflow for running a protocol" on page 210

Disabling and enabling a device in the device file

When to temporarily disable a device

You might want to temporarily disable a device when:

- Running a protocol using a subset of devices
- Running protocols that use different pipette heads or deck setups
- Performing maintenance on a pooled or non-pooled device

The disabled device will be not be available during a protocol run. If the software encounters a task pointing to the disabled device during a run, the protocol will skip the task and continue the run. If the disabled device is part of a pool of devices and the other devices in the pool are available, the task will be executed using the available device.

Note: Up to 10 devices in a single device file can be disabled.

To disable a task without disabling the device, see "Disabling and enabling tasks" on page 641 for more information.

Devices that cannot be disabled

The following devices cannot be disabled:

- All robots, for example the BenchBot Robot and Direct Drive Robot
- Automation Control Unit
- BioCel I/O Interface

When to enable a device

You might want to enable a device when:

- You are running a protocol that uses a device that was disabled for another protocol
- You have finished performing maintenance on a device

Before you start

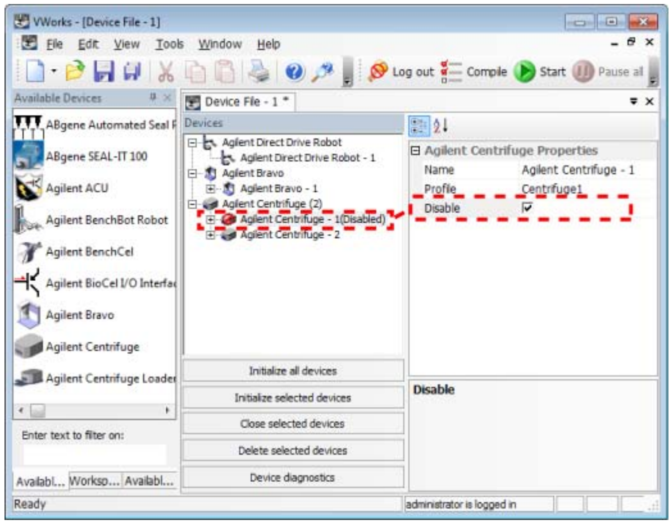
Before you enable a device, make sure that you have set the teachpoints for the device. For more information see "Adding devices" on page 25 and the user guide for the specific device.

Procedure

To disable a device in a device file:

1 In the VWorks window, choose **File > Open** and select the device file (.dev) that you want to edit. The device pane appears in the VWorks window and displays the list of devices.

2 Select the device in the **Devices** area and then select the **Disable** check box under the device properties area, or right-click the device and select **Disable device**. In the Device area, appears on the device icon, and (Disabled) is appended to the device name.



To enable a device in a device file:

1. In the VWorks window, choose **File > Open** and select the device file (.dev) that you want to edit. The device pane appears in the VWorks window and displays the list of devices.
2. Clear the **Disable** check box under the device properties area. Alternatively, right-click the device name and select **Enable device**.

In the Devices area, the ⃠ icon disappears from the device and the name is no longer appended with the (Disabled) status.

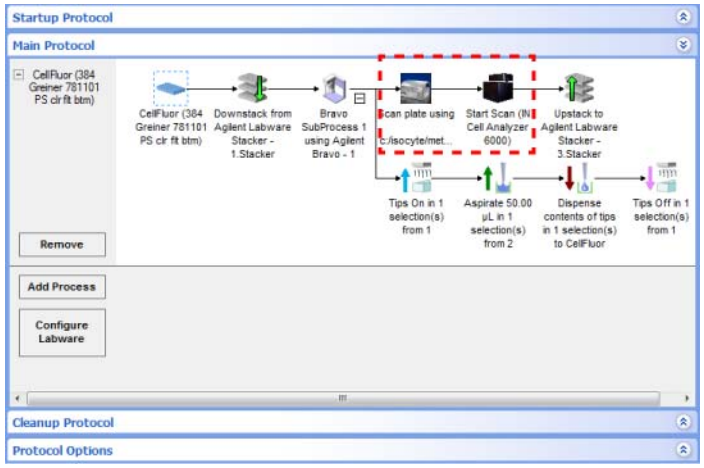
If you receive a VWorks Error message when enabling a device, stating the number of added devices has exceeded the number of devices allowed by your license, you will need to disable a device to continue. Alternatively, you can contact Automation Solutions Technical Support to inquire about increasing the number of devices for your license.

Example situations for disabling a device

Running a protocol using a subset of devices

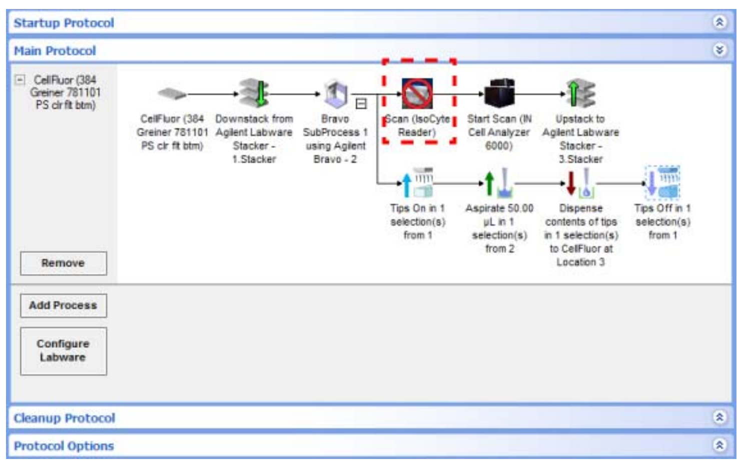
For example, your endpoint measurement for two assays use different readers that are not simultaneously accessible but the microplates are prepared using a similar procedure. In this case, you can disable the reader that you are not using and modify the task parameters for the common tasks, keeping the existing profiles and teachpoints.

1. In the VWorks window, choose **File > Open** and select the device file (.dev) that is associated with the protocol.
2. Ensure both readers are added and configured in the device file. See "Adding devices" on page 25 for instructions.
3. Open the protocol you want to run and ensure the Scan tasks for both readers are added to the protocol.



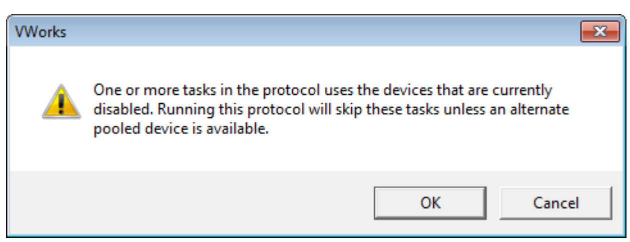
4 In the device file, disable the reader that is not used in the protocol, in this example the IsoCyte Reader. See “To disable a device in a device file:” on page 215 for instructions.

In the protocol, the Scan (IsoCyte Reader) task associated with the disabled device, displays the icon.

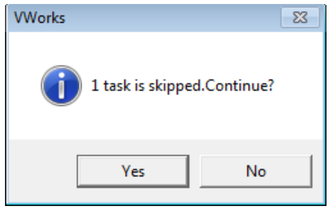


5 Set the task parameters for the Scan (Cell Analyzer) and ensure the task parameters for the other tasks are correct.

6 When you are ready to run the protocol, click **Start**. A disabled device message appears and asks you to confirm that you want to skip the disabled tasks.



7 Click **OK** to confirm that the tasks should be skipped. A second message appears and asks you to confirm the number of tasks to be skipped. In the following example, one task is skipped.



8 Click **Yes** to confirm the number. The Run Configuration Wizard opens. Follow the instructions in the wizard. For details on the wizard, see "Starting the protocol run" on page 237. During the run, the tasks associated with the disabled device are skipped. In the preceding example, the Scan (IsoCyte Reader) task is skipped because the IsoCyte Reader device is disabled.

Running protocols that use different pipette heads or deck setup

For example, you might want to run two protocols that use a Bravo Platform but each one requires a different pipette head. One protocol uses 96-well microplates and the other 384-well microplates. Instead of having two device files, each with a unique Bravo Platform profile for a specific head, you can create one device file in which you add two Bravo Platform devices, each with a unique profile and setup. When running one of the protocols, the Bravo Platform device with the configuration that is not in use is disabled.

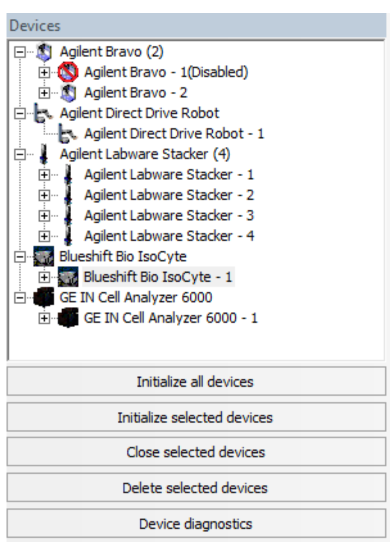
1 In the VWorks window, choose **File > Open** and select the device file (.dev) that is associated with one of the protocols that use the Bravo Platform.

2 Add a second Bravo Platform and configure it to use the second profile and setup. See "Adding devices" on page 25 for instructions.

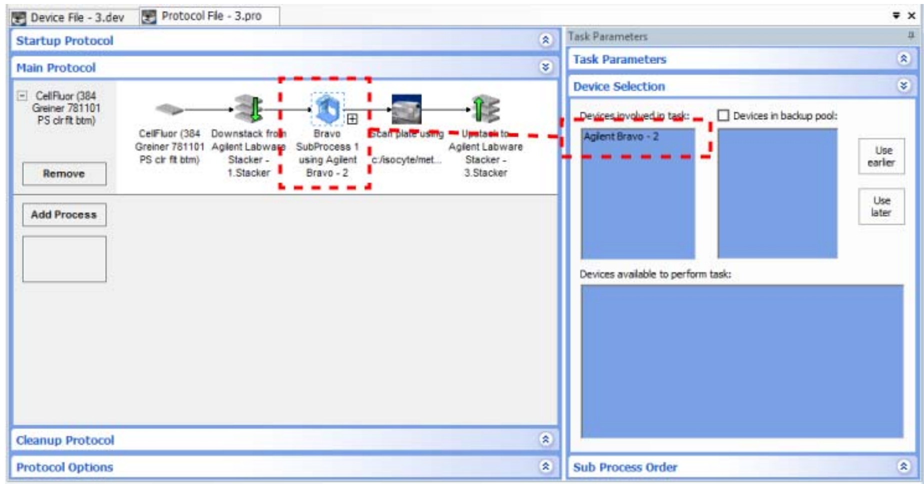
3 Open the protocol you want to run. In this example, it is the one that uses 384-well microplates (Bravo -2 profile).

4 Disable the Bravo Platform device with the profile (Bravo-1) that is not used in the protocol. See "To disable a device in a device file:" on page 215 for instructions.

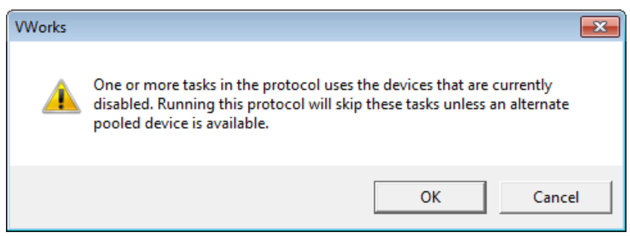
Note: The disabled Bravo device will be removed from the available device list.



5 In the protocol, ensure the Bravo subprocess is associated with the correct Bravo device. See “SubProcess (Bravo, Vertical Pipetting Station)” on page 388 for instructions.



6 When you are ready to run the protocol, click **Start**. A disabled device message appears and asks you to confirm that you want to skip the disabled tasks.



7 Click **OK** to run the protocol. The Run Configuration Wizard opens. Follow the instructions in the wizard. For details on the wizard, see “Starting the protocol run” on page 237.

Performing maintenance on a non-pooled device

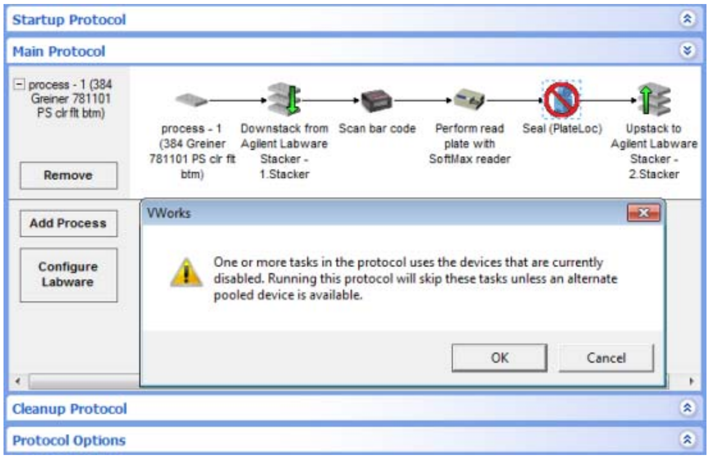
While replacing a roll of seal on a PlateLoc Sealer, you want to continue to perform tasks on the other devices.

1 Open the protocol you want to run.

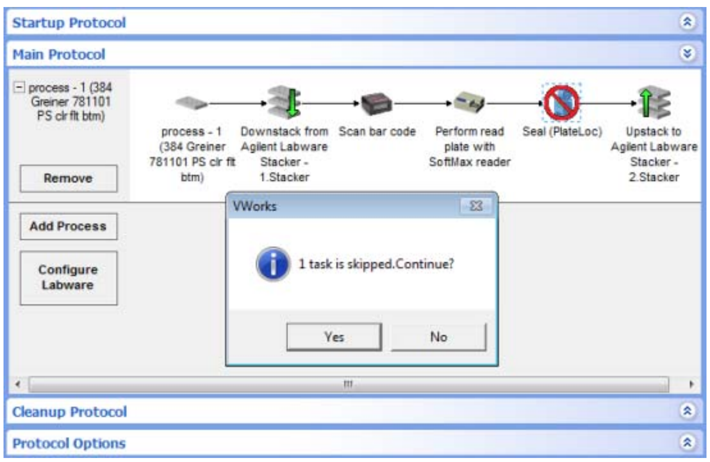
2 Open the device file associated with the protocol.

3 Disable the PlateLoc Sealer associated with the protocol. See “To disable a device in a device file:” on page 215 for instructions.

4 When you are ready to run the protocol, click **Start**. A disabled device message appears and asks you to confirm that you want to skip the disabled tasks.



5 Click **OK** to confirm that the tasks should be skipped. A second message appears and asks you to confirm the number of tasks to be skipped. In the following example, one task is skipped.



6 Click **Yes** to confirm the number. The Run Configuration Wizard opens. Follow the instructions in the wizard. For details on the wizard, see "Starting the protocol run" on page 237. During the run, the tasks associated with the disabled device are skipped. In the preceding example, the Seal task is skipped because the PlateLoc Sealer device is disabled.

Performing maintenance on a pooled device

While performing maintenance on a PlateLoc Sealer that is a member of a pool of PlateLoc Sealers, you want to continue to run a protocol using the other devices.

1 Open the protocol you want to run.

2 Open the device file associated with the protocol.

3 Disable the PlateLoc Sealer you want to service. See "To disable a device in a device file:" on page 215 for instructions.

Note: If the disabled PlateLoc Sealer device was associated with a Seal task, the log contains a warning indicating that because no device is assigned to the Seal task, the task will be skipped during the protocol run.

4. Reassign another PlateLoc Sealer from the device pool to the Seal task and verify that the task parameters are correct.

   Note: If another PlateLoc Sealer was placed in the backup pool, the software automatically moves the backup device to the Devices involved in this task area and you will not have to reassign another device.

5. Run the protocol. The software uses the alternate available device.

   Note: If you do not reassign another device to the affected task, a message appears when you compile or run the protocol and allows you to skip the task.

Related information

For information about...	See...
VWorks software components	"Relationship of VWorks components" on page 4
Adding a device	"Adding devices" on page 25
Preparing devices for a run	Device user guide
Creating protocols	• "Creating a protocol: basic procedure" on page 13 • "Creating a protocol: advanced topics" on page 73
Starting a protocol	"Starting the protocol run" on page 237
Add a SubProcess	"SubProcess (Bravo, Vertical Pipetting Station)" on page 388
Adding a task	"Adding and deleting tasks" on page 53
Simulating a run	"Simulating the protocol run" on page 64
Running a protocol	"Workflow for running a protocol" on page 210

Setting log file directories

About the log files

The VWorks software records events that occur and stores the information in the following logs:

- *Main Log*. Contains all of the actions that occur in the software.
- *Pipette Log*. Contains all pipetting transfer tasks.
- *Time Constraints Log*. Contains all information about time- limited tasks.

You cannot edit or delete log entries within the VWorks software, but you can specify where they are stored on the computer. This topic describes how to change the log file location.

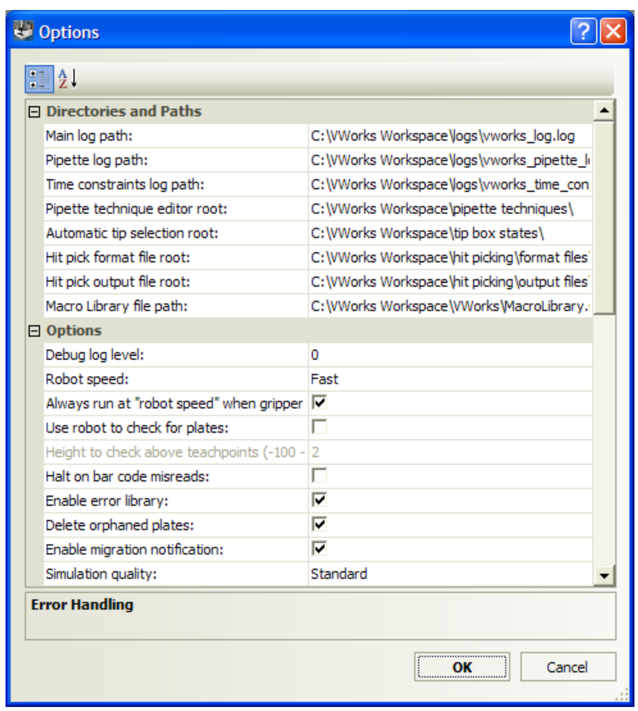
Procedure

You can set the log file directory during setup or the first time a protocol is run in the software. You do not need to set the log file directory every time you run a protocol.

> **CAUTION** The settings in the Options dialog box apply to all protocol runs. Always check the settings before you start a run.

To change the location of the log files:

1 Select **Tools > Options.** The Options dialog box opens. The Directory and Paths area lists the different log files and their directories.



2. To change the log file location, click the log path, and then click the ... button that appears. The Open dialog box opens.
3. Locate and select the desired location and click **Open.** The new path displays in the Options dialog box.
4. Click **OK** to save the changes and close the Options dialog box.

Related information

For information about...	See...
Viewing the log files	"Viewing logs" on page 631
Setting error-handling options	"Setting error-handling options" on page 229
Setting up email notification	"Setting up email notification" on page 232
Adding an alarm	"Adding an alarm" on page 37
Running a protocol	"Workflow for running a protocol" on page 210

Setting general and view options

Setting the general options

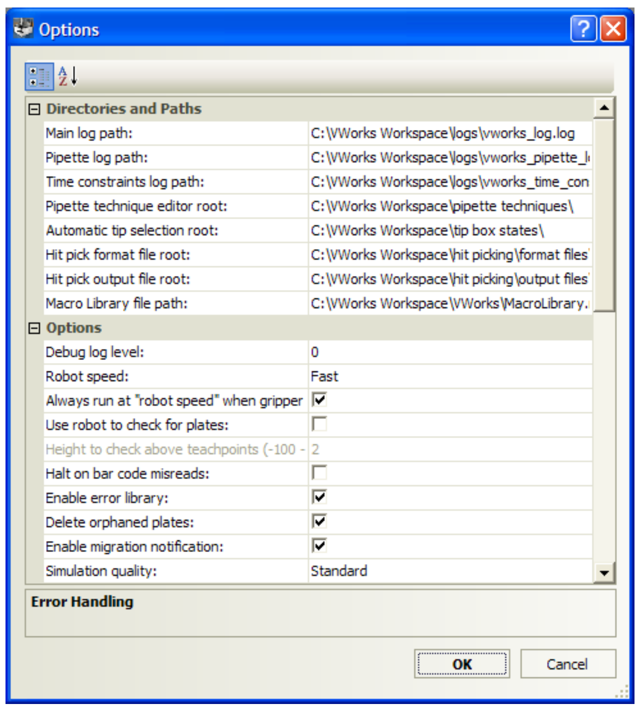
General options allow you to specify the amount of debug information to display in the Main Log, set the robot speed, enable the error library, and other preferences.

You can set the general options during setup or the first time a protocol is run in the software. You do not need to set the general options every time you run a protocol.

CAUTION The settings in the Options dialog box apply to all protocol runs. Always check the settings before you start a run.

To set the general options:

1 Select **Tools > Options.** The Options dialog box opens.



2 In the **Options** area, set or select the following options:

Option	Description
Debug log level	The amount of debug information presented in the Main Log. This field should be used by Automation Solutions Technical Support or advanced users for troubleshooting purposes. Valid values are: • *0*. No debug information. This default value is for typical use. • *1–5*. Various levels of plugin information. • *6–100*. Additional debug information. The larger the value, the more information is presented.
Robot speed	The maximum general speed at which the robot will move during the run. If you are testing a new protocol or learning to use the system, run the robot at a slow or medium speed to reduce the risk of crashes. In addition to this general robot speed, you can set: • *Robot-handling speed for labware*. Specifies the maximum speed at which the robot can move when handling a specific type of microplate. This parameter is set in the Labware Editor on the Plate Properties tab. If this speed differs from the general robot speed, the robot uses the slower of the two speeds. For more details on the Labware Editor, see the *VWorks Automation Control Setup Guide*. • *Always run at robot speed when gripper is empty*. This option allows the robot to move at a faster speed when no labware is in the gripper and yet retain the ability to move more slowly when carrying labware.
Always run at "robot speed" when gripper is empty	The option to move at the Robot speed setting, above, when no labware is in the robot gripper. For example, if the Robot speed is fast, but the speed for the labware is slow: • *Select the check box (default)*. Allows the robot to move fast when moving to the labware pickup location, while the gripper is still empty. • *Clear the check box*. Results in the robot moving at slow speed both while moving to the pickup location and while picking up and placing the labware. For the maximum throughput, ensure this option is selected.
Use robot to check for plates	The option that requires the robot to move to all positions defined in the device file to check that the positions are empty. In devices that have more than one location such as the Microplate Rotator, Linear Translator, and the Microplate Centrifuge, the robot will check to make sure all locations are empty. *Note:* This option applies only to the BioCel System robots.

Option	Description
Height to check above teachpoints	This parameter is available if you select Use robot to check for plates. The height you specify is the offset added to the *z*-axis coordinate of the teachpoint being checked. Default: 2.0 mm
Halt on barcode misreads	The option that requires the software to pause the protocol when a barcode scanned does not match the barcode in the input file. Causes of the misread include missing barcode labels, damaged labels, or wrong labels.
Enable Error Library	The option to turn on access to the Error Library.
Delete Orphaned Plates	The option to clear from memory any labware that were left in storage devices temporarily before the run was aborted. Doing so allows the use of the storage location in the next protocol run. For example, during a run, a labware is placed in a Plate Hotel slot while it waits for a device. You abort the run. Before you can restart the run, you must physically remove orphaned labware left in the system from the previous run, such as the labware placed temporarily in the Plate Hotel. However, the software still remembers that the labware is in the Plate Hotel slot. Selecting this option clears the software memory and permits that location to be used in the new run.
Enable migration notification	The option that requires the software to display a message whenever you try to open a protocol written in the previous versions of the VWorks software.
Simulation quality	The accuracy of the simulation. Select one of the accuracy levels: • *Standard*. The lowest level of accuracy, because the software does not query the device plugins. Certain physical constraints are not simulated, so this option results in the fastest simulation. • *More accurate green dots*. The next level of accuracy where the simulation might take longer than Standard quality. Similar to the Standard quality option, the software does not query the device plugins for physical constraints. However, the software ensures the green dots in the Protocol area are positioned accurately during the simulation. • *Dots + communicate with plug-ins*. The highest level of accuracy. The software queries the device plugins to ensure every move is physically permissible, resulting in a slower but more accurate simulation.

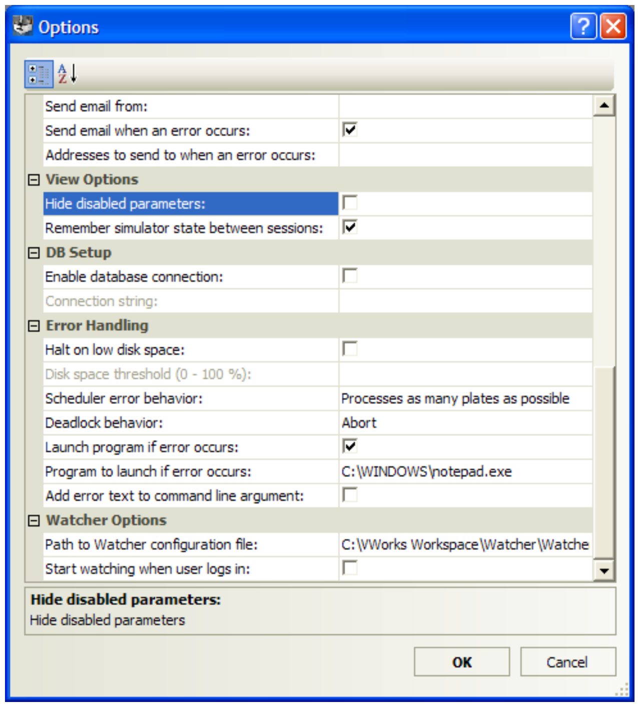
Setting the view options

View options allow you to hide parameters that are not in use and remember the simulation mode when you restart the software.

You set the view options when you first set up the software. You do not need to set the view options every time you run a protocol.

To set the view options:

1 In the **View Options** area, set the following options:



Option	Description
Hide disabled parameters	The option to hide parameters that are not in use: • To hide the unavailable parameters, select the check box. • To show the unavailable parameters as grayed out items, clear the check box. For example, in the Aspirate task, if you select Perform tip touch, additional parameters appear, and you can specify the sides of wells to use for tip touch. If you do not select Perform tip touch, the additional parameters can be hidden or grayed out, depending on whether you selected Hide disabled parameters.
Remember simulator state between sessions	The option to remember the simulation mode (Simulation is on, or Simulation is off) when you restart the software.

2 Click **OK** to save the changes and close the Options dialog box.

Related information

For information about...	See...
Setting the log file directories	"Setting log file directories" on page 222
Resolving barcode reader error messages	"Resolving barcode reader error messages" on page 647
Setting error-handling options	"Setting error-handling options" on page 229
Setting up email notification	"Setting up email notification" on page 232
Adding an alarm	"Adding an alarm" on page 37
Running a protocol	"Workflow for running a protocol" on page 210
Setting up the error library	"Setting up automated error responses" on page 654

Setting error-handling options

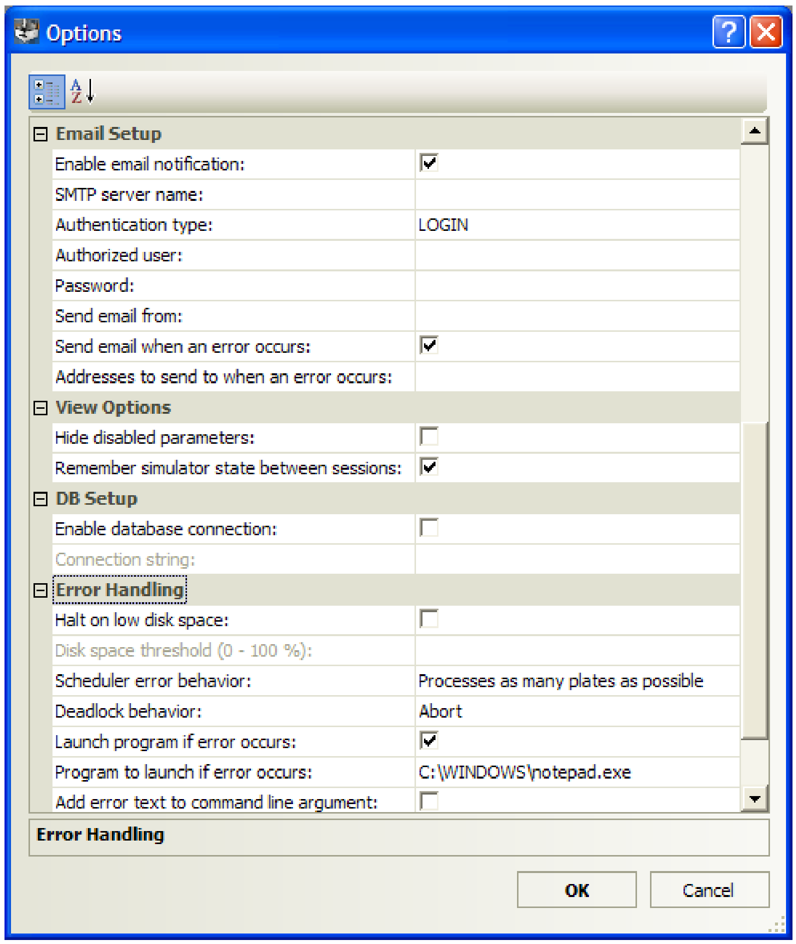
Procedure

You can set the error-handling options during setup or the first time a protocol is run in the software. You do not need to set the error-handling options every time you run a protocol.

CAUTION The settings in the Options dialog box apply to all protocol runs. Always check the settings before you start a run.

To set the error-handling options:

1 Select **Tools > Options.** The Options dialog box opens.



2 In the **Error Handling** area, set or select the options:

Option	Description
Halt on low disk space	The option that requires the software to stop scheduling tasks when the computer disk space is low. If you select this option, you must specify the **Disk space threshold.**

Option	Description
Disk space threshold	The percentage of disk space at which the software will halt scheduling tasks.
Scheduler error behavior	The action the software should take if an error occurs during the run. Select one of the following actions: • Process as many plates as possible • Continue processing without starting any new plates • Stop scheduler
Deadlock behavior	The action the software should take if a deadlock occurs during the run. Select one of the following actions: • Abort • Show the System State Editor (pauses the run and displays the System State Editor)
Launch program if error occurs	Starts a specified software application anytime a VWorks error occurs.
Program to launch if error occurs	The file path to the executable file of the application that you want to start when an error occurs. To set the file path, click the field, and then click the ... button that appears. In the Open dialog box, select the file path for the executable file (.exe) or batch file (.bat).
Add error text to command line argument	Passes the text of the error message to the specified application. For example, if this check box is selected and an error occurs containing the text, “Your PlateLoc is out of seal,” a command will be generated that says: <filepath>\ABC.exe Your PlateLoc is out of seal If this check box is cleared, the following command would be generated for the same error: <filepath>\ABC.exe

3 When you are finished, click **OK** to save the changes and close the Options dialog box.

Related information

For information about...	See...
Setting the log file directories	“Setting log file directories” on page 222
System State Editor	“Recovering from deadlocks” on page 648

For information about...	See...
Automated error responses	"Setting up automated error responses" on page 654
Posting messages about VWorks events, such as errors and deadlocks	"Setting up automatic online notification" on page 235

Setting up email notification

About email notification

You can specify that the VWorks software send you an email or text message when a run error occurs. Setting up email notification also enables bug reporting.

Alternatively, you can configure VWorks to post online messages on Twitter. For details, see “Setting up automatic online notification” on page 235.

Requirements

The following are required for the email notification to work:

- The VWorks computer must be connected to a network with internet access.
- The outgoing email server must be set up on the system's computer.

Work with your IT organization to meet these requirements.

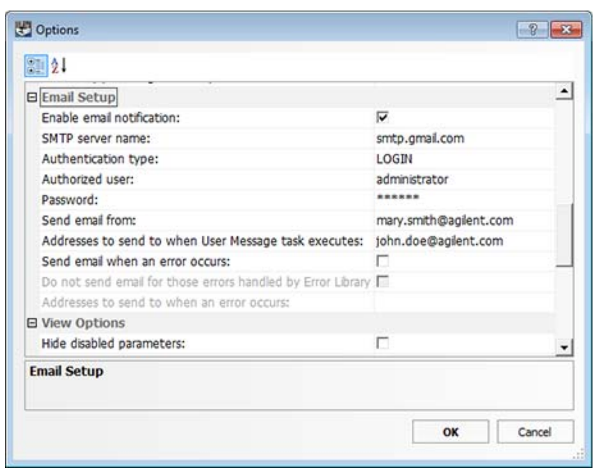
Procedure

You can set up email notification during setup or the first time a protocol is run in the software. You do not need to set up email notification every time you run a protocol.

CAUTION The settings in the Options dialog box apply to all protocol runs. Always check the settings before you start a run.

To set up email notification:

1 In the **VWorks** window, select **Tools > Options.** The Options dialog box opens.



2 Scroll down to the **Email Setup** area, and then specify the following:

Option or parameter	Description
Enable Email notification	The option that enables email communication from within the VWorks software. For example, you can report a bug from within the software by selecting Help > Report a Bug. The completed report is emailed to Agilent Technologies. Select the option to enable email communication. Clear the check box to disable email communication.
SMTP server name	The name of your outgoing email server. Contact your IT organization for details.
Authentication type	The information that might be required by the server. Contact your IT organization to obtain the information.
Authorized user	The user name used to access the email server. Contact your IT organization to obtain the information.
Password	The password that permits access to the email server. Contact your IT organization to obtain the information.
Send email from	The email address used in bug report emails to indicate the sender of the report. Agilent Technologies will reply to this email address when responding to the bug report.
Addresses to send to when User Message task executes	The addresses to which the outgoing message is sent when the User Message task runs. Click the browse button and type the email addresses in the Input Text dialog box that appears. Be sure to use semicolons between email addresses. To be notified on your mobile phone with a text message, type the email address for your mobile phone number. The typical format is: number@mobile_carrier_domain.com. Check with your mobile phone service provider. Any charges you might incur and the way messages are delivered and displayed depends on your wireless device and service plan.
Send email when an error occurs	The option to send an email or text message when an error occurs during the run.
Do not send email for those errors handled by Error Library	The option to not send emails if the errors are being handled by the Error Library.

Option or parameter	Description
Addresses to send to when an error occurs	The addresses to which the outgoing message is sent when an error occurs. Click the browse button and type the email addresses in the Input Text dialog box that appears. Be sure to use semicolons between email addresses. To be notified on your mobile phone with a text message, type the email address for your mobile phone number. The typical format is: number@mobile_carrier_domain.com. Check with your mobile phone service provider. Any charges you might incur and the way messages are delivered and displayed depends on your wireless device and service plan.

3 Click **OK** to return to the VWorks window.

Related information

For information about...	See...
Setting up online notification using Twitter	"Setting up automatic online notification" on page 235
Reporting problems	"Reporting problems" on page 661
Log file directory settings	"Setting log file directories" on page 222
General and view options	"Setting general and view options" on page 224
Error-handling options	"Setting error-handling options" on page 229
Starting a protocol run	"Starting the protocol run" on page 237

Setting up automatic online notification

About this topic

The VWorks software can automatically notify you about system status via email and online message postings. This topic describes how to configure online notification using Twitter. To configure VWorks email notification, see "Setting up email notification" on page 232.

About posting messages using an online service

The online posting feature can notify you about various system events, such as errors, deadlocks, and protocol and process starts and finishes. You can receive the online notification on any device where you can access a Twitter account. For example, suppose your lab has several workstations running protocols overnight. You could use a cell phone that has internet access to check on the run progress during a long commute home on the train.

The Twitter postings provide the flexibility to opt in or out at anytime. Whereas the email notification can be turned on or off only from within the VWorks software.

Before you start

The following are required for the online message posting to work:

- The VWorks computer must be connected to a network with internet access.
- The Twitter account must be set up ahead of time, and you must have the login information (user name and password) for the Twitter account.

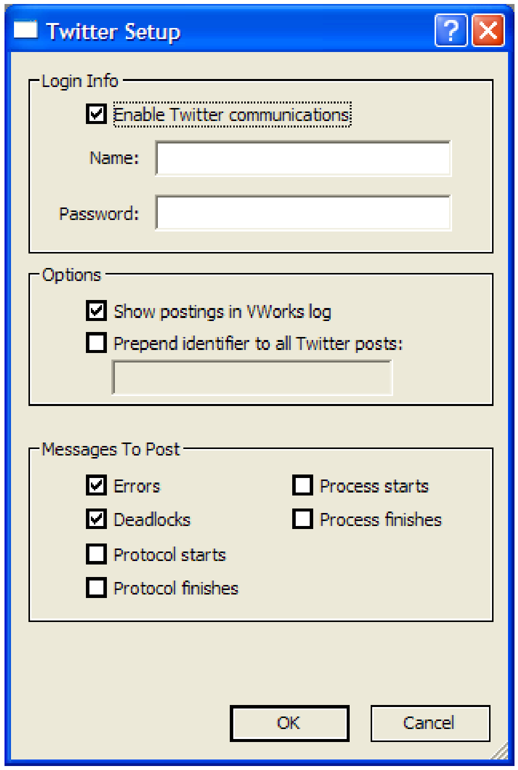
Procedure

You can configure the online notification during setup or the first time a protocol is run in the software. You do not need to configure online notification every time you run a protocol.

CAUTION The settings for online posting apply to all protocol runs. Always check the settings before you start a run.

To configure automatic message posting:

1 In the **VWorks** window, choose **Tools > Open Hooks Plugin for > Twitter.dll**. The Twitter Setup dialog box opens.



2 Under **Login Info**, select **Enable Twitter communications**, and type the **Name** and **Password** for the Twitter account that you want to use.

3 Under **Options**, select either of the following options, or clear the check boxes if you choose not to use these options:

- **Show postings in VWorks log**. If you want the postings to appear in the VWorks log, select this check box.
- **Prepend identifier to all Twitter posts**. To add a prefix to the start of each posting, select this check box, and then type the prefix that you want to include.

  For example, if you have multiple systems posting messages to the same Twitter account, you can use the system name as a prefix.

4 Under **Messages To Post**, select the types of messages that the VWorks software should post to the Twitter account, for example, errors and deadlocks (defaults).

5 Click **OK** to return to the VWorks window.

You may want to perform a dry run of a short protocol to test the posting. For example, you can select Protocol starts as a message to post, run a protocol, and then go to the Twitter account to view the message.

IMPORTANT If the number of online postings exceeds the Twitter account's maximum for a given time period, no further VWorks messages can be posted until the limit is reset. In this case, the VWorks software adds an info message to the main log for each attempted posting.

Related information

For information about...	See...
Adding alarms for error messages	"Adding an alarm" on page 37
Setting up email notification	"Setting up email notification" on page 232
Setting up automated error responses	"Setting up automated error responses" on page 654
Viewing logs	"Viewing logs" on page 631
Tracking the run progress of instances, processes, or devices	"Tracking the run progress of instances or devices" on page 255
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661

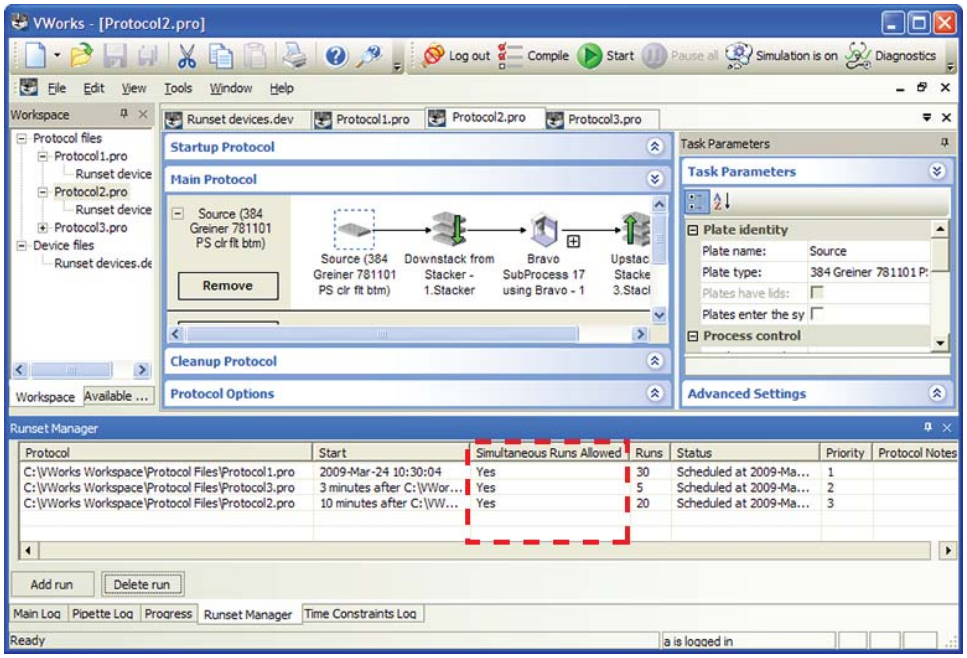
Starting the protocol run

Before you start

CAUTION The instructions in this topic assume that the protocol is free of errors.

Before you start a run, make sure you review the protocol. To prepare for the run, you should:

- Find out what devices are used in the protocol and prepare them for operation. For example, you might need to load a roll of seal on the PlateLoc Sealer or install a pipette head on a Vertical Pipetting Station. See the device user guides for prerun-check and homing instructions.
- Find out what labware are used in the protocol and where they should be positioned before the run starts. For example, you might have to load labware into a stacking device.
- Make sure waste bins are empty and reservoirs are filled.
- Check the protocol for User Message tasks.
- If you are starting a runset, verify whether simultaneous runs should be allowed. You can view the setting for this rule in the Runset Manager tab. To change this protocol rule, see "Specifying protocol rules" on page 34.



About scheduling runs

When you start a run, you have the option of scheduling the protocol to run immediately or at a future date and time. In addition, you can specify that the protocol start while another is still running. Running multiple protocols simultaneously can maximize device use and throughput.

IMPORTANT You can schedule multiple protocols to run simultaneously if they all reference the same device file.

When you schedule multiple runs:

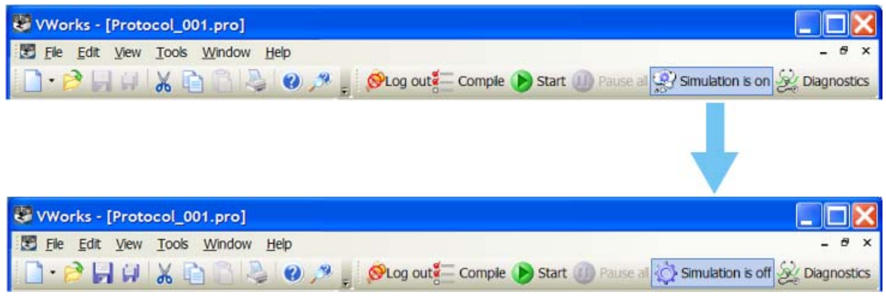
- You can start a run while existing protocols are already running or are scheduled to run.
- Protocols that are running simultaneously can share devices. The priority of device use is specified in the device selection area when you set the task parameters.
- You can manage a set of runs using the Runset Manager. For details, see "Managing runsets" on page 243.

Procedure

You can start a run if you have administrator, technician, or operator privileges.

To start a run:

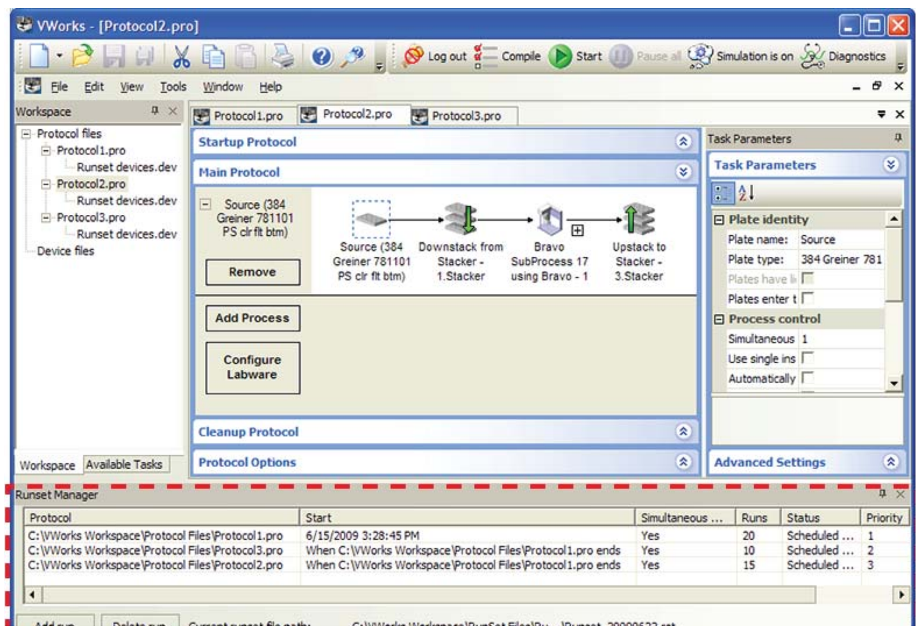
1 In the VWorks window, turn off the simulation mode: On the toolbar, click **Simulation is on**. The button changes to **Simulation is off.**



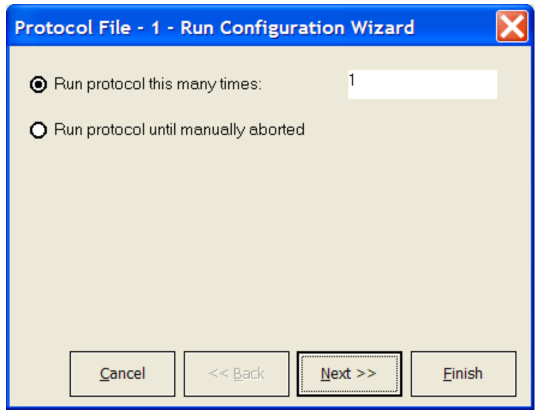
2 Do one of the following:

- Click **Start** on the toolbar.

- If you have a runset open, view the scheduled start times in the **Runset Manager** tab. To change a protocol start time, double-click the protocol name. For details about runsets, see "Managing runsets" on page 243.



The Run Configuration Wizard dialog box opens.



The Run Configuration Wizard allows you to:

- Specify the number of times to run the protocol.
- Schedule the run to start. You can start the run:
  - As soon as possible
  - On a specific date and at a specific time
  - At a fixed time after a selected protocol starts so that the two protocols are running simultaneously
  - At a fixed time after a selected protocol run ends
- Set the priority of runs (if multiple protocols are scheduled).
- Specify the starting barcode (if any).
- Type notes about the protocol.

CAUTION If you select the **As soon as possible** option, the protocol can start to run immediately after you complete the configuration wizard. Before you click **Finish**, verify that the system is set up and the protocol is ready to run.

IMPORTANT You must have technician or administrator privileges to set priorities. If you have operator privileges, the runs you start will always have the lowest priorities and the Higher priority and Lower priority buttons will be disabled.

3 Follow the instructions in the wizard and click **Next** or **Finish.**

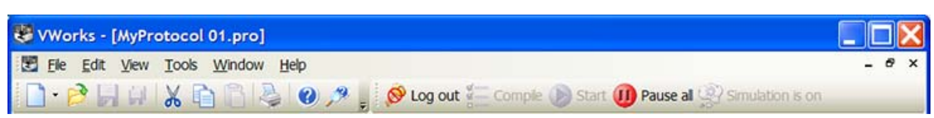
When you click **Finish,** the run starts if it is scheduled to run as soon as possible.

If this is the first run with the device file, and it is the only protocol running, the software confirms communication with all devices, and then the devices home. (If other protocols are already in progress, the software does not need to establish communication with the devices.)

When the run starts

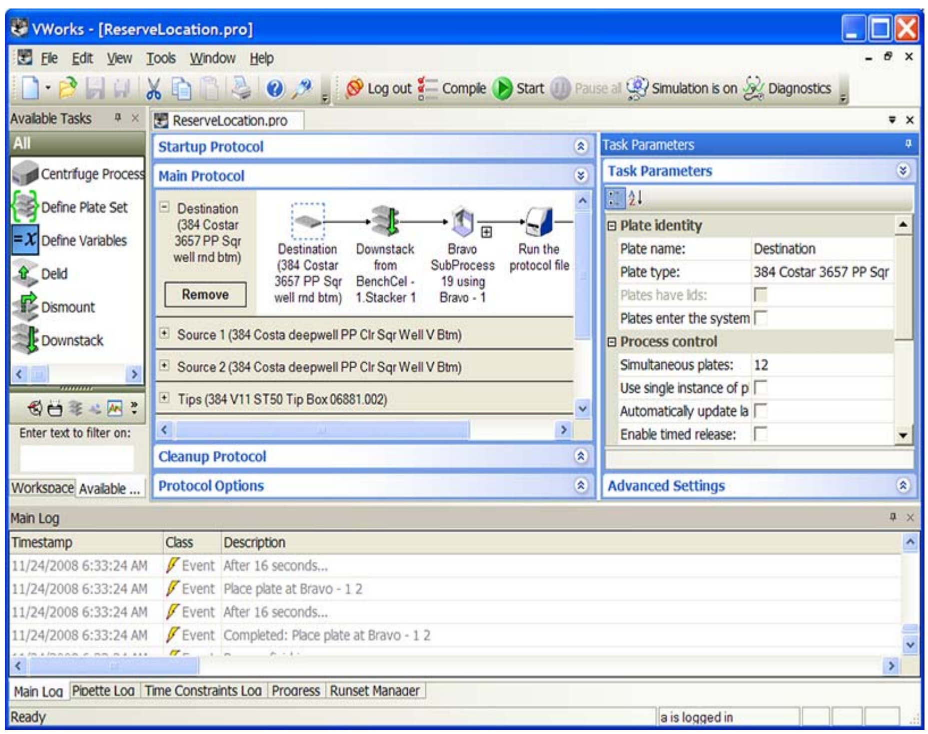
After you start the run, the following occur:

- The **Pause all** button becomes available.



- If the protocol contains tasks pointing to a disabled device, the software displays a warning message and a ⃠ is displayed on the affected tasks. If you click OK, you will get a second message asking you to confirm the number of tasks to be skipped.
- If you selected the **Use robot to check for plates** option (Tools > Options dialog box), the robot moves to all teachpoints defined in the device file (except those associated with storage devices such as the Plate Hub Carousel, StoreX incubator, and Cytomat incubator) to make sure they are unoccupied. If the device contains more than one location, such as a Shuttle or a Microplate Centrifuge, the robot will check all possible locations in these devices. An error message displays if a position is occupied. Remove any labware from the positions. After you fix the error, click **Retry.**
- The software checks that the protocol you are starting is using the same device file as the other protocols that are already running. If it uses a different device file, an error message appears.
- The protocol instructions are performed.
- If User Message tasks are included in the protocol, the software prompts you to respond to them as they appear.

  If there are no User Message tasks to remind you to empty liquid waste containers and refill liquid source containers, set your own reminders using lab timers.
- If it is not already listed, the protocol appears in the Runset Manager tab.
- Log messages appear in the Main Log, Pipette Log, and Time Constraints Log tabs. The log messages are recorded in the log file that is stored in the location you specified in the Tools > Options dialog box.



Related information

For information about...	See...
Setting global options	• "Setting log file directories" on page 222 • "Setting general and view options" on page 224 • "Setting error-handling options" on page 229 • "Setting up email notification" on page 232
Disabling a device in a device file	"Disabling and enabling a device in the device file" on page 215
Allowing or disallowing simultaneous runs	"Specifying protocol rules" on page 34
Managing runsets	"Managing runsets" on page 243
Use robot to check for plates option	"Setting general and view options" on page 224
Setting teachpoints on a device	Device user guide
User Message tasks	• "User Message" on page 599 • "Using simple variables" on page 77
Monitoring the protocol run	"Monitoring the overall run progress" on page 254
Pausing a protocol run	"Pausing the run" on page 261

Managing runsets

About runsets and the Runset Manager

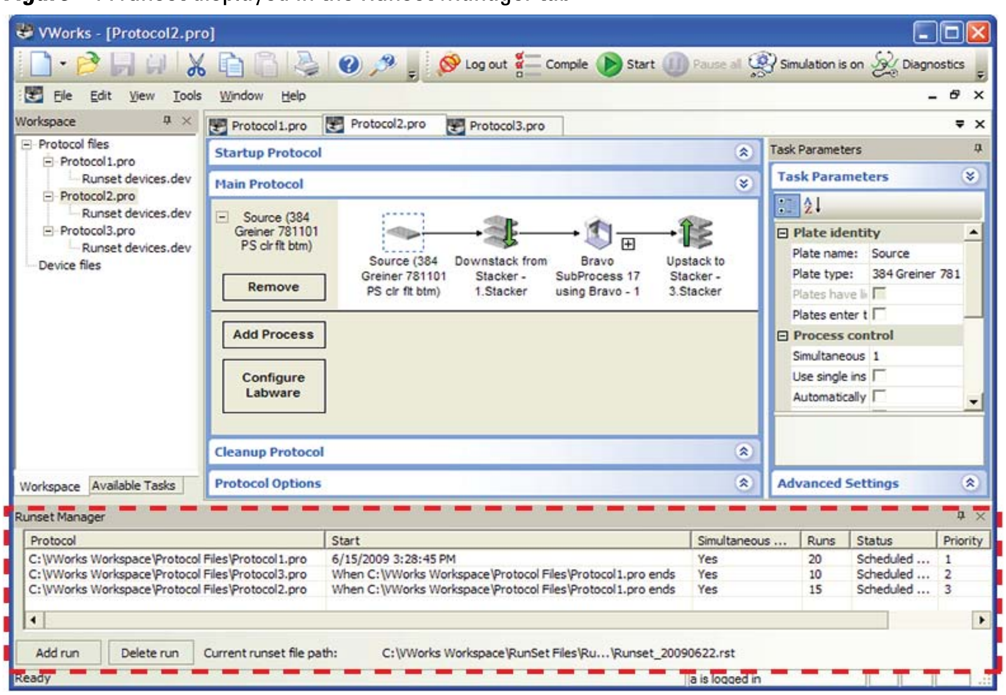
A runset is a collection of protocol runs that can be scheduled in advance to be performed without operator intervention. The following table provides an overview of how the Runset Manager works.

Step	Task
1	You add the protocols to a runset and specify the schedule for each protocol, or you can open an existing runset.
2	The Runset Manager determines the most efficient way to run the protocols based on the following: • The specified run start times. • Any specified run priorities. • Whether the protocols share the same device file. • Whether the protocol rules permit simultaneous runs.
3	The Runset Manager opens and compiles each protocol before running it the specified number of times.
4	The Runset Manager repeats step 3 for each protocol in the runset.

The following figure shows the Runset Manager tab, which lists each protocol in the runset on a separate row.

Note: To show or hide the Runset Manager tab, choose **View > Runset Manager**.

Figure A runset displayed in the Runset Manager tab



The Runset Manager tab contains the following columns, which list the runset parameters for each protocol.

Column name	Description
Protocol	The location and name of the protocol file.
Start	The scheduled start time of the protocol.
Simultaneous Runs Allowed	The protocol rule, set in the Protocol Options area, that specifies either: • *Yes (default)*. The Runset Manager can run the protocol while another protocol is running. • *No*. The protocol cannot run simultaneously with another running protocol.
Runs	The number of times the protocol is scheduled to run.
Status	The status of the protocol: • *Scheduled*. The protocol has a scheduled start time. • *Pending*. The protocol is being prepared to run. • *Running*. The protocol is currently running. • *Completed*. The protocol run has finished. • *Expired*. The protocol was scheduled to start at a time that has already past. • *Paused*. The protocol run is paused. • *Aborting*. The protocol is in the process of being aborted. • *Aborted*. The protocol was aborted. • *No new plates*. The protocol run is not delivering new microplates in the system.
Priority	The schedule priority of the protocol relative to the others in the list, where 1 is the first priority.
Protocol Notes	Any notes about the protocol that were entered in the Run Configuration Wizard.

Understanding how the run sequence is determined

The Runset Manager determines the run sequence of the protocols based on:

- *Scheduled start times*
  - As soon as possible (default)
  - On a specific date and at a specific time
  - Dependencies on other protocols:
    - At a fixed time after a selected protocol starts
    - At a fixed time after a selected protocol run ends

    The Runset Manager handles any protocols with such dependencies as a unit.

- *Whether the protocols have run priorities.* If the protocols in a runset have conflicting run times, the Runset Manager uses the priority settings to resolve which protocol to schedule before the others.
- *Whether simultaneous runs are allowed.* The Runset Manager may reschedule protocol start times if the protocol rules allow simultaneous runs. By default, protocols allow simultaneous runs. You can change this rule for each protocol in the Protocol Options area. See "Specifying protocol rules" on page 34.

The following scenarios provide a few examples.

Runset scenario 1. Start times with dependencies on other protocols

The Runset Manager processes the protocols in the set that have start time dependencies on one another as a unit. For example, assume that a runset includes the following four protocols, each of which can be completed in 10 minutes. The Runset Manager would change the run sequence as follows:

Run sequence: A, C, D, B

Protocol	Device file	Start time
A	1	1:00 pm
B	2	1:15 pm
C	1	30 minutes after protocol A ends
D	1	60 min. after protocol A ends

Even though no other protocols are running at 1:15 pm, the Runset Manager will reschedule the protocol B start time because of the start time dependencies of protocols C and D on protocol A.

Runset scenario 2. Start times with specific dates and times

If you change the start time of the protocols from scenario 1 to the specific start times in the following table, the run sequence would be as follows:

Run sequence: A, B, C, D

Protocol	Device file	Start time
A	1	1:00 pm
B	2	1:15 pm
C	1	1:30 pm
D	1	1:45 pm

Each protocol's start time is independent of the others in the set, so the Runset Manager processes each protocol individually. Although protocol B uses a different device file than the others, no other protocol is running at 1:15 pm, so the sequence is unchanged.

Scenario 3. As-soon-as-possible start times

If you change the start times of the protocols from scenario 1 to the as-soon-as-possible (ASAP) option, the run sequence would be as follows:

Run sequence: A, C, D, B

Protocol	Device file	Start time
A	1	As soon as possible
B	2	As soon as possible
C	1	As soon as possible
D	1	As soon as possible

Because protocol B does not use the same device file, the Runset Manager postpones the start until after completing the runs that share a common device file.

Additionally, if a protocol cannot be started for some reason, the Runset Manager can dynamically adjust the timing of the protocols that have as-soon-as-possible start times. For example, if protocol A cannot be started, the Runset Manager could change the protocol A start to depend on the protocol B end. If protocol A still fails to start after protocol B ends, the Runset Manager could change the protocol A start to depend on the protocol C end. If additional protocols were in the set, the start time could continue to be adjusted in this way until protocol A could be started successfully.

Simultaneous runs

If simultaneous runs were allowed in the previous runset scenarios:

- *Scenarios 1 and 2*. The run sequences would not change because of the specified start times and dependencies.
- *Scenario 3*. The run sequence could change to ACD,B. The protocols that share device file 1 would run simultaneously. After the ACD runs ended, the Runset Manager would initialize the device file 2 devices and start protocol B.

Creating a runset

To create a runset, you must have administrator or technician privileges.

To create a runset:

1. Choose **File > New > Runset**.

   Note: If the Runset Manager tab is not visible, choose **View > Runset Manager**.

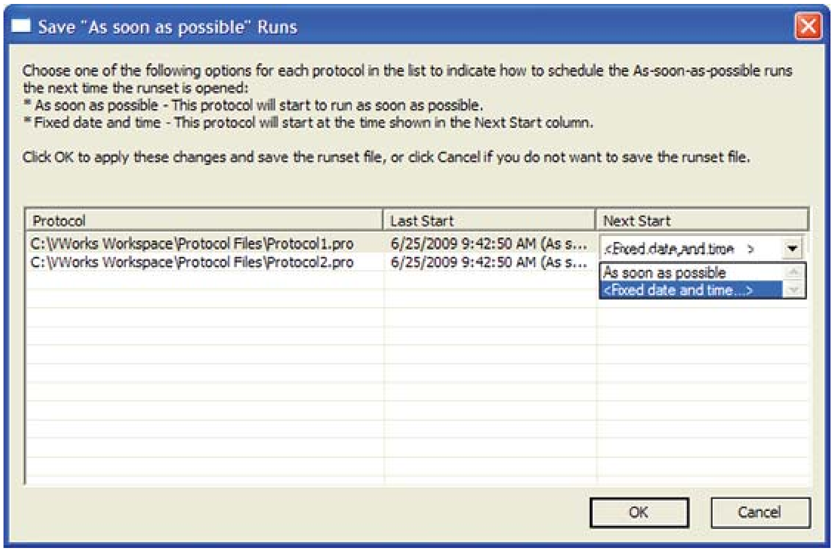
2. Add protocols to the runset using one of the methods in "Adding protocols to a runset" on page 248.

   IMPORTANT Simultaneously running protocols must specify the same static or configured labware.

3. Select **File > Save Runset.**
4. In the **Save As** dialog box, specify the file location and file name for the .rst file, and then click **Save**.

   If any of the protocols in the runset have the **As soon as possible** designation, proceed to step 5. Otherwise, the software saves the runset.

5 In the **Save “As soon as possible” Runs** dialog box, verify that all the protocols with the **As soon as possible** setting can be run as soon as possible upon reopening the runset.



To change the **Next Start** setting, click the field in the column, and select **Fixed date and time**. Upon reopening the runset, the operator will be prompted to provide a new start time.

Click **OK** to save the runset.

Alternatively, you can simply add protocols to the runset using one of the methods in “Adding protocols to a runset” on page 248, and then choose **File > Save Runset As**.

If you want to export a runset for use on another computer, see “Exporting and importing protocols and associated components” on page 626.

Adding protocols to a runset

CAUTION To avoid potential hardware crashes, verify whether each protocol that you add can run simultaneously with the other protocols in the runset. To change the protocol rules, see "Specifying protocol rules" on page 34.

IMPORTANT Simultaneously running protocols must specify the same static or configured labware.

CAUTION If you select the **As soon as possible** option, the protocol can start to run immediately after you complete the configuration wizard. Before you click **Finish**, verify that the system is set up and the protocol is ready to run.

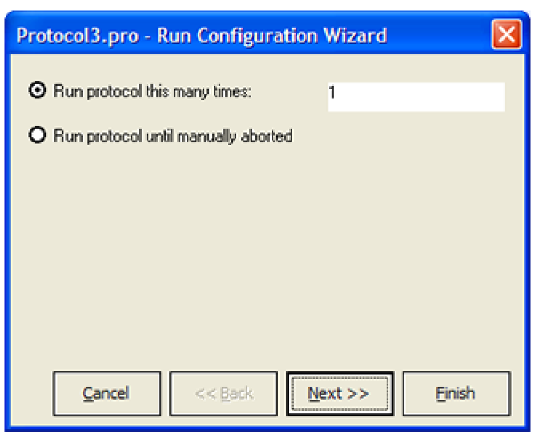
You can add a protocol to a runset in the following ways.

- Use the Add run button in the Runset Manager tab.
- Drag the protocol file name from the Workspace tab to the Runset Manager tab.
- Start a protocol run. See "Starting the protocol run" on page 237.

To add protocols using the Add run button:

1 In the **Runset Manager** tab, click **Add run**.

2 When the **Open a protocol file** dialog box appears, select the protocol (.pro), and then click **Open**. The Run Configuration Wizard appears.



3 In the Run Configuration Wizard:

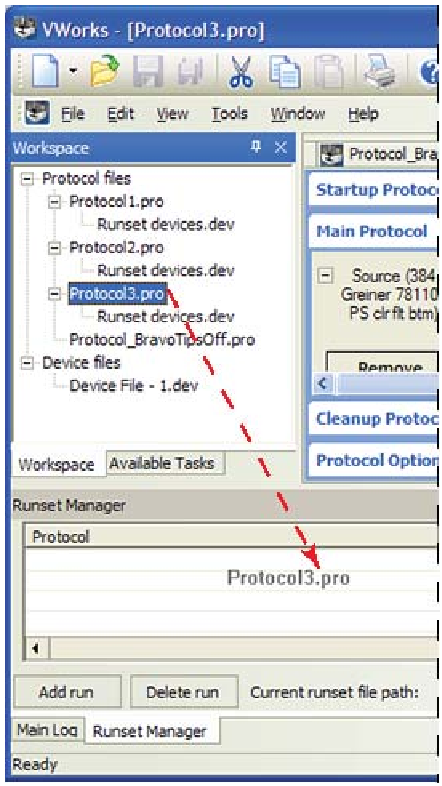
- Specify the number of times to run the protocol.
- Schedule the run start times.
- Set the priority of runs (if multiple protocols are scheduled).

- Specify the starting barcode (if any).
- *Optional*. Type notes about the protocol.

4 Repeat steps 1 to 3 for each protocol you want to add.

To add protocols using the drag-and-drop feature:

1 Drag the file name of the protocol from the **Workspace** tab to the **Runset Manager** tab.



Note: To add more protocols to the Workspace tab, see “Opening a protocol” on page 213.

2 In the Run Configuration Wizard:

- Specify the number of times to run the protocol.
- Schedule the run start time.
- Set the priority of runs (if multiple protocols are scheduled).
- Specify the starting barcode (if any).
- *Optional*. Type notes about the protocol.

3 Repeat steps 1 and 2 for each protocol you want to add.

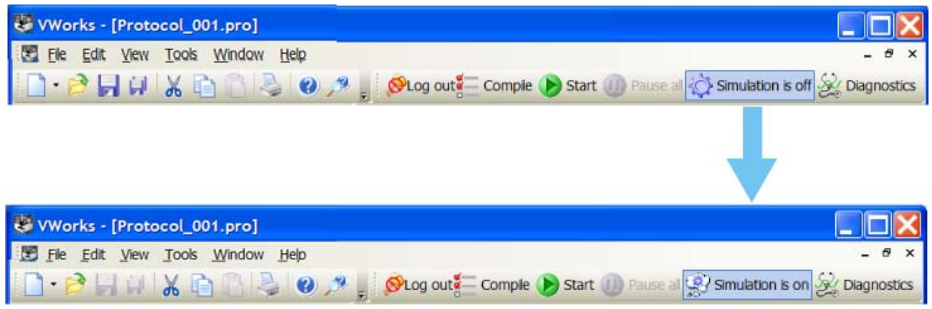
Opening runsets

IMPORTANT Verify whether the protocols in the runset can run simultaneously with other protocols (default). To change the protocol rules, “Allow this protocol to execute while other protocols are running” on page 35.

If you want to import a runset, see “Exporting and importing protocols and associated components” on page 626.

To open a runset:

1 *Optional*. To simulate the protocols in the runset before starting the actual runs, turn on the simulation mode. To turn on simulation, click **Simulation is off** on the toolbar. The button changes to **Simulation is on.**



2 Select **File > Open.** The Open dialog box appears.

3 Select the .rst file, and then click **Open.**

IMPORTANT If a protocol in the runset is scheduled to run as soon as possible, a message warns you that the run will start immediately when you open the runset. Make sure that the system is ready for the run to start before you continue.

Filtering the list of protocols displayed

You can apply a filter to display a subset of the protocols in the runset.

To filter the list of runs displayed in the Runset Manager tab:

1 Right-click anywhere in the **Runset Manager** tab.

2 In the shortcut menu that appears, select the desired filter:

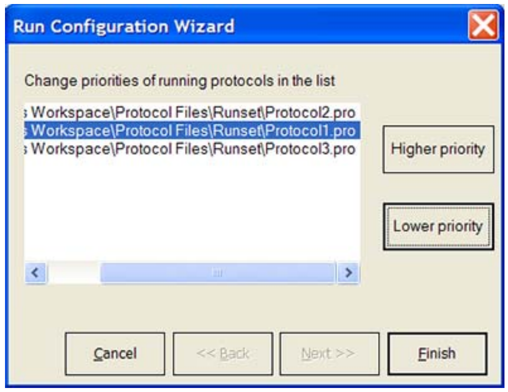
Filter command	Description
Show all	Displays all protocols in the runset.
Filter by row	Displays the protocols that meet the criteria you select. For example, you can display all protocols that are run 10 times.
Filter by column	Displays the protocols that meet the criteria you select.
Use last filter	Displays the protocols that meet the last filter criteria you selected.

Changing run priorities

To change the priority of the protocol runs:

1 In the **Runset Manager** tab, right-click anywhere in the protocol table.

2 In the shortcut menu that appears, select **Adjust run priority.** The Run Configuration Wizard dialog box appears.



3 Select the protocol whose priority you want to change, and then click **Higher Priority** to move it up the list, or click **Lower Priority** to move it down the list.

4 Click **Finish.** The Priority column shows the change in priority.

Aborting a protocol in a runset

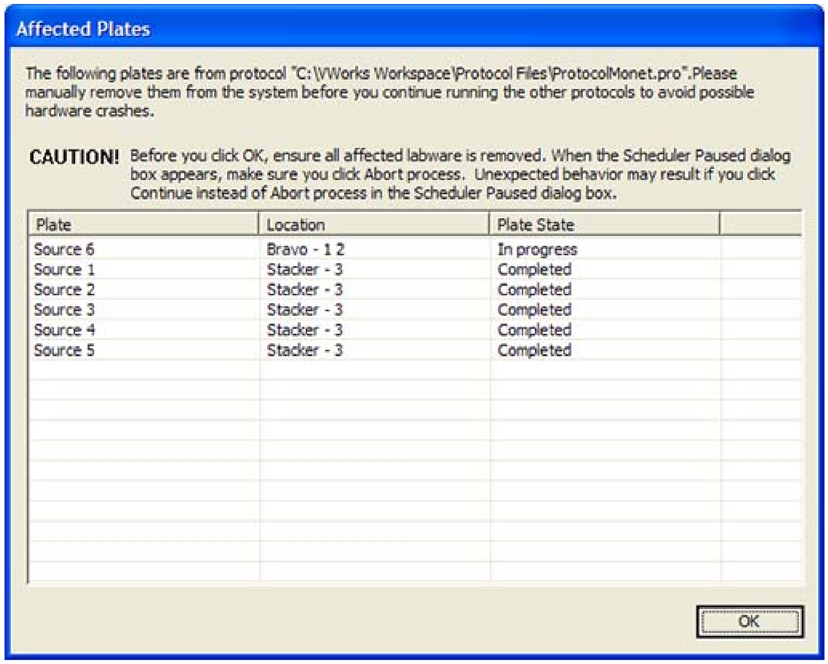
If you abort a protocol in a runset, the software also aborts any other protocols that have time dependencies associated with the aborted protocol.

To abort a protocol in a runset:

1 In the **Runset Manager** tab, right-click the protocol name, and then choose **Abort this run** from the shortcut menu.

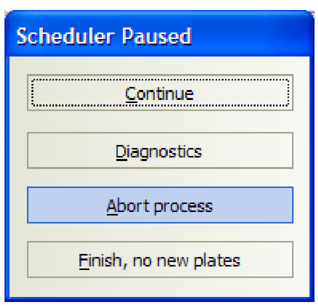
- *If the protocol is scheduled but has not yet started.* The software aborts the protocol and any other protocols that have dependencies on the aborted protocol. The Status column in the Runset Manager tab shows the time at which the protocols were aborted.
- *If the protocol has already started.* The software aborts the protocol. The Status column in the Runset Manager tab shows the time at which the protocol was aborted.
- *If you abort a running protocol and subsequent runs are scheduled.* The Affected Plates dialog box appears and displays the status of all the labware in the selected protocol.

CAUTION Before you click OK, proceed to step 2. Manually remove all the labware associated with the aborted protocol to ensure that you avoid a potential hardware crash.

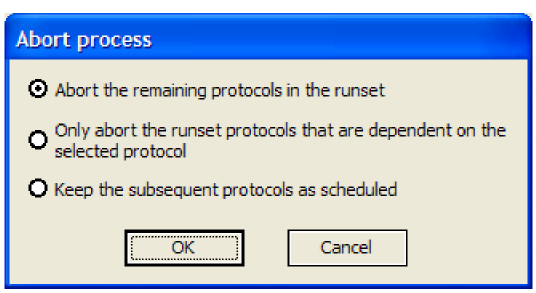


2 Manually remove the physical labware, which is associated with the aborted protocol, from the system.

3 In the **Affected Plates** dialog box, click **OK**. The Scheduler Paused dialog box appears, and the Status column in the Runset Manager tab indicates that the protocol is aborting.



4 In the **Scheduler Paused** dialog box, click **Abort process**. The Abort Process dialog box appears.



5 In the **Abort process** dialog box, choose one of the following options, and then click **OK**:

- Abort the remaining protocols in the runset

- Only abort the runset protocols that are dependent on the selected protocol
- Keep the subsequent protocols as scheduled

The Runset Manager updates the Status column.

Note: For details on the other options in the Scheduler Paused dialog box, see "Pausing the run" on page 261.

6 Verify that the remaining protocols in the runset have appropriate start times.

Rescheduling a protocol in a runset

To reschedule a protocol in a runset:

1 In the **Runset Manager** tab, right-click the protocol name, and choose **Adjust run start time and dependencies** from the shortcut menu.

2 In the **Run Configuration Wizard**, reset the protocol start time. The software automatically resets any other protocols that have time dependencies associated with the rescheduled protocol.

Deleting protocols in a runset

You can use the following procedure to delete a protocol that is not currently running. To abort a protocol that is in progress, see "Aborting a protocol in a runset" on page 251.

To delete a protocol from a runset:

1 In the **Runset Manager** tab, select the protocol.

2 Click **Delete run.** If no other protocols have dependencies on the selected protocol, the software removes the protocol from the list.

If other protocols in the runset have start time dependencies on the selected protocol, a message appears and warns you that continuing will abort the dependent protocols. After the protocol is deleted, adjust the start times of the remaining protocols.

Related information

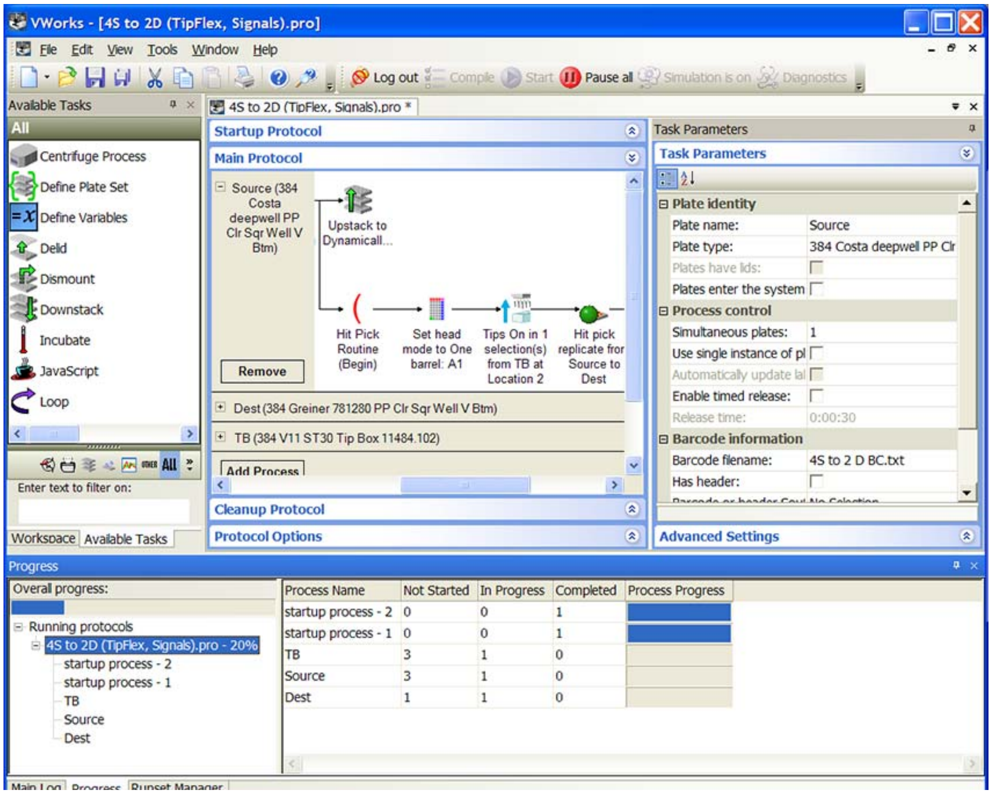
For information about...	See...
Allowing or disallowing simultaneous runs	"Specifying protocol rules" on page 34
Starting a protocol	"Starting the protocol run" on page 237
Monitoring runs	"Monitoring the overall run progress" on page 254
Pausing runs	"Pausing the run" on page 261

Monitoring the overall run progress

Procedure

To check the overall run progress:

In the VWorks window, select **View > Progress.** The Progress tab appears at the bottom of the window and displays the following:



Area or table	Description
Overall progress	Displays a progress bar to indicate how far along the system is in all running protocols.
Running protocols	Contains the list of protocols that are currently running. You can select a protocol name to display the progress of all processes in the protocol, or you can select a specific process and display its progress.
Display	Based on your selection in the Overall progress area, shows an aspect of the overall run progress.

Related information

For information about...	See...
Tracking run progress of instances, processes, or devices	"Tracking the run progress of instances or devices" on page 255
Pausing the run	"Pausing the run" on page 261
Stopping the run	"Stopping the run" on page 268

Tracking the run progress of instances or devices

About this topic

This topic explains the following:

Displaying the run progress in the Gantt Chart format

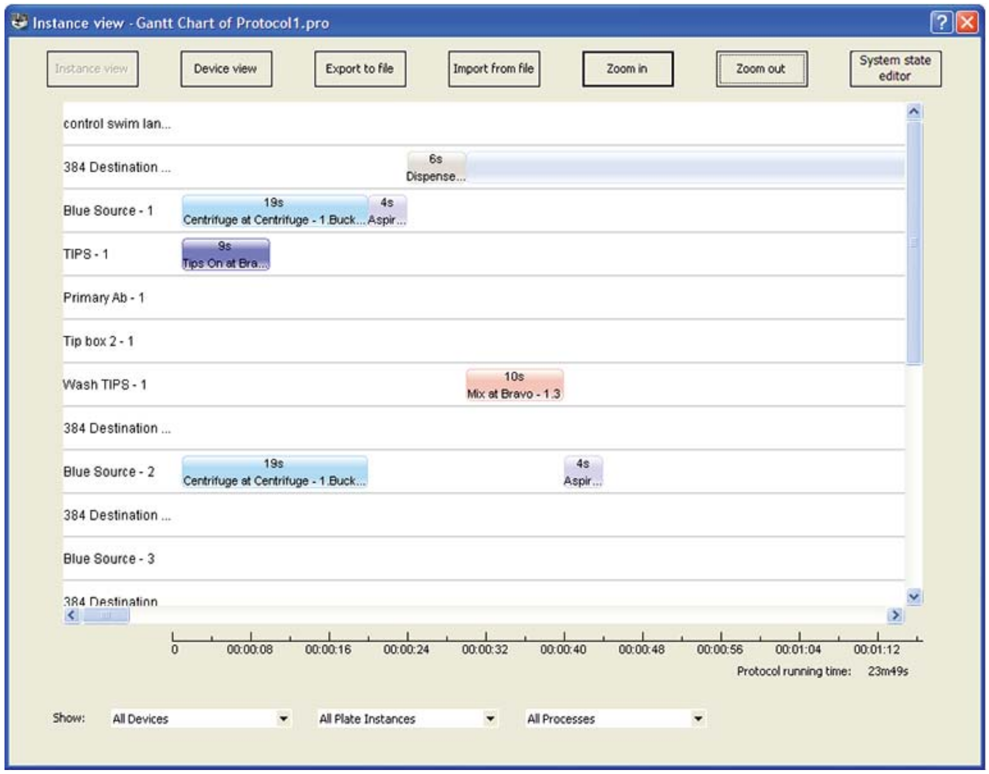
While the protocol is running, you can visually monitor the progress in real-time in the Gantt Chart dialog box. In addition, you have the option of displaying the progress by instance or device.

Run information displayed in the Gantt Chart format enables you to monitor resource usage during the run. Too much spacing (time) between tasks might indicate poor device usage or potential bottlenecks. After you examine the run information, you can:

- Determine the causes of bottlenecks and remove them.
- Add devices or try to run multiple protocols simultaneously to improve performance and throughput.

To visually track the progress of labware instances or devices:

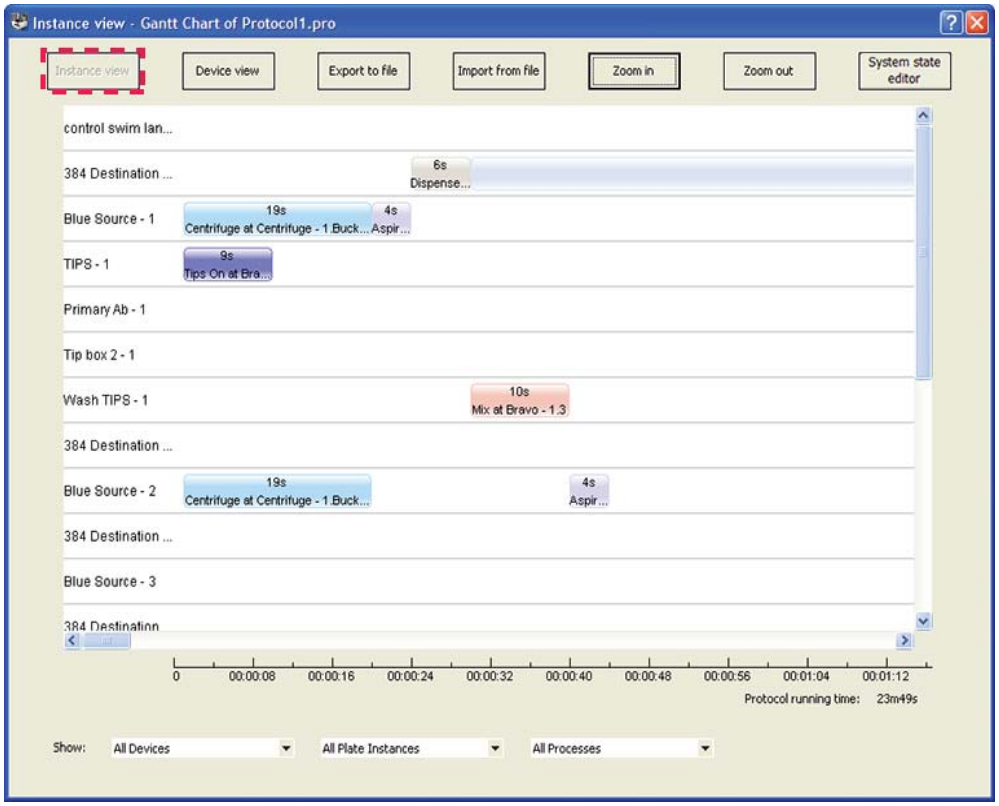
In the VWorks window, select **Tools > Gantt Charts** while a run is in progress, after a run is finished, or when a run has a deadlock error. The Gantt Chart dialog box opens.



Displaying the progress by plate instance

To display the progress by individual plate instances:

In the **Gantt Chart** dialog box, click **Instance view.**



The Instance View presents a graph of the process plate instances (vertical axis) as a function of time (horizontal axis). Each row represents the timeline of a process plate instance. During a run, a row displays the tasks that are performed on a plate instance. The length of the task block represents the duration of the task. The actual duration of each task is displayed on the task block.

Note: A task that does not involve a process plate is not displayed.

If an error occurred during the run, you can use the Instance view to determine the plate instance at which the error occurred. If a deadlock occurred or if the run was aborted, a red X symbol appears on the task at which it occurred.

From within the Instance view, you can double-click a task to edit its parameters.

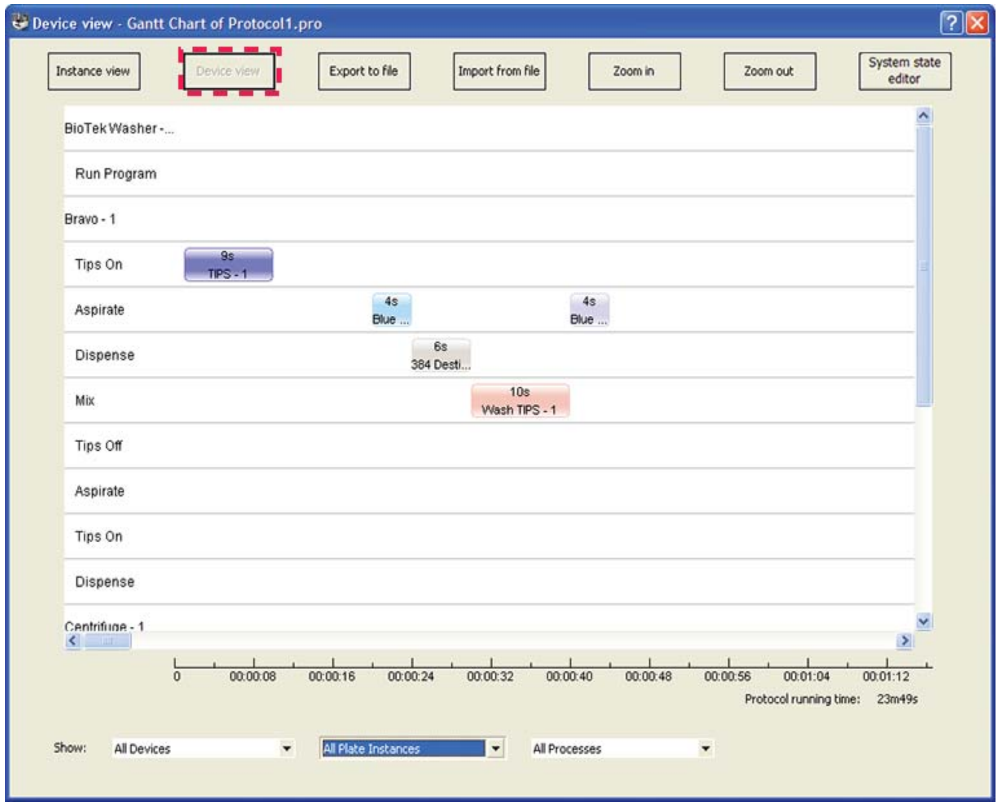
To edit a selected task parameter within the Instance view:

Right-click a task in the graph, and then select **Edit Parameter.** The software minimizes the Gantt Chart dialog box and displays the task within the protocol so that you can edit the parameters.

Displaying the progress by device

To display the progress by device:

In the **Gantt Chart** dialog box, click **Device view.**



The Device View presents a graph of devices (vertical axis) as a function of time (horizontal axis). A row represents the timeline of a device. During a run, each row expands and lists the tasks that the device is performing. The process plate on which the task is performed is shown on the task timeline. The length of the process-plate block represents the duration of the task on the plate.

To optimize the run, look for wide spacing between tasks and determine whether adding devices to the run can improve the throughput. Ideally, the spacing between the tasks should be kept to a minimum.

Zooming the display

You can zoom in or zoom out to change the horizontal (time) scale of the graphs. Each time you zoom in, you are reducing the time increments displayed. Each time you zoom out, you are increasing the time increments displayed.

For example, if the current scale is in increments of 4 seconds, zooming in reduces the increments to 2 seconds. Zooming out increases the increments to 8 seconds.

Note: Zooming has no affect on the scale of the vertical-axis.

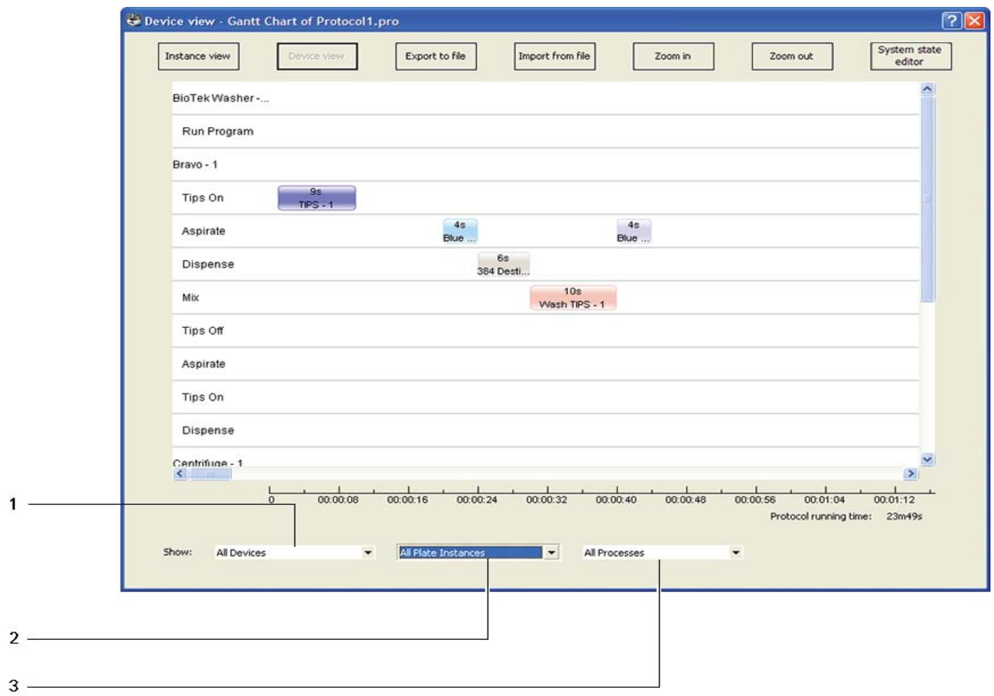
To zoom in or zoom out of the current view:

Click **Zoom in** or **Zoom out.**

Filtering the displayed information

While in the instance or device view, you can filter the information displayed to focus on areas of interest. For example, you can choose to display the information from a single device instead of all devices in the run.

The filter selections are at the bottom of the Gantt Chart dialog box.



To filter the information displayed in the Gantt Chart dialog box:

1 Select one of the following from the Device list:

Selection	Description
All devices	Displays all devices used in the protocol.
Selected devices	Allows you to specify which devices to display in the dialog box.

2 Select one of the following from the Plate Instances list:

Selection	Description
All plate instances	Displays all plate instances.
Completed plate instances	Displays only plate instances that have finished processing.
Plate instances in progress	Displays only plate instances that are currently in progress.

3 Select one of the following from the Process Plate list:

Selection	Description
All processes	Displays all plate processes.
Completed processes	Displays only plate processes that are finished.

Selection	Description
Processes in progress	Displays only plate processes that are currently in progress.
Selected processes	Displays only the plate processes that you have selected.

Opening the System State Editor

You can open the System State Editor to recover from a deadlock error.

To open the System State Editor:

In the **Gantt Chart** dialog box, click **System State Editor.** The System State Editor dialog box opens.

For information about deadlock recovery and the System State Editor, see "Recovering from deadlocks" on page 648.

Note: If you want the System State Editor dialog box to open automatically whenever a deadlock occurs, choose **Tools > Options**. Under **Error Handling** in the Options dialog box, choose **Deadlock behavior > Show the System State Editor**. For more information, see "Recovering from deadlocks" on page 648.

Exporting and importing Gantt charts

To export the run information in the Gantt format:

1. In the **Gantt Chart** dialog box, click **Export to file.** The Save As dialog box opens.
2. Select the folder in which you want to save the file, and type a name for the file, and then click **Save.** A .gnt file is created in the folder you specified.

To import the run information in the Gantt format:

1. In the **Gantt Chart** dialog box, click **Import from file.** The Open dialog box opens.
2. Select the .gnt file, and then click **Open.** The run information appears in the Gantt Chart dialog box.

Related information

For information about...	See...
The System State Editor	"About the System State Editor" on page 648
Monitoring the overall run progress	"Monitoring the overall run progress" on page 254
Viewing logs	"Viewing logs" on page 631

Pausing the run

Ways to pause runs

You can pause runs using one of the following methods:

- "Pausing all runs" on page 261
- "Pausing selected runs in a runset" on page 262

In addition, a protocol run will automatically pause when it reaches a User Message or Wait for User (Bravo) task. For details, see "Pausing at a User Message or Wait for User task" on page 263.

Pausing all runs

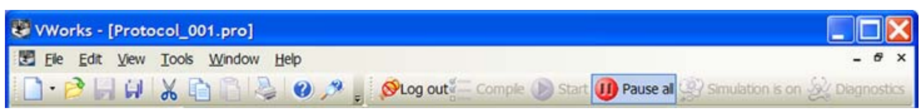
You can pause all runs that are in progress, and then continue the runs when you are ready. Pause all runs when you want to:

- Add or remove labware
- Add buffer to a reservoir
- Diagnose a problem
- Perform an operation that is not part of the protocol

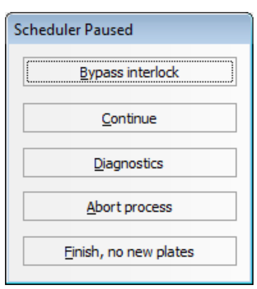
If you need to stop a run in an emergency, press the emergency-stop button on the pendant. See the device user guide for the procedure.

To pause all runs currently in progress:

1 In the **VWorks** window, click **Pause all** on the toolbar.



The Scheduler Paused dialog box opens.



Note: The Bypass interlock button is only available if your system is equipped with an Automation Control Unit and the protocol is not running in simulation mode.

2 *Systems that do not have an Automation Control Unit.* If you moved or removed labware from the system, or if you fixed a device problem so that locations on the device are now available, select **Tools** > **System State Editor**. Change the process plate status and the device location status. For instructions, see "About the System State Editor" on page 648.

3 In the **Scheduler Paused** dialog box, click one of the following:

Command	Description
Bypass interlock	Bypasses the interlock so that you can enter the system. For detailed instructions, see "Bypassing the interlock" on page 264. Bypass interlock is only available if: • Your system is equipped with an Automation Control Unit. • The protocol is not running in simulation mode. *Note:* If you click Bypass interlock, you will not be able to open any diagnostics dialog boxes.
Continue	Resumes the protocol run. **IMPORTANT** When you pause one or more runs, the Orbital Shaking Station stops. Before you continue the run, open Teleshake Diagnostics to restart the shaking.
Diagnostics	Allows you to select the device diagnostic software to open and troubleshoot a problem or perform a manual operation. *Note:* If you click Diagnostics and open a device diagnostics dialog box, you will not be able to open the Bypass Interlock dialog box.
Abort process	Aborts the current command or task in the run. Select Abort if you have determined that the protocol run is not recoverable.
Finish, no new plates	Resumes the protocol run. Processes that are currently in progress will finish. However, no new labware will be delivered into the system.

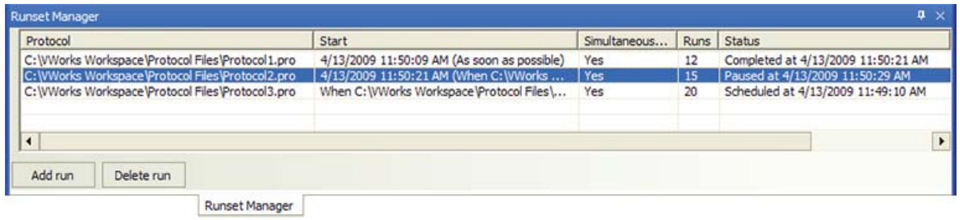
Pausing selected runs in a runset

If you want to bypass the interlock and enter the system, use the Pause All command. See "Pausing all runs" on page 261. To stop a run in an emergency, use the emergency-stop button on the pendant. See the safety instructions for the system, workstation, or device for details.

To pause one protocol run in the Runset Manager:

1 In the **Runset Manager** tab, right-click the protocol you want to pause.

2 In the shortcut menu that appears, select **Pause this run.** The selected protocol is paused, and its status is shown in the Runset Manager.

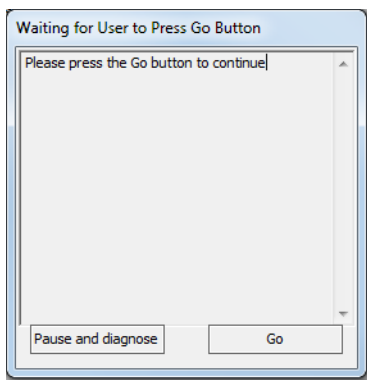


Alternatively, you can select **Temporarily prevent new plates from entering system** in the shortcut menu to allow other runs to resume. Processes that are currently in progress will finish. No new labware will be delivered into the system.

Pausing at a User Message or Wait for User task

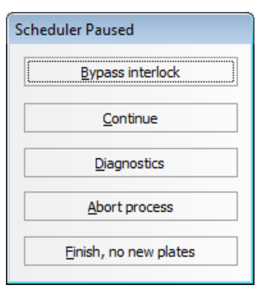
A run will automatically pause when it encounters a User Message or Wait for User task. These tasks pause the run to remind the operator to perform a task manually, such as refill a reservoir or replace labware.

When the system pauses, the message or Waiting for User dialog box appears.



To confirm that you want to pause or to diagnose a problem:

1 Click **Pause and Diagnose**. The Scheduler Paused dialog appears.



Note: Bypass interlock is only available if your system is installed with an Automation Control Unit and the protocol is not running in simulation mode.

2 *Systems that do not have an Automation Control Unit*. If you moved or removed labware from the system, or if you fixed a device problem so that locations on the device are now available, select **Tools** > **System State Editor**. Change the process plate status and the device location status. For instructions, see "About the System State Editor" on page 648.

3 In the **Scheduler Paused** dialog box, click one of the following:

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>Bypass interlock</td><td>Bypasses the interlock so that you can enter the system and refill a reservoir or replace labware. For detailed instructions, see “Bypassing the interlock” on page 264.

Bypass interlock is only available if:
• Your system is installed with an Automation Control Unit.
• The protocol is not running in simulation mode.

Note: If you click Bypass interlock, you will not be able to open any diagnostics dialog box.</td></tr>
<tr><td>Continue</td><td>Resumes the protocol run.

IMPORTANT When you pause one or more runs, the Orbital Shaking Station stops. Before you continue the run, open Teleshake Diagnostics to restart the shaking.</td></tr>
<tr><td>Diagnostics</td><td>Allows you to select the device diagnostic software to open and troubleshoot a problem or perform a manual operation.

Note: If you click Diagnostics and open a device diagnostics dialog box, you will not be able to open the Bypass Interlock dialog box.</td></tr>
<tr><td>Abort process</td><td>Aborts the current command or task in the run. Select Abort if you have determined that the protocol run is not recoverable.</td></tr>
<tr><td>Finish, no new plates</td><td>Resumes the protocol run. Processes that are currently in progress will finish. However, no new labware will be delivered into the system.</td></tr>
</table>

Bypassing the interlock

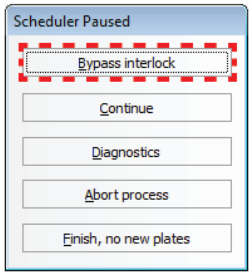
Note: This section is only applicable to systems that have an Automation Control Unit installed.

When the protocols are paused, you can bypass the interlock and enter the system to manually perform an operation, such as replacing labware or refilling a reservoir.

IMPORTANT For safety reasons, when the protocols are paused and the interlock is bypassed, the system will not be able to start or perform any operation until you resume the run.

To bypass the interlock:

1 In the Scheduler Paused dialog box, click **Bypass interlock**.



The Bypass Interlock dialog box appears.

2 Check the message at the top of the Bypass Interlock dialog box:

- *Wait*. The interlock will remain armed while devices are finishing their current tasks. You must wait until all of the devices have paused before entering the system.

  CAUTION Entering the system while devices are performing tasks will trip the interlock, potentially causing the run to be unrecoverable.

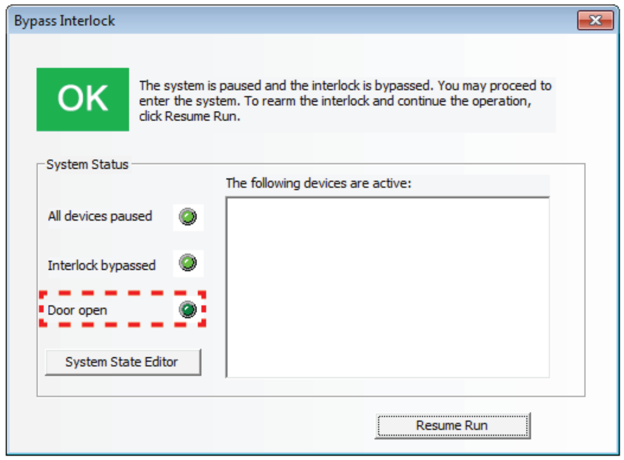
- *OK*. When all of the devices have paused, the interlock will be bypassed, and you can enter the system.

  Note: The yellow INTERLOCK BYPASS light on the Automation Control Unit turns on.

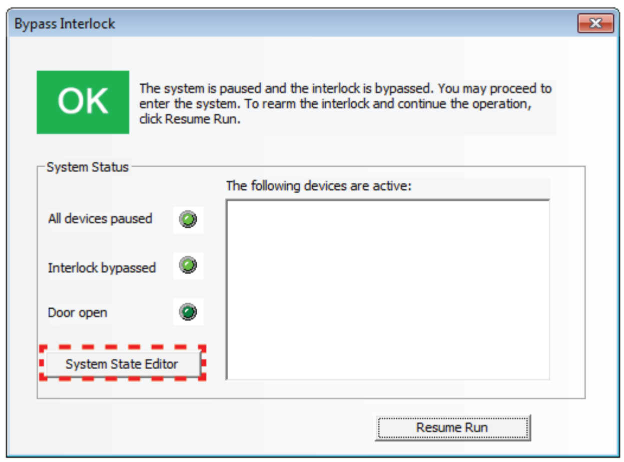
3 When you are finished with your task inside the system, close the system doors or move away from the Light Curtain. In the Bypass Interlock dialog box, check that the **Door open** indicator light is off.

If the Door open indicator light is on, make sure the doors are closed fully or remove any object that is interrupting the Light Curtain.

CAUTION Make sure the **Door open** light is off before you resume the run. Resuming the run arms the interlock. An open door or interrupted Light Curtain will trip the interlock, potentially causing the run to be unrecoverable.



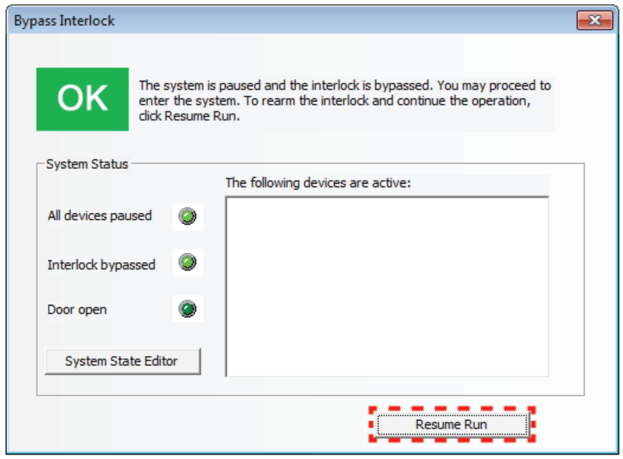
4 If you replaced or removed labware from the system, or if you fixed a device problem so that locations on the device are now available, click **System State Editor** in the **Bypass Interlock** dialog box.



5 In the **System State Editor** dialog box, update the labware, device, or location information. For instructions, see "About the System State Editor" on page 648. When you are finished, click **Accept All Changes** in the System State Editor.

6 In the **Bypass Interlock** dialog box, click **Resume Run** to rearm the interlock and return to the Scheduler Paused dialog box.

> **CAUTION** Make sure the **Door open** light is off before you resume the run. Resuming the run arms the interlock. An open door or interrupted Light Curtain will trip the interlock, potentially causing the run to be unrecoverable.



Note: The green INTERLOCK NORMAL light on the Automation Control Unit turns on.

Related information

For information about...	See...
The System State Editor	"About the System State Editor" on page 648
Viewing logs	"Viewing logs" on page 631
Monitoring the protocol run	"Monitoring the overall run progress" on page 254
User Message task	"User Message" on page 599
Wait for User (Bravo) task	"Wait for User (Bravo)" on page 605

Stopping the run

If you want to stop a run and later continue the run, use one of the pause methods described in "Pausing the run" on page 261.

If you need to stop a run in an emergency, use the hardware Emergency Stop button. See the device user guide for the procedure.

CAUTION You cannot resume a protocol run after you press the hardware Emergency Stop button. To recover the system after an emergency stop, see the device user guide.

Related information

For information about...	See...
Pausing runs	"Pausing the run" on page 261
Starting runs	"Starting the protocol run" on page 237
Managing runsets	"Managing runsets" on page 243
Monitoring runs	"Monitoring the overall run progress" on page 254

7
Setting parameters for I/O-handling tasks

This chapter contains the following topics:

Digital Output

Description

The Digital Output task (Digital Output) changes the state of a digital signal. For example, you can use the task to turn on, turn off, open, or close the following:

- Alarm
- UV lamp
- Vacuum pump
- Ventilation fan
- Waste bin door

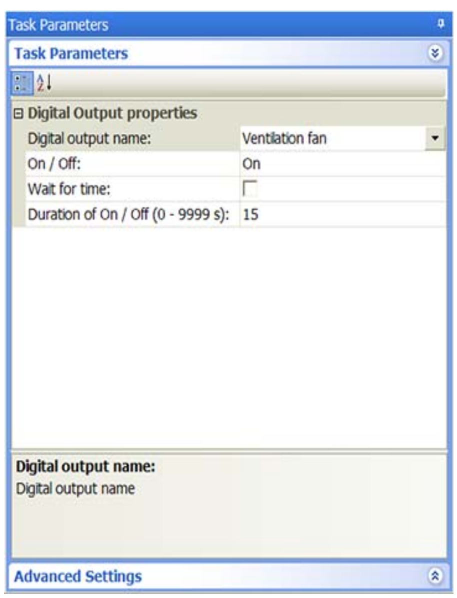
You can also specify the length of time to leave the digital signal in the new state.

The digital signals you can turn on or off are configured in the IO Manager and the I/O device diagnostics (such as ACU Diagnostics). For details, see "Managing digital signals" on page 671 and the I/O device user documentation, such as the *Automation Control Unit User Guide*.

Task is available for...	Task is available in...
Automation Control Unit, BioCel I/O Interface, or equivalent device (Digital Output)	Startup Protocol Main Protocol Cleanup Protocol

Task parameters

After adding the Digital Output task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Digital output name	The name of the item whose state you want to change. Select from the list of names. The names are set up in the I/O device diagnostics.
On/Off	The new state of the item. Select On or Off. When you select On, the task icon changes to Set Digital Output. When you select Off, the task icon changes to Clear Digital Output.
Wait for time	The option to turn on or turn off the selected item and wait the specified duration before advancing to the next task. If you select the option, make sure you add another Digital Output task to reverse the state of the item at the desired point in the protocol. You must also select this option in the second task. If you do not select the option, the task will turn on or off the selected item, and then advance to the next task while the item is in the new state. If you specified duration, the item will reverse to its previous state after the duration ends.
Duration of On/Off (s)	The length of time, in seconds, the item should remain in the new state. For example, if you want to leave the UV lamp on for 15 seconds, type 15. At the end of the specified duration, the item reverses to its previous state. If you specify a duration, you do not need to add a second Digital Output task to reverse the state. If Wait for time is selected, this is the length of time, in seconds, the task waits before advancing to the next task.

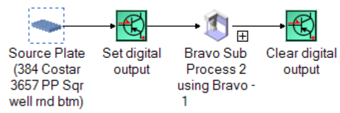
Example 1: Turn on a ventilation fan while reagent is dispensed

Goal

The ventilation fan must be on while a particular reagent is being dispensed on the Bravo Platform.

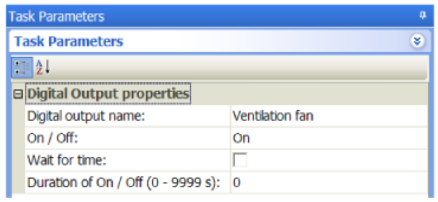
Implementation

Add a Digital Output task before and after the Bravo Subprocess task that contains the dispensing subroutine. The first Digital Output task (Set digital output) turns on the fan. The second Digital Output task (Clear digital output) turns off the fan.

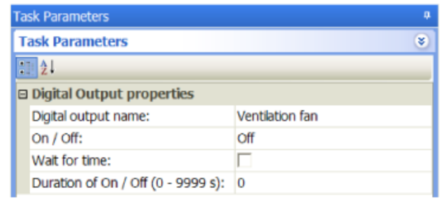


The task parameters for the digital output tasks are:

Set digital output



Clear digital output



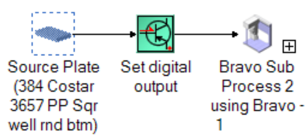
Example 2: Turn on a ventilation fan for 15 seconds

Goal

The ventilation fan must be on for 15 seconds while a particular reagent is being dispensed on the Bravo Platform.

Implementation

Add a Digital Output task before the Bravo Subprocess task that contains the dispensing subroutine. The Digital Output task (Set digital output) turns on the fan for 15 seconds. A second Digital Output task (Clear digital output) is not necessary.



The task parameters for the Digital Output task are:

Related information

For information about...	See...
Adding devices	"Adding devices" on page 25
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Wait for Input task	"Wait for Input" on page 274
Assigning signals to lights, alarms, pass-through gates, and sensors	"Managing digital signals" on page 671

For information about...	See...
I/O device diagnostics	I/O device user documentation, such as the *Automation Control Unit User Guide*

Wait for Input

Description

The Wait for Input task (Wait For Input) requires that one or more conditions are met before starting the next task in the protocol. For example, you can require that the system environment reach a certain temperature and humidity before the Main Protocol starts.

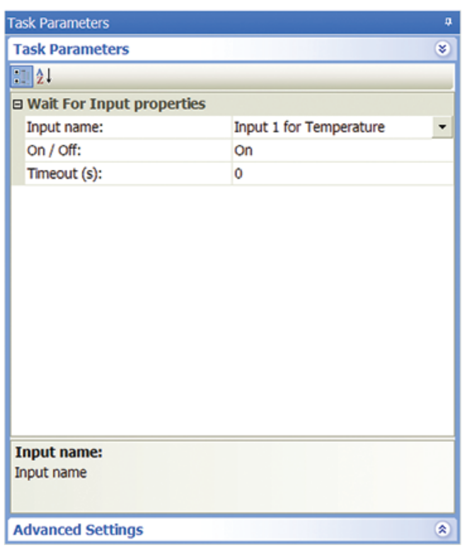
The digital signals you can turn on or off are configured in the IO Manager and the I/O device diagnostics (such as ACU Diagnostics). For details, see “Managing digital signals” on page 671 and the I/O device user documentation, such as the *Automation Control Unit User Guide*.

Task is available for...	Task is available in...
Automation Control Unit, BioCel I/O Interface, or equivalent (Wait For Input)	Startup Protocol Main Protocol Cleanup Protocol

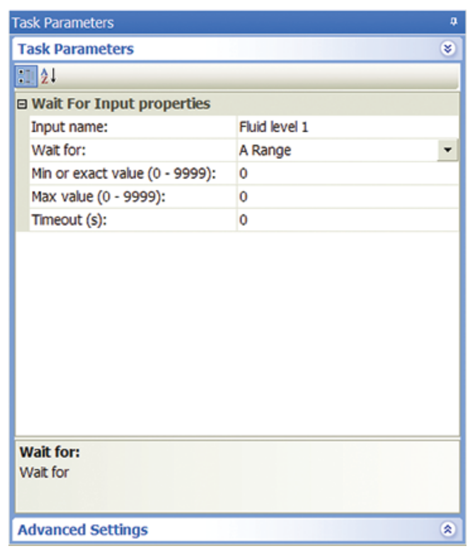
Task Parameters

After adding the Wait for Input task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:

Digital



Analog



Parameter	Description
Input name	The name of the condition you want to check. Select from the list of condition names. You can select either a digital input or an analog input condition. The names are set up in the I/O device diagnostics.
On/Off	The new state of the item. This parameter appears only if you select a digital input condition.
Wait for	The condition to be met: • Exactly • A range • At least • At most This parameter appears only if you select an analog input condition.
Min or exact value	The minimum value or the exact value of the condition. Examples: • If you want to specify an exact value of 50, you can type 50 in this field. • If you want to specify a minimum (at least) value of 100, type 100 in this field. • If you want to specify a range such as 90 to 100, type 90 in this field, and then type 100 in the **Max value** field. This parameter appears only if you select an analog input condition.
Max value	The maximum value of the condition. Examples: • If you want to specify a maximum (at most) value of 110, type 110 in this field. • If you want to specify a range such as 90 to 100, type 90 in the **Min or exact value** field, and then type 100 in this field. This parameter appears only if you select an analog input condition.
Timeout (s)	The length of time, in seconds, the software should wait for the condition to be met before taking the action you specify in the **On timeout** field. *Note:* 0 means wait indefinitely.

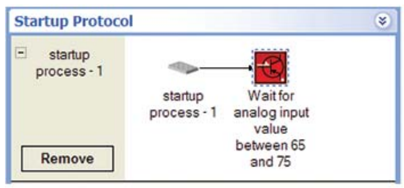
Example

Goal

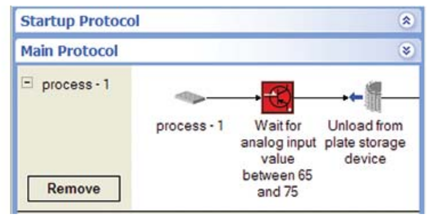
The humidity within the system chamber must be at least 65% RH before the protocol run can start. The humidity should not exceed 75% RH. The software will wait indefinitely for the condition to be satisfied.

Implementation

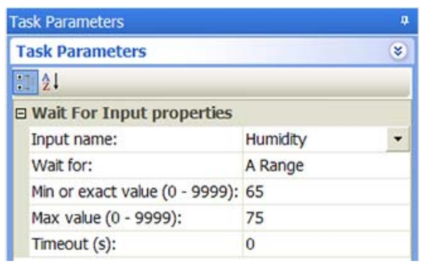
Configure the I/O device to include a humidity signal channel (in this example, an analog input channel). Add a Wait for Input task in the Startup Protocol or at the beginning of the Main Protocol.



or



The task parameters for the Wait for Input task are:



Related information

For information about...	See...
Adding devices	"Adding devices" on page 25
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Digital Output task	"Digital Output" on page 270
Assigning signals to lights, alarms, pass-through gates, and sensors	"Managing digital signals" on page 671
I/O device diagnostics	I/O device user documentation, such as the *Automation Control Unit User Guide*

8
Setting parameters for microplate-handling tasks

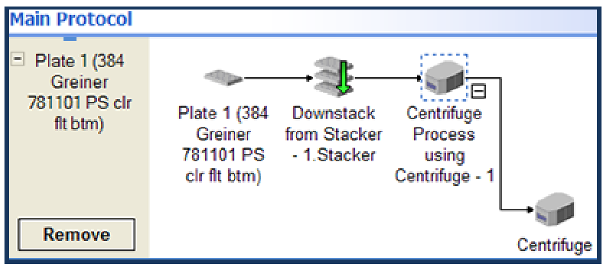
This chapter contains the following topics:

Centrifuge Process

Description

IMPORTANT The Centrifuge Process task is primarily used in protocols created in VWorks software, installer 11.1 or earlier. If you are running VWorks software, installer 11.2 or later, be sure to use the CentrifugeAuto task instead of the Centrifuge Process task. For instructions, see "CentrifugeAuto" on page 286.

The Centrifuge Process task for the Centrifuge (Centrifuge Process (Centrifuge)) and the Centrifuge with Loader (Centrifuge Process (Centrifuge Loader)) indicates the start of a protocol subroutine that employs the Microplate Centrifuge (or Centrifuge). Within the subprocess, the software automatically adds the (Centrifuge) task to spin microplates according to the task parameter settings.



One microplate with a counterweight or two microplates can be spun at one time. Counterweights can be placed manually or robotically.

Task is available for...	Task is available in...
Centrifuge, or Centrifuge with Loader	Main Protocol

Task parameters

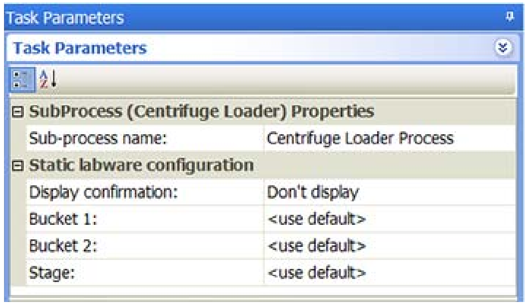
You set parameters for:

- *Centrifuge Process*. Contains the counterweight position information. For more information, see "Centrifuge Process" on page 278.
- *Centrifuge task*. Contains the spin parameters and counterweight selections. For more information, see "Centrifuge" on page 279.

Centrifuge Process

IMPORTANT For BioCel System protocols, you can optimize the placement of microplates in the Microplate Centrifuge by adding a Place Plate (to platepad) task between two consecutive Centrifuge Process tasks or between an Unload and Centrifuge Process task.

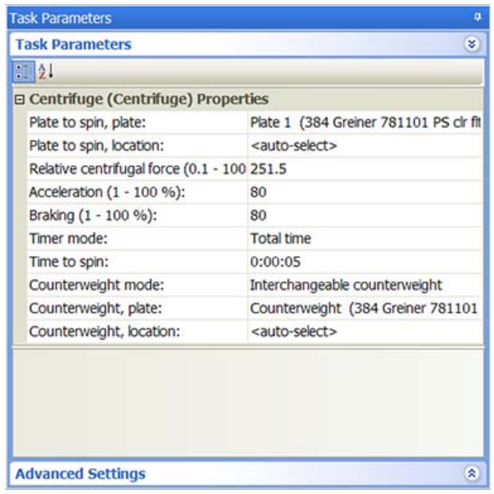
After you add the Centrifuge Process task (subprocess) at the desired point in the protocol, select the subprocess icon and set the following parameters in the **Task Parameters** area:



Parameter	Description
Sub-process name	The name of the subprocess. Select from the list of available subprocesses currently in the protocol.
Display confirmation	The option to display a message at the beginning of the protocol run to remind you to verify the physical locations of the labware match what you specified in the software.
Bucket 1/Bucket 2	The counterweight you want to use in the bucket location. You can select from the list of labware for either Bucket 1 or Bucket 2. The counterweight selection depends on the counterweight mode you select in the Centrifuge task: • If you are using the **Interchangeable counterweight** or **Use 2 protocol plates** counterweight mode, select **<use default>.** • If you are using the **Fixed Counterweight** mode, select a labware for either Bucket 1 or Bucket 2. For a description of the counterweight modes, see "Centrifuge" on page 279.
Stage (*Centrifuge with Loader only*)	The stage at the Centrifuge Loader.

Centrifuge

When you add the Centrifuge Process task (subprocess) at the desired point in the protocol, the Centrifuge task is automatically added. Select the Centrifuge task and set the following parameters in the **Task Parameters** area:



Parameter	Description
Plate to spin, plate	The microplate that will be spun. *Note:* Select the counterweight microplate only if you plan to use the **Fixed counterweight** mode.
Plate to spin, location	The device location to use. For example, you can select a centrifuge bucket. <auto-select> automatically places the labware at the first-available or appropriate location for the task. Make sure the **Plate to spin, location** selection does not conflict with the **Counterweight, location** selection.
Relative centrifugal force (0.1–1006.2 × g)	The rotor velocity, as a multiple of gravity.
Acceleration (1–100%)	The rate of centrifugation, as a percent of maximum acceleration.
Braking (1–100%)	The deceleration of the centrifuge, as a percent of maximum deceleration.
Timer mode	How the specified spin time is implemented: • **Total time.** The specified spin time includes acceleration and braking. • **Time at speed.** The specified spin time does not include acceleration and braking.
Time to spin	The length of time to spin the microplates in the desired mode.

Parameter	Description
Counterweight mode	The counterweight method used. Select one of the following: • **Fixed counterweight.** The counterweight is manually placed in the centrifuge before starting the run. The counterweight remains in the centrifuge during the entire run. You must remember to place the counterweight in the device before the run and remove it after the run is finished. **IMPORTANT** Be sure to select the counterweight in the Centrifuge Process Task Parameters area. See "Task parameters" on page 278. • **Interchangeable counterweight.** The counterweight is on a platepad or Plate Hub Carousel. The robot will move it into the centrifuge during the run and return it to the platepad when the run is finished. If you select this mode, use the **Counterweight, plate** parameter to select the counterweight you want to use. • **Use 2 protocol plates.** Instead of a counterweight, a second process plate is used during the run. You must process two or more microplates in this mode. During the run, the robot places the first two microplates into the centrifuge. After spinning, the robot replaces the two microplates with the next pair, and so on. If you are processing an odd number of microplates, and three microplates are remaining, the robot will not remove the second from the last microplate so that the last microplate is paired. If you are processing an odd number of microplates, and: – A process contains only one Centrifuge Process task and it uses this counterweight mode, set the **Simultaneous plate** parameter to 2. – A process contains two or more Centrifuge Process tasks, and they all use this counterweight mode, set the **Simultaneous plates** parameter to 3 or greater.

Parameter	Description
Counterweight, plate	The microplate or labware you want to use as the counterweight. This parameter is only available if you selected the **Interchangeable counterweight** method. If the list of counterweights do not appear, make sure: • You have added a device such as the Platepad for the counterweight. See "Adding devices" on page 25 for instructions. • You have configured the labware correctly. See "Configuring labware" on page 40 for instructions.
Counterweight, location	The device location to use. For example, you can select a centrifuge bucket. <auto-select> automatically places the labware at the first-available or appropriate location for the task. Make sure the **Counterweight, location** selection does not conflict with the **Plate to spin, location** selection.

Device selection

You must select a device for Centrifuge Process tasks. If you have multiple devices of the same type, you can:

- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the list. However, if all of the devices in the list are in an error state, the software will automatically use the device in the backup pool.

IMPORTANT The multiple devices must have the same setup and configuration.

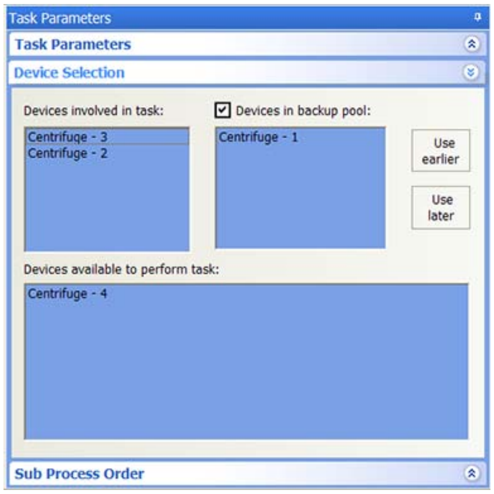
After adding the Centrifuge Process task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click the desired device in the **Devices available to perform task** area to move it to the **Devices involved in task** area. If you have multiple devices of the same type, you can move them to the **Devices involved in task** area.
2. If you have multiple devices in the **Devices involved in task** area, select a device, and then click **Use earlier** or **Use later** to prioritize it.
3. *Optional.* Select backup devices to use in case the primary device in the **Devices involved in task** area encounters an error.
   a. Select **Devices in backup pool.**

b Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.

c If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



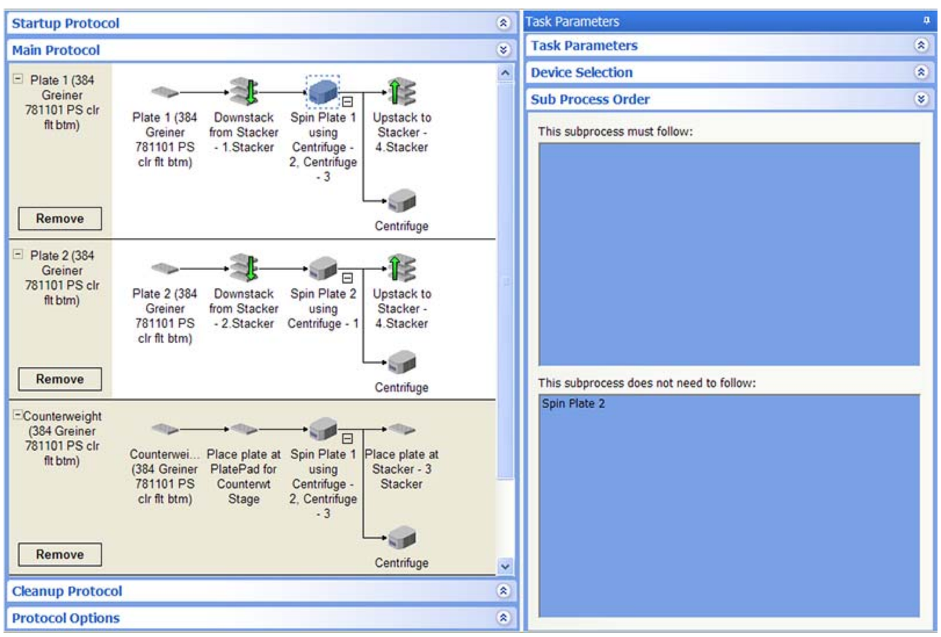
Subprocess order

If more than one subprocess uses the same configured labware, and the subprocesses are in different protocol processes, you can specify the sequence in which the subprocesses will be performed.

To specify the sequence in which the subprocesses will be performed on the same configured labware:

1 In the protocol process, select the sub-process that contains the task that uses the configured labware.

In the following example, the Spin Plate 1 process is selected.



2 In the **Task Parameters** area, click **Sub Process Order.**

3 In the **Sub Process Order** area, double-click the subprocess names to rearrange the order.

In the example shown in step 1, the order is Spin Plate 1 and Spin Plate 2.

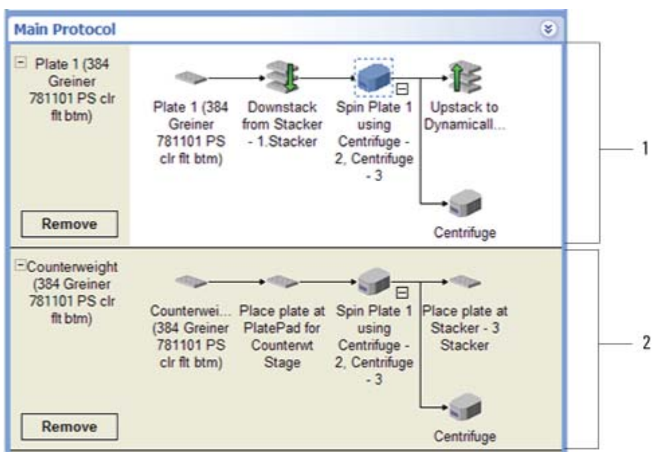
Example: Interchangeable counterweight mode

Goal

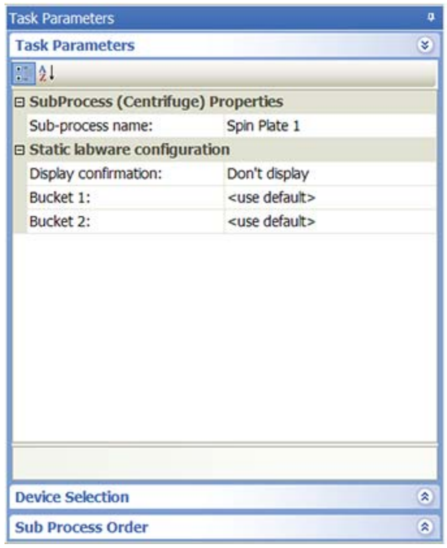
Downstack microplates from a Labware Stacker, spin the microplates in the Microplate Centrifuge, and then upstack the microplates to an available Labware Stacker. Use the counterweight that is stored on a designated platepad.

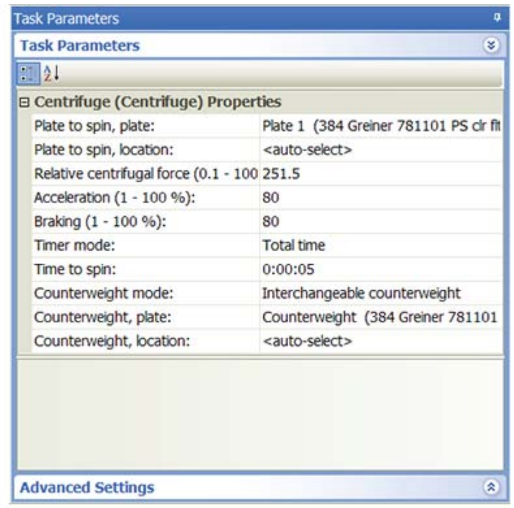
Implementation

(1) Add a process for microplates that will be downstacked, spun, and upstacked. Add a Downstack, Centrifuge Process, and Upstack task in the order shown. (2) Configure the counterweight on a platepad and call it Counterweight.



The task parameters for the Centrifuge Process and the Centrifuge task are shown. Notice that the Bucket n parameters are set to <use default>, because the Interchangeable counterweight mode is selected.





Because the robot will move the counterweight from the designated platepad to the centrifuge during the run, select the Interchangeable counterweight mode. For the **Counterweight, plate** parameter, select Counterweight.

Note: The robot will move the counterweight back to the platepad when the run is finished.

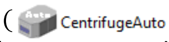
Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Configuring labware	"Configuring labware" on page 40
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

CentrifugeAuto

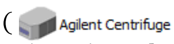
Description

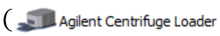
Note: If you want to manage counterweights and counterweight modes using JavaScript, see the examples in "Using JavaScript with the CentrifugeAuto task" on page 108.

If you have VWorks software installer 11.2, make sure that you use the CentrifugeAuto task. The CentrifugeAuto task (

) performs all the functions of the Centrifuge Process task but does not require a subprocess.

The CentrifugeAuto task (

) can be used with the Microplate Centrifuge (

) and Centrifuge with Loader (

) devices to spin microplates according to the task parameter settings. The CentrifugeAuto task enables you to spin one microplate with a counterweight or spin two microplates from a protocol process, at the same time. Counterweights can be placed manually or robotically. When using counterweights, you can use multiple counterweights for protocols with multiple processes. This is accomplished by using the Configure Labware feature to manage the centrifuge tasks.

Task is available for...	Task is available in...
Centrifuge	Main Protocol
Centrifuge with Loader	Main Protocol

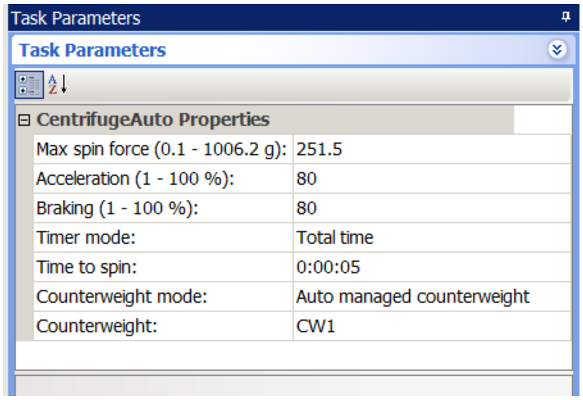
The CentrifugeAuto task cannot be used in the following cases:

- Within a Loop task
- Within a Spawn process as a subroutine, as this will result in a deadlock
- Within a Macro
- As the first task in a process

You cannot assign multiple devices to a single CentrifugeAuto task.

Task parameters

After adding the CentrifugeAuto task in the main protocol, set the **CentrifugeAuto Properties** in the **Task Parameters** area:



Parameter	Description
Max spin force (0.1 - 1006.2 × g)	The rotor velocity, as a multiple of gravity.
Acceleration (1–100%)	The rate of centrifugation, as a percent of maximum acceleration.
Braking (1–100%)	The deceleration of the centrifuge, as a percent of maximum deceleration.
Timer mode	The way the spin time is implemented: • **Total time.** The specified spin time includes acceleration and braking. • **Time at speed.** The specified spin time does not include acceleration and braking.
Time to spin	The length of time to spin the microplates in the desired mode.
Counterweight mode	The method used to keep the Centrifuge balanced during a spin.
Fixed counterweight	The manual mode for placing the counterweight. Place the counterweight in the device before the run (in bucket 2) and remove it after the run is finished. **CAUTION** Do not use the Fixed counterweight mode with other counterweight modes on the same device. Running a protocol with mixed counterweight modes will result in a compiling error. Ignoring the error and continuing to run the protocol could result in a collision.

Parameter	Description
Auto managed counterweight	A counterweight managed as a **Configure Labware** process. The robot will move the counterweight from a platepad or Plate Hub Carousel (specified by the plate process) into the Centrifuge during the run and return it to the platepad when the run is finished. If you select this mode, use the **Counterweight** parameter to select the **Configure Labware** process that corresponds to the counterweight you want to use. **Guidelines for use** To avoid deadlocks when there are two or more CentrifugeAuto tasks in a protocol: If the tasks are placed before pipetting tasks for both source and destination processes, and if the destination plate is spun first, set up an additional temporary-storage location for the destination plate. This mode can be used in combination with the Use 2 protocol plates mode in the same protocol. See “Guidelines for use with two modes (Auto managed and Use 2 protocol plates)” on page 289 for additional guidelines.
Use 2 protocol plates	A second process plate used as the counterweight. During the run, the robot places the first two microplates into the centrifuge. After spinning, the robot replaces the two microplates with the next pair, and so on. If you are processing an odd number of microplates, and three microplates are remaining, the robot will not remove the second from the last microplate so that the last microplate is paired. **Guidelines for use** To avoid deadlocks, whenever using this task, set **Simultaneous plates** to 2 or greater and run the protocol at least twice. To avoid deadlocks when there are two or more CentrifugeAuto tasks in a protocol: • If the tasks are within the same process and are using different devices and if Simultaneous plates is set to 2, run an even number of microplates. • If the tasks are within the same process and are using the same device and if you are running an odd number of plates, set up an additional temporary storage location to accommodate the odd plate. • If the tasks are placed before pipetting tasks for both source and destination processes, and if the destination plate is spun first, set up an additional temporary-storage location for the destination plate. • If using the same Centrifuge with Loader, ensure the tasks are not adjacent to each other. *Note:* This is true only for the Centrifuge with Loader and not for the Centrifuge. This mode can be used in combination with the Auto managed mode in the same protocol. See “Guidelines for use with two modes (Auto managed and Use 2 protocol plates)” on page 289 for additional guidelines.

Parameter	Description
Counterweight	A configured labware used as the counterweight. This parameter is available only if you selected the **Auto managed counterweight** mode. Tip boxes and reservoirs, or any other labware not compatible with the Microplate Centrifuge or Centrifuge with Loader, cannot be used as counterweights. If the list of counterweights do not appear, make sure you have configured the labware correctly. See "Example: Auto managed counterweight mode" on page 290 for instructions.

Guidelines for use with two modes (Auto managed and Use 2 protocol plates)

Use the following guidelines to avoid deadlocks when the protocol has two or more CentrifugeAuto tasks, and one task uses the Auto managed mode and another uses the Use 2 protocol plates mode.

Multiple CentrifugeAuto tasks using the same device

If the tasks are using the same device, run an even number of plates.

A Reorder task is between two CentrifugeAuto tasks

- If using the Reorder task between two CentrifugeAuto tasks with the same device, and if placing the auto-managed CentrifugeAuto task before the Reorder task, set the **Number of plates** to store in the Reorder task to an even number.
- If using the Reorder task between two CentrifugeAuto tasks and if placing the Use 2 protocol plates mode CentrifugeAuto task before the Reorder task, set the **Number of plates** to store in the Reorder task to an even number.

A long Incubation task is between two CentrifugeAuto tasks

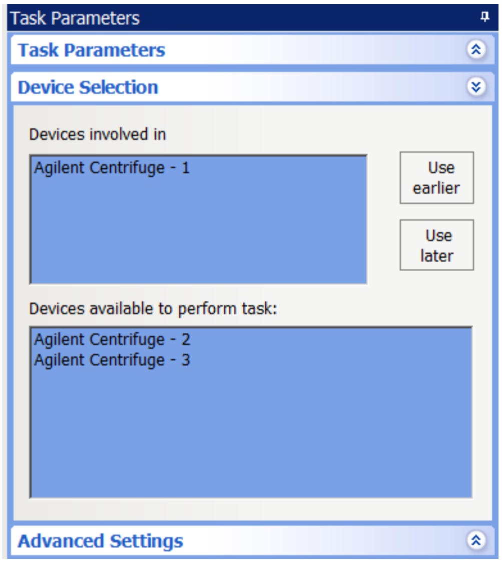
If placing a long Incubation task between two CentrifugeAuto tasks that use the same device, and if the Auto managed CentrifugeAuto task is before the Incubation task, set the number of incubation locations to an even number.

Device selection

You must select a device for each CentrifugeAuto task. After adding the CentrifugeAuto task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

Double-click the desired device in the **Devices available to perform task** area to move it to the **Devices involved in task** area. Only one device can be assigned to a given task.



Example: Auto managed counterweight mode

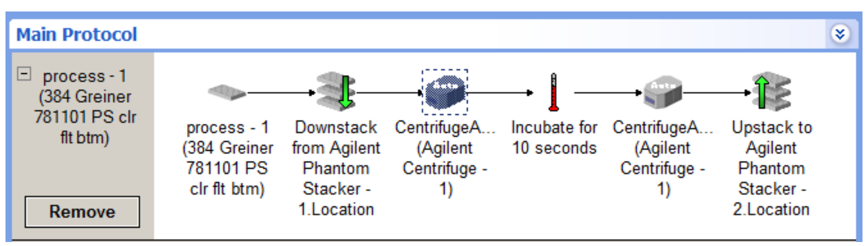
Goal

Perform two spins on one microplate with a different task between spins, in this case, an incubate task. Both spins are performed with the same device and each spin uses a different counterweight.

Implementation

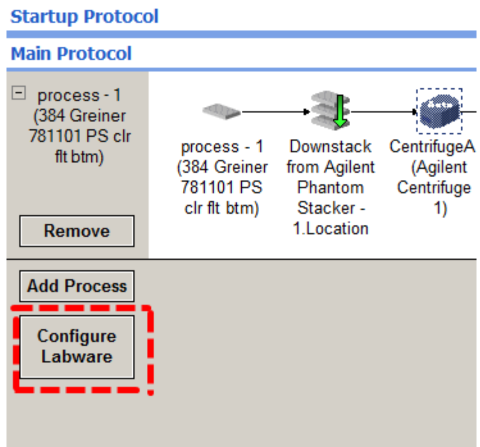
1 Create a protocol process that includes a Downstack, Spin, Incubate, a second Spin, and an Upstack task.

2 Select the parameters in the **Task Parameters** area for the Downstack, Incubate, and Upstack tasks.



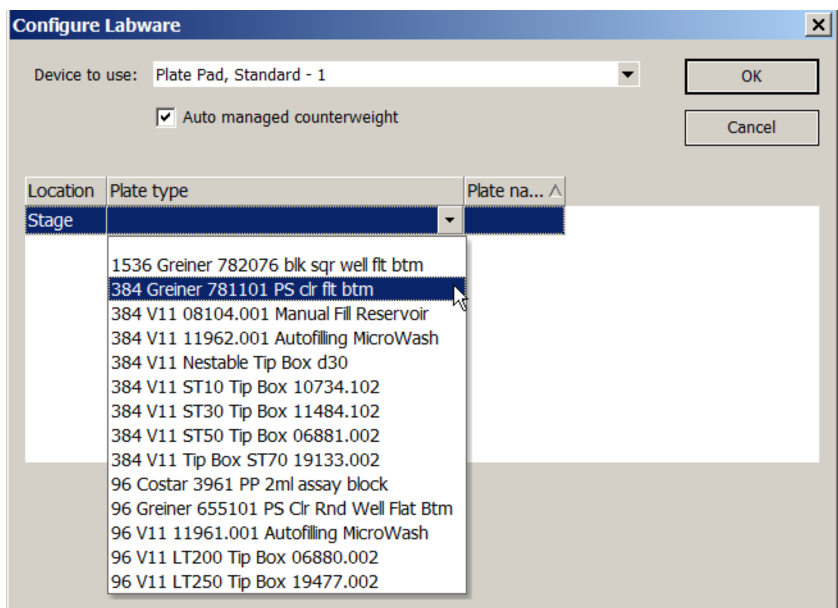
3 Create a Configure Labware process for each counterweight as follows:

a Click **Configure Labware**.

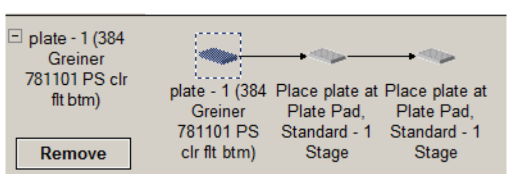


b In the **Configure Labware** dialog box, do the following:

- Select the location of a counterweight from the **Device to use** list.
- Select **Auto managed counterweight**.
- Select the microplate from the **Plate type** list.
- Click **OK**.

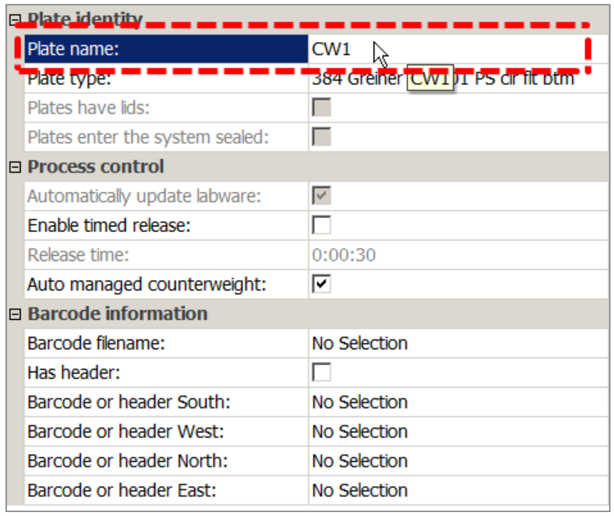


The configure labware process appears.



c In the **Task Parameters** area, type a name for the Configure Labware process in the **Plate name** field (in this case, **CW1**).

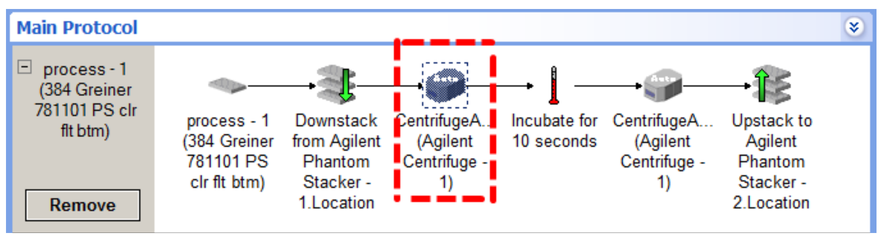
Note: When the **Auto managed counterweight** option is selected, the **Simultaneous plates** and **Automatically update labware** options are not available.



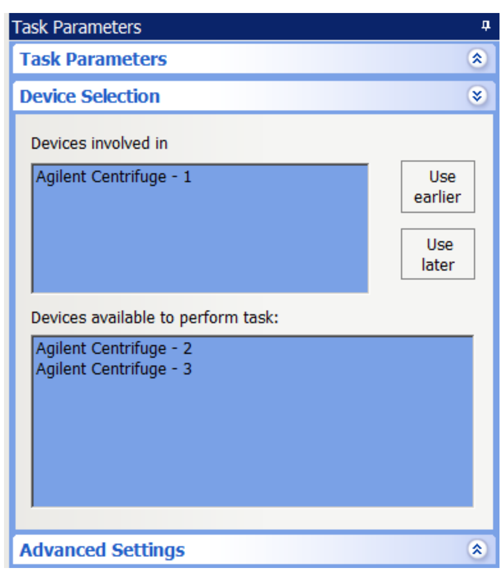
d Repeat steps 3a through 3c for the second counterweight.

4 Select the device for the **CentrifugeAuto** tasks:

a Select the first **CentrifugeAuto** task in the process.



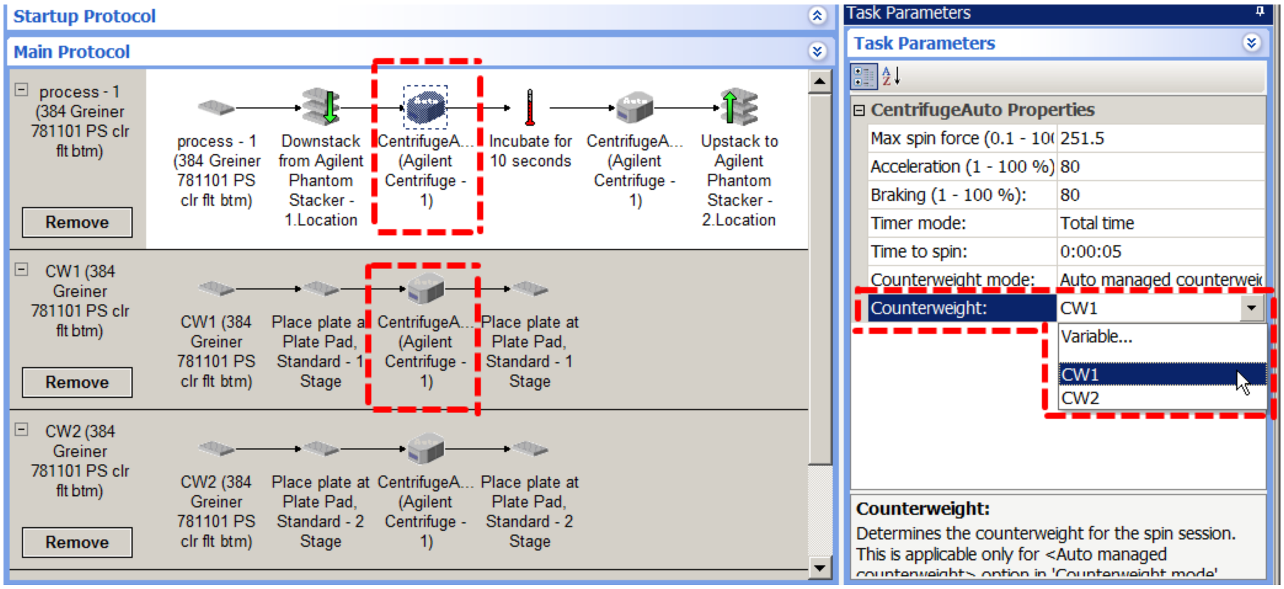
b In the **Device Selection** area, drag **Agilent Centrifuge-1** to the **Devices involved in task** area.



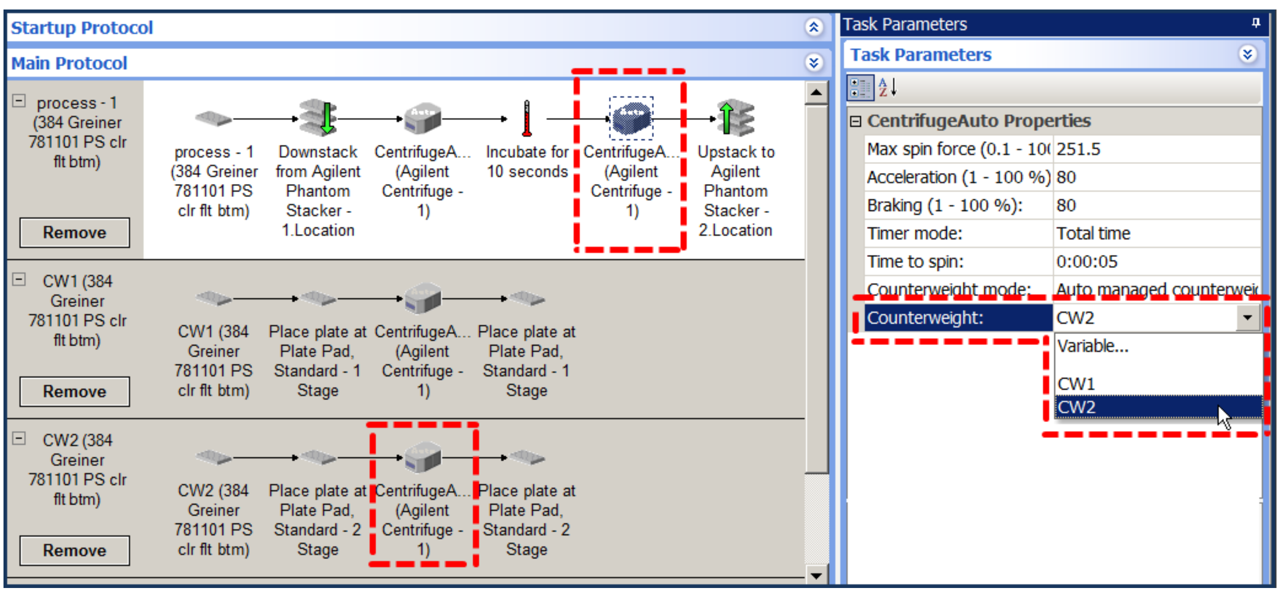
c Repeat steps 4a and 4b for the second **CentrifugeAuto** task.

5 When setting the parameters of the first **CentrifugeAuto** task, select the **Auto managed counterweight** mode, and then select **CW1** from the **Counterweight** list.

Note: A CentrifugeAuto task is automatically added to the Configure Labware process chosen for the counterweight.



6 When setting the parameters of the second **CentrifugeAuto** task, select the **Auto managed counterweight** mode, and then select **CW2** from the Counterweight list.



7 Before you start the run, make sure the counterweights are at their starting locations.

Plate movement

When the protocol runs, the robot will place the downstacked microplate in bucket 1 of the centrifuge and the counterweight, CW1, into bucket 2. Counterweight CW1 will be swapped out for counterweight CW2 after the microplate is incubated and after it is placed back into bucket 1 of the Centrifuge. Both counterweight plates will automatically be placed back on their starting locations at the end of the process.

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Configuring labware	"Configuring labware" on page 40
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

Configure Static Labware

Description

IMPORTANT The VWorks Automation Control software is backward-compatible with protocols created in VWorks4 version 6.2.3 or earlier and will continue to support static labware configuration procedures. However, Agilent Technologies recommends that you use the concept and procedure in "Configuring labware" on page 40 when writing new protocols.

Available only in the Startup protocol, the Configure Labware task (Configure Labware) allows you to assign static labware to locations on a device. The labware will remain at these locations during the protocol.

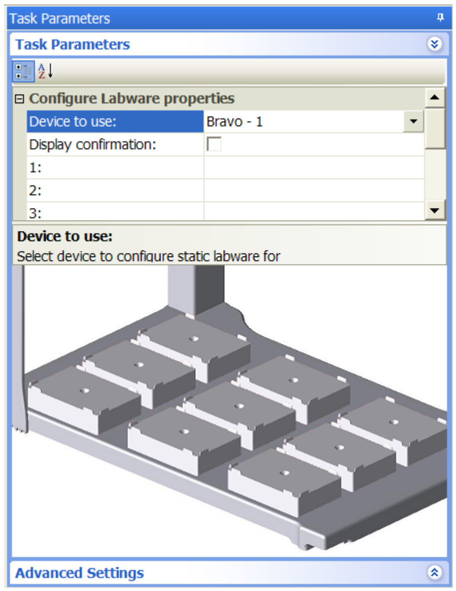
Note: You can also configure static labware in the device subprocess task in the main protocol. Configure labware in the device subprocess if you want to override the labware configuration in the startup protocol. Configure labware in a Startup Protocol if the labware configuration will be used in all the Main Protocol subprocesses.

In general, you configure static labware before the first task in a protocol. If you have multiple processes in the protocol, configure the labware once before the first task of the first process.

Task is available for...	Task is available in...
Any device	Startup Protocol

Task parameters

After adding the Configure Labware task in the startup protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Device to use	The device on which the labware is placed. Select from the list of available devices.
Display confirmation	The option to display a message to check the physical placement of the labware against the assignment in the software.
Device location	The list of locations on the device. For example, if you select Bravo Pipettor as the device to use, deck locations 1 through 9 appear in the list. Click the device location, and then select the labware type.

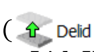
Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide

For information about...	See...
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Subprocess task	"SubProcess (Bravo, Vertical Pipetting Station)" on page 388
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
BioCel I/O Interface Diagnostics	*BioCel System User Guide*

Delid

Description

The Delid task (

) removes lids from microplates using a device such as a Bravo Pipettor, Lid Hotel Station, or Vacuum Delid Station (a vacuum-based delidding device). You can use the Delid task to remove and discard a lid, or you can pair it with the Relid task to retain the removed lid and later place the lid back on the microplate.

Note: You can only delid process plates and configured labware. You cannot delid static labware.

Task is available for...	Task is available in...
BenchCel Workstation	Startup Protocol Main Protocol Cleanup Protocol
Bravo Platform	Startup Protocol Main Protocol, Bravo Subprocess Cleanup Protocol
Lid Hotel Station	Startup Protocol Main Protocol Cleanup Protocol
Vacuum Delid Station	Startup Protocol Main Protocol Cleanup Protocol

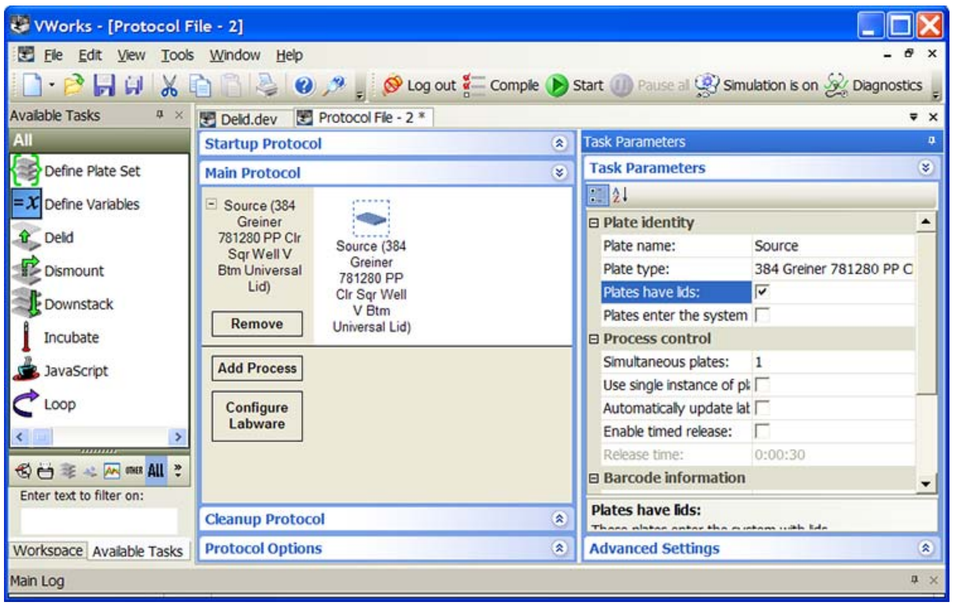
Requirements

Labware entry

In the Labware Editor, make sure the labware you want to delid has lid specifications in the **Plate Properties** tab. For detailed instructions, see the *VWorks Automation Control Setup Guide*.

Plate parameters

When you set parameters for the process plate, make sure you select the **Plates have lids** option. For detailed instructions, see "Setting plate parameters" on page 46.



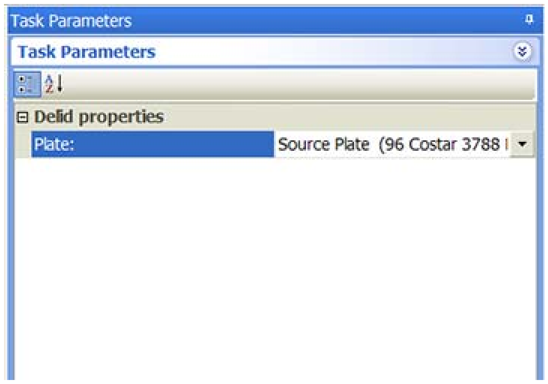
Bravo Platform

To use the Delid task, you must first configure the Lid Hotel in Bravo Diagnostics. For detailed instructions, see the *Bravo Automated Liquid Handling Platform User Guide*.

Task Parameters

Bravo Platform

To delid microplates on the Bravo Platform, you add the Delid task within the Bravo Subprocess. After adding the Delid task at the desired point in the subprocess, select the microplate you want to delid in the **Task Parameters** area, as the following example shows.



Parameter	Description
Plate	The microplate that will be delidded.

Lid Hotel Station or Vacuum Delid Station

The Delid task does not have task parameters when using the Lid Hotel Station or the Vacuum Delid Station device. However, you must select the device you want to use.

Device selection

You must select a device for Delid tasks that use the Lid Hotel Station or the Vacuum Delid Station Station. You do not need to select a device when the Delid task is within a Bravo Subprocess, because the software assumes you want to use the Bravo Platform.

If you have multiple devices of the same type, you can prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.

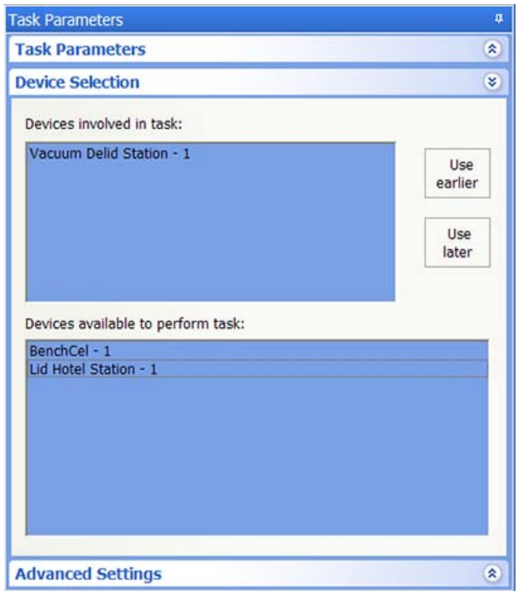
IMPORTANT The multiple devices must have identical setup and configuration.

After adding the Delid task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1 Double-click the desired device in the **Devices available to perform task** area to move it to the **Devices involved in task** area. If you have multiple devices of the same type, you can move them to the **Devices involved in task** area.

2 If you have multiple devices in the **Devices involved in task** area, select a device, and then click **Use earlier** or **Use later** to prioritize it.



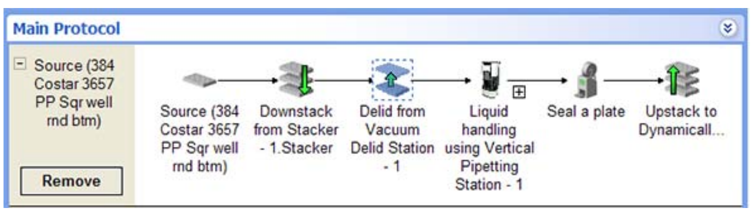
Example: Delid microplates

Goal

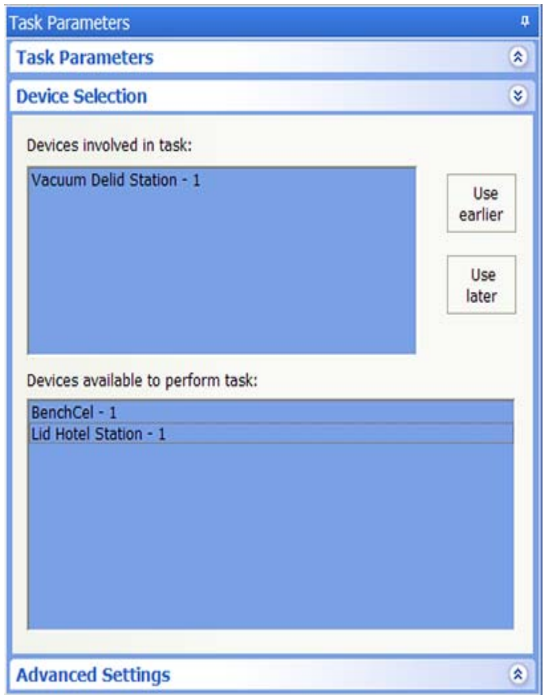
After a microplate is downstacked from the Labware Stacker device, remove the lid for liquid-handling tasks on the Vertical Pipetting Station. A Vacuum Delid Station is available to remove the lid and drop it into the waste bin.

Implementation

Add a Delid task after the Downstack task. The Delid task removes the lid from the microplate and drops it into the waste bin.



The device selection for the Delid task is:



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Relid task	"Relid" on page 333
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Dismount

IMPORTANT The Dismount task is always paired with the Mount task. Before adding the Dismount task, review the description for the Mount task. See “Mount” on page 307.

The Dismount task () removes labware, such as a filter microplate, from the top of an elution microplate or reservoir that is resting on a platepad. Always paired with a Mount task, the Dismount task uses information that is specified in the Mount task.

For example, you can use the Mount task to place a filter microplate on a reservoir that catches the elution. After the centrifuge process, you can use the Dismount task to remove the filter microplate from the reservoir and discard it.

Task is available for...	Task is available in...
Platepad	Main Protocol

Task parameters

The Dismount task does not have task parameters.

Example: Dismount a filter microplate from a collection microplate

Goal

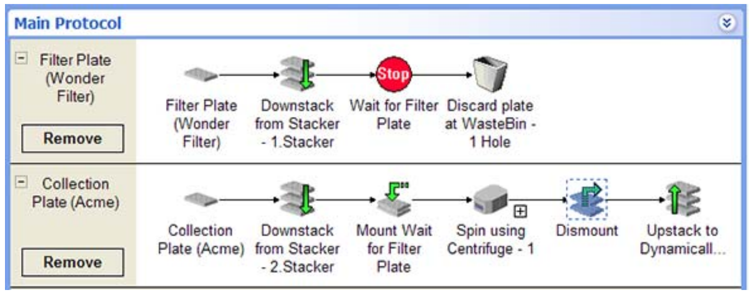
Downstack a collection microplate from a Labware Stacker device and place it on a platepad. Downstack a filter microplate from another Labware Stacker device and place it on top of the collection microplate. Centrifuge the assembly (filter microplate with collection microplate). Remove the filter microplate from the collection microplate and discard it. Upstack the collection microplate containing the elution for future processing.

Implementation

Add one process for the collection microplate and another process for the filter microplate. See “Example: Mount a filter microplate on a reservoir that collects elution” on page 308 for details about the Mount and Wait For tasks.

In the filter microplate process, add a Waste task after the Wait For task. The Waste task discards the filter microplate after the centrifuge is finished.

In the collection microplate process, add the following tasks after the Mount task: Centrifuge, Dismount (to remove the filter microplate from the collection microplate), and Upstack (to store the collection microplate containing the elution).



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Mount task	"Mount" on page 307
Downstack task	"Downstack" on page 354
Upstack task	"Upstack" on page 382
Waste task	"Waste" on page 346
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Incubate

Description

The Incubate task (Incubate) moves labware to a location, leaves it there for a specified time period, and then moves it from the location. The number of labware that can be incubated simultaneously is limited by the number of locations that are available for holding labware. Typically, you incubate labware on platepads or in Plate Hotels.

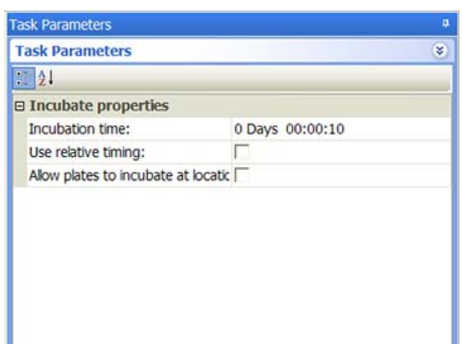
Note:

- To incubate a labware in a storage device such as a Plate Hub Carousel or the LiConic StoreX incubator, use the Storage Incubate task.
- To incubate labware on the Bravo Platform, use the Reserve Location task.

Task is available for...	Task is available in...
Platepad	Main Protocol
Plate Hotel	Main Protocol
Vertical Pipetting Station	Main Protocol

Task parameters

After adding the Incubate task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Incubation time	The length of time to incubate the labware. *Note:* The actual incubation period might be longer than the incubation time you specify. The actual incubation period is affected by the scheduling and operating speed of the robot.

Parameter	Description
Use relative timing	The option to start timing the incubation after the previous labware has finished incubating. *Note:* The option can result in a longer incubation period than specified. Select this option to prevent bottlenecking at a downstream device. Alternatively, use the **Release time** parameter when setting process plate properties.
Allow plates to incubate at locations with BCRs\	The option to permit a labware to be placed at a location with a barcode reader. The barcode reader is not used during the Incubate task.

Device selection

You must select a device for Incubate tasks. If you have multiple devices of the same type, you can:

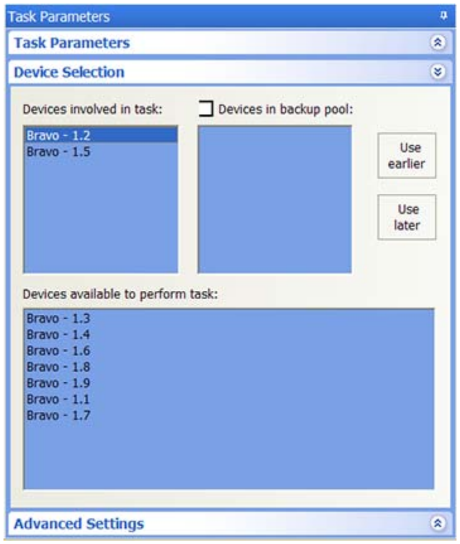
- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the list. However, if all of the devices in the list are in an error state, the software will automatically use the device in the backup pool.

IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Incubate task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click the desired device in the **Devices available to perform task** area to move it to the **Devices involved in task** area. If you have multiple devices of the same type, you can move them to the **Devices involved in task** area.
2. If you have multiple devices in the **Devices involved in task** area, select a device, and then click **Use earlier** or **Use later** to prioritize it.
3. *Optional.* Select backup devices to use in case all of the devices in the **Devices involved in task** area encounter an error.
   a. Select **Devices in backup pool.**
   b. Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.
   c. If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



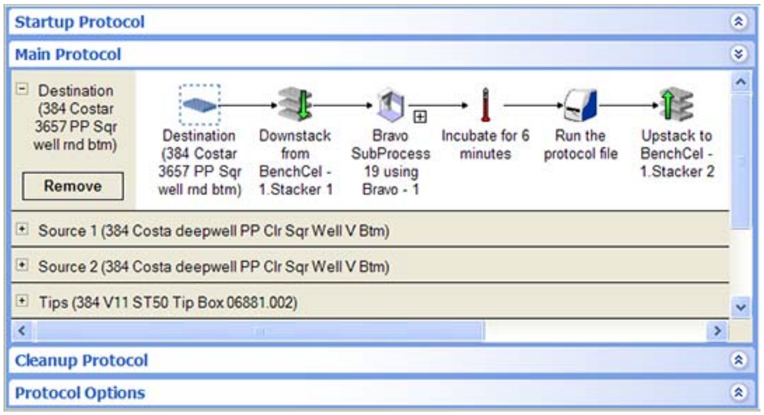
Example: Incubate a microplate after mixing two reagents

Goal

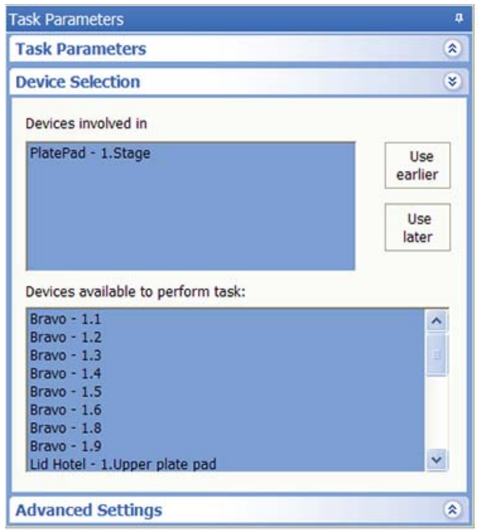
Aspirate contents from two source microplates, dispense into a destination microplate. Incubate the mixture for at least 6 minutes, and then read the microplate.

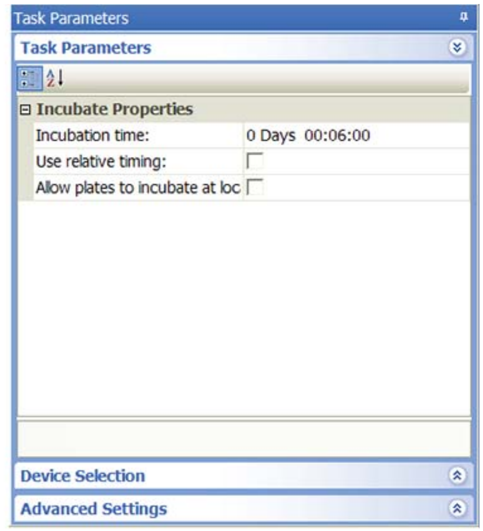
Implementation

In the protocol, add an Incubate task after the liquid-handling tasks as shown in the following example.



For the Incubate task, select platepad-1 for the incubation location. Set Incubation time to 6 minutes in the Task Parameters area as shown.





Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Process plate properties	"Setting plate parameters" on page 46
Storage Incubate task	"Storage Incubate" on page 373
Reserve Location task	"Reserve Location" on page 337
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate-handling tasks" on page 277
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Mount

Description

The Mount task (Mount) places a labware, such as a filter microplate, on top of a collection microplate or reservoir that is resting on a platepad. The Mount task works with the Wait For task and is paired with the Dismount task.

For example, you can use the Mount task to place a filter microplate on a reservoir that catches the elution. After the centrifuge process, you can use the Dismount task to remove the filter microplate from the reservoir and discard it.

The Mount task becomes available when you add a Platepad device in the device file. The task can be used only in the Main Protocol.

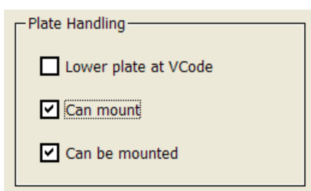
Task is available for...	Task is available in...
Platepad	Main Protocol

Requirements

Labware definition

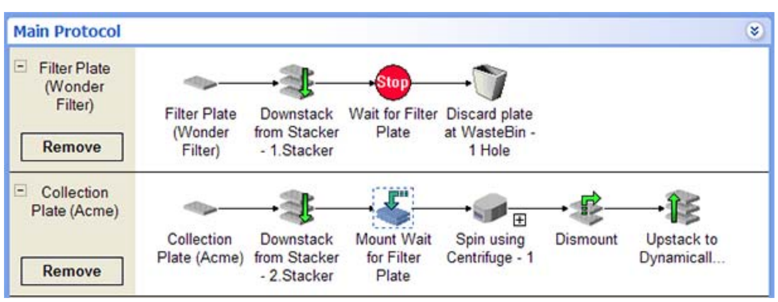
The labware involved in the Mount task must have the following Plate Handling properties in the Labware Editor:

- *Can be mounted*. Select the **Can be mounted** property to define microplates or reservoirs that will be mounted (will be on the bottom).
- *Can mount*. Select the **Can mount** property to define microplates that will mount on top of the elution microplate or reservoir.



Process setup

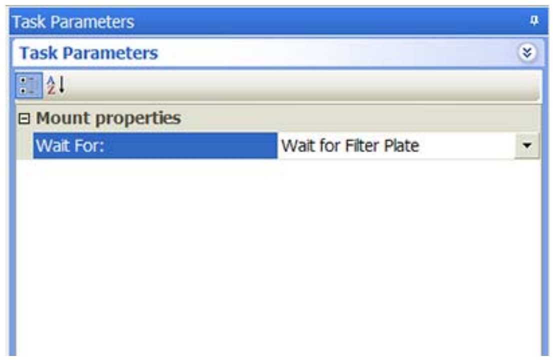
Each labware involved in the Mount task should be set up as independent processes in the protocol. For example, if you want to place a filter microplate on top of a reservoir or collection microplate that catches the elution, you should create two processes as shown in the following example:



Notice that a Wait For task is added to the Filter Plate process at the point where the mount task would occur. A Mount task is added to the Collection Plate process at the desired point. See "Example: Mount a filter microplate on a reservoir that collects elution" on page 308 for a detailed description.

Task parameters

After adding the Mount task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Wait for	The name of the Wait For task that you want to pair with this Mount task. Select from the list of Wait For tasks. The list is based on the Wait For tasks that have been added in the protocol.

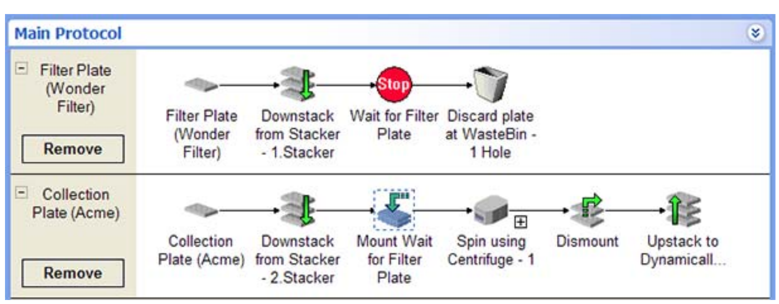
Example: Mount a filter microplate on a reservoir that collects elution

Goal

Downstack a reservoir (Collection Plate) from a Labware Stacker device and place it on a platepad. Downstack a filter microplate (Filter Plate) from another Labware Stacker device and place it on top of Collection Plate.

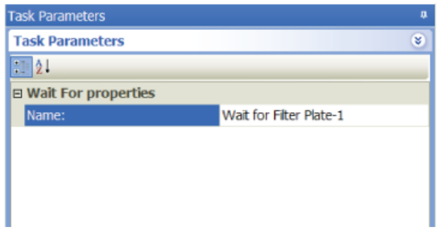
Implementation

Create one process for the Collection Plate and another process for Filter Plate. In the Filter Plate process, add a Wait For task after the Downstack task. In the Collection Plate process, add a Mount task after the Downstack task.



The task parameters for the Wait For task and the Mount task are:

Wait For



Mount

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Dismount task	"Dismount" on page 301
Wait For task	"Wait For" on page 603
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Move to Location (Bravo)

Description

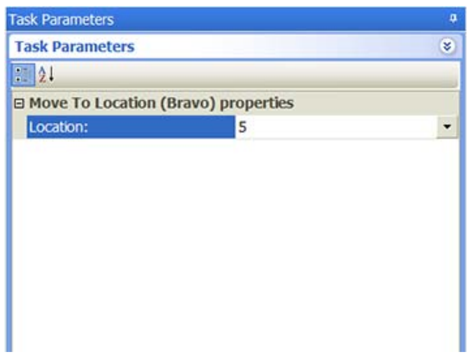
The Move to Location task (Move To Location (Bravo)) moves the pipette head to a safe distance above a specified deck location. The safe distance is set in the Bravo Pipettor profile.

Typically, this task is used in Startup and Cleanup Protocols to move the pipette head out of the way. For example, you can move the pipette head to location 5 before placing a microplate at location 1.

Task is available for...	Task is available in...
Bravo Platform	Startup Protocol Main Protocol, Bravo Subprocess Cleanup Protocol

Task parameters

After adding the Move To Location task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location	The deck location the pipette head will move to. The height above the deck location is determined by the *z*-axis safe position setting in the profile.

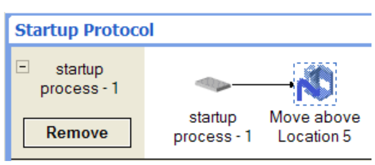
Example: Move a Bravo pipette head before the main protocol starts

Goal

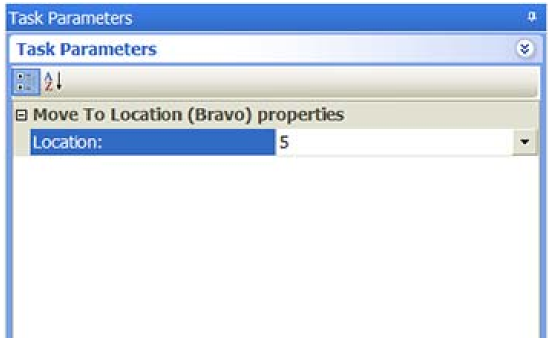
Before the main protocol starts, move the Bravo pipette head to location 5.

Implementation

In the startup protocol, add a Move to Location task. Specify that the pipette head should move to location 5.



The task parameter for the Move to Location task is:



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Place Plate

Description

The Place Plate task (Place Plate) moves labware to a specified location. If the location has a barcode reader installed, the Place Plate task moves the labware to the location and reads the barcode.

Task is available for...	Task is available in...
Any device	Main Protocol

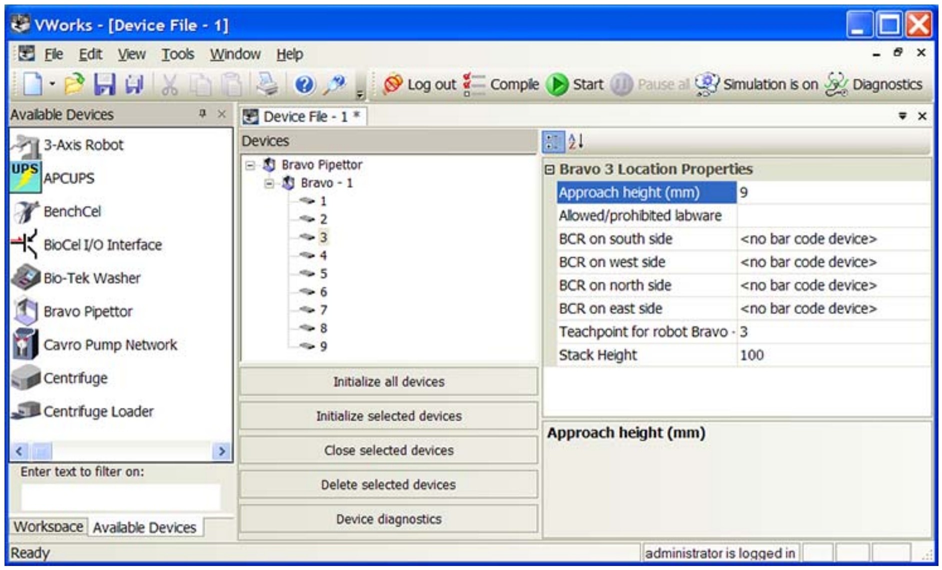
Requirements

To read barcodes at the specified location such as a platepad, you must:

- Set up the barcode reader in the device file.
- Add the device that has the barcode reader, and then specify the location of the barcode reader. For example, if the barcode reader is installed on a platepad, specify the side on which the barcode reader is installed.

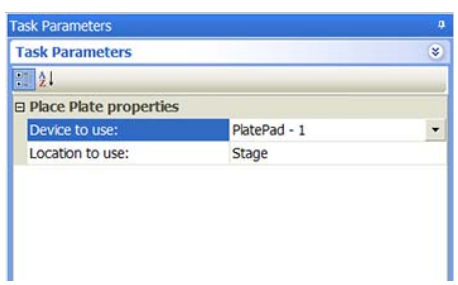
If you are placing a labware on a device or accessory, make sure:

- The device or accessory is configured.
- The Device Properties approach height clears the top of tall accessories.



Task parameters

After adding the Place Plate task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Device to use	The device where you want to place the labware.
Location to use	The location on the device to place the labware.

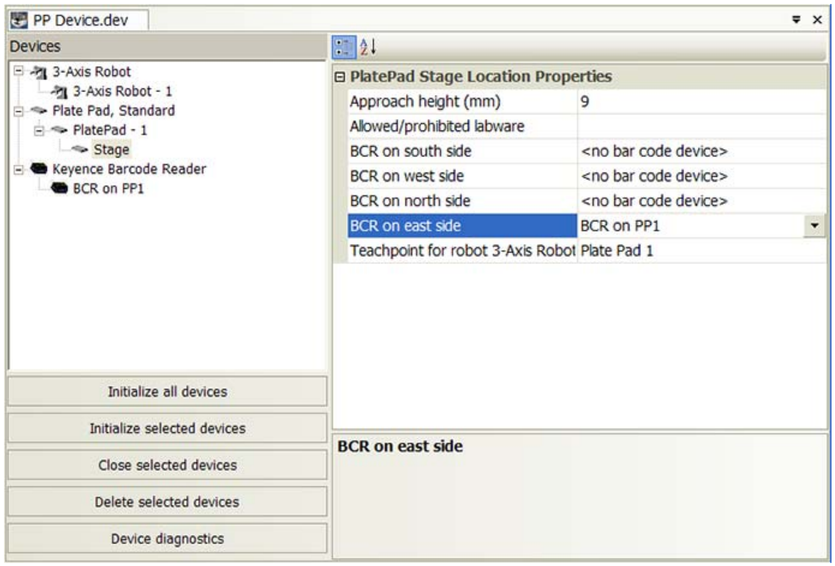
Example: Place a microplate at a platepad and read its barcode label

Goal

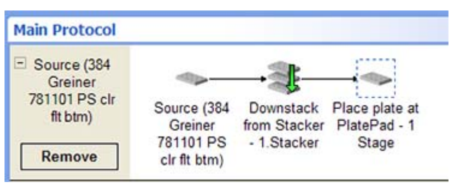
Place the Source Plate at the specified platepad and read the barcode label. The barcode reader is installed on the east side of the platepad.

Implementation

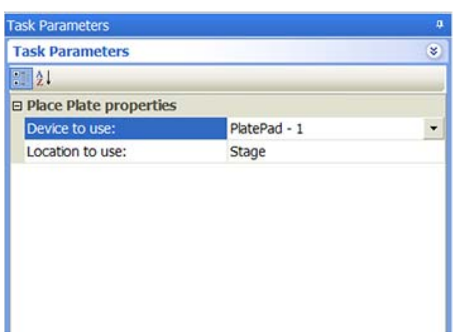
When adding the platepad in the device file, specify the barcode reader on the east side of the platepad.



Add a Place Plate task at the desired point in the protocol.



The task parameters for the Place Plate task are:



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Pierce Plate (Seal Piercer)

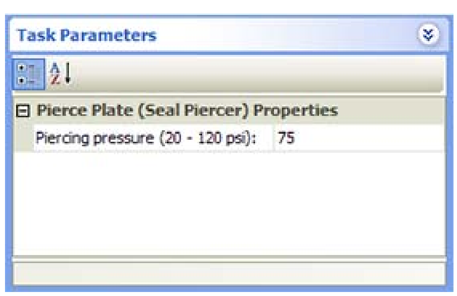
Description

The Pierce Plate (Seal Piercer) task (Pierce Plate (Seal Piercer)) pierces a microplate seal using a Seal Piercer.

Task is available for...	Task is available in...
Seal Piercer	Main Protocol

Task parameters

After adding the Pierce Plate (Seal Piercer) task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Piercing pressure	The air pressure, in psi, used during the piercing process.

Device selection

You must select a device for Pierce Plate (Seal Piercer) tasks. If you have multiple devices of the same type, you can:

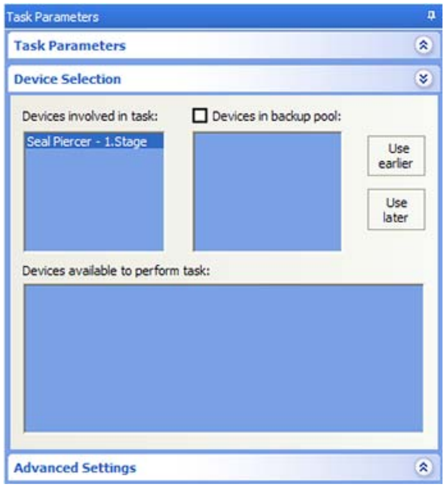
- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the list. However, if of all the devices in the list are in an error state, the software will automatically use the device in the backup pool.

IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Pierce Plate (Seal Piercer) task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click a device in the **Devices available to perform task** area to add it to the **Devices involved in task** area.
2. To prioritize its use, select the device in the **Devices involved in task** area, and then click **User earlier** or **Use later.** The devices that are higher in the list are favored by the software scheduler during the protocol run.
3. *Optional.* Select backup devices to use in case all of the devices in the **Devices involved in task** area encounter an error.
   a. Select **Devices in backup pool.**
   b. Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.
   c. If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



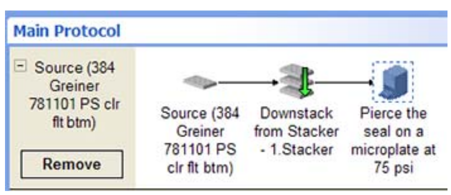
Example: Pierce a microplate seal

Goal

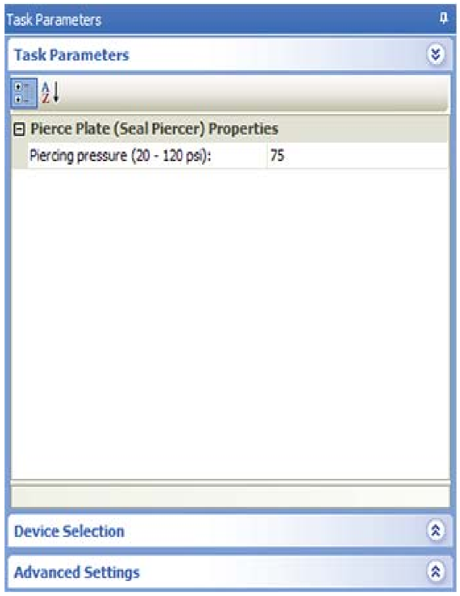
Pierce the Source Plate after it is downstacked from the Labware Stacker.

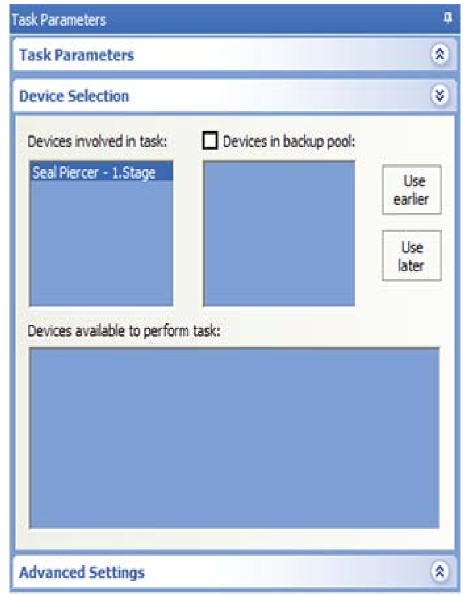
Implementation

Add a Pierce Plate (Seal Piercer) task after the Downstack task. Set the piercing pressure at 75 psi.



The task parameter and device selection for the Pierce Plate (Seal Piercer) task are:





Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Print and Apply

Description

The Print and Apply task (Print and Apply) allows you to:

- Specify the contents to be printed on a microplate label.
- Print the specified contents on a label.
- Apply the label to a labware.

Task is available for...	Task is available in...
Microplate Labeler	Main Protocol

Requirements

Before adding the Print and Apply task, make sure you complete the following:

Step	For this task...	See...
1	Set up the Microplate Labeler.	"Setting up the Microplate Labeler" on page 318
2	Specify the label format.	"Specifying the label format" on page 319
3	*Optional*. Create barcode data files.	"Creating barcode data files" on page 319
4	Select the option to lower the plate stage when labeling.	"Lowering the plate stage" on page 320
5	*Optional*. Specify the label location on labware for barcode reading.	"Specifying label location for barcode reading" on page 321

Setting up the Microplate Labeler

For instructions on how to set up the Microplate Labeler (installing the device, adding the device in the VWorks software, creating a profile for the device in diagnostics, and so on), see the *Microplate Labeler User Guide*.

Specifying the label format

In Microplate Labeler Diagnostics, you specify the label format (label design). The format you specify includes the following:

- Number of fields to be printed (up to six are permitted)
- Type of content to be populated in a field (human-readable text or barcode)
- Font type of the human-readable text, or symbology of the barcode
- Position (coordinates) of the information on the label
- Field attributes, such as field size
- Position (coordinates) of the information on the label

For instructions, see the *Microplate Labeler User Guide*.

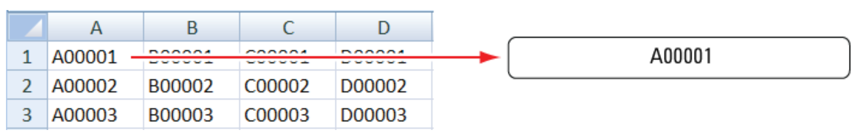
After you specify the label format, you can use the Print and Apply task to specify the content that will populate each field. When you run the protocol that contains the Print and Apply task, the software substitutes the data (label content) for the text and barcode fields in the label format, and then prints the label.

Creating barcode data files

The Print and Apply task provides many methods to create barcode label contents. Two of the methods require the use of barcode data files:

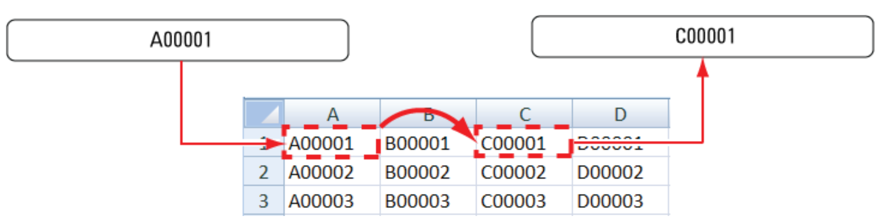
- The software reads data from the data file and prints that data on the label.

  Note: You specify the row and column number of the starting value when you set the Print and Apply task parameter. For instructions, see "Setting the task parameters" on page 323.

- The software reads the barcode on one side of a labware, looks up the barcode (key) in the data file, locates the data found in another column (value), and prints and applies the new label to the same side. The primary use of this function is for label replacement.

  Note: You specify the key and value columns when you set the Print and Apply task parameter. For instructions, see "Setting the task parameters" on page 323.

Barcode data files must meet the following requirements:

- Be a comma-delimited text file with the .csv file name extension
- *Optional.* Contain a header, which can be in any format.
- Contain at least one column. For example, the file can contain four columns, each representing a side of the microplate.

The following example shows a data file displayed in Excel. The file contains four columns, each representing a side of the microplate. In addition, the file contains a header in row 1.

	A	B	C	D
1	North	South	East	West
2	A00001	B00001	C00001	D00001
3	A00002	B00002	C00002	D00002
4	A00003	B00003	C00003	D00003
5				

The data file can be stored anywhere on the computer that runs the VWorks software.
If you plan to use data files, make sure the files meet the requirements described in this section. In addition, make sure you select the correct data file when setting the Print and Apply task parameters. See "Setting the task parameters" on page 323.

Lowering the plate stage

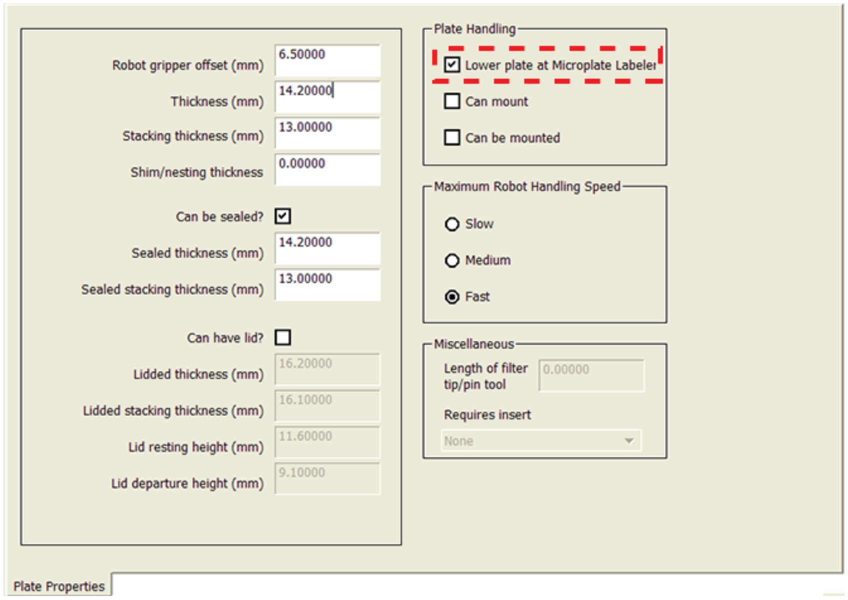
The Microplate Labeler plate stage has two vertical positions, top and bottom, which are used to accommodate two different sized microplates. By default, the system uses the top position during a protocol run. If the labware has a tall skirt (**1**) or has a raised surface on the sides (**2**), you can specify that the Microplate Labeler use the bottom position (or lower the stage) during the run so that the label can be applied above the skirt or raised surface.

IMPORTANT Always perform a dry run to verify that the positions are correct for the labware you are processing.

Note: The two vertical plate stage positions are set mechanically. To adjust the positions, see the *Microplate Labeler User Guide* for instructions.

To lower the Microplate Labeler plate stage during a run:

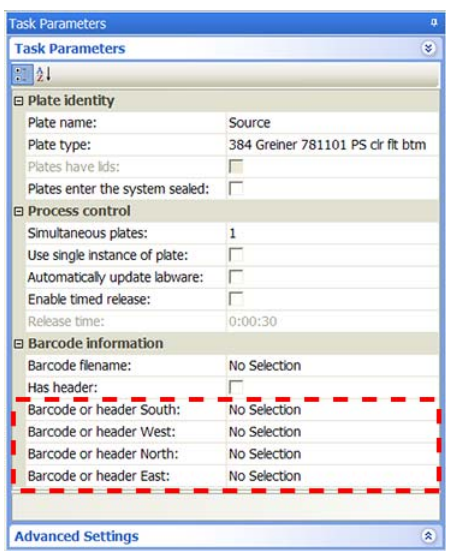
1 Open the **Labware Editor,** and then click the **Plate Properties** tab.
2 Select the labware from the list on the left of the dialog box.
3 In the **Plate Handling** area, select **Lower plate at Microplate Labeler.**



Specifying label location for barcode reading

If you plan to use a barcode reader at the Microplate Labeler, you must specify the location of the barcode label on the labware so that the system will know which side to scan during the run.

You specify the barcode label location when you set the plate parameters. For more information, see "Setting plate parameters" on page 46.



If you plan to scan a new label after it is applied, you must also select the Verify barcodes after Print and Apply option in the Microplate Labeler Diagnostics Profiles tab. For more information, see the *Microplate Labeler User Guide*.

Selecting devices for the task

After adding the Print and Apply task, you must select a device for the task before you can set the task parameters. If you have multiple devices of the same type, you can:

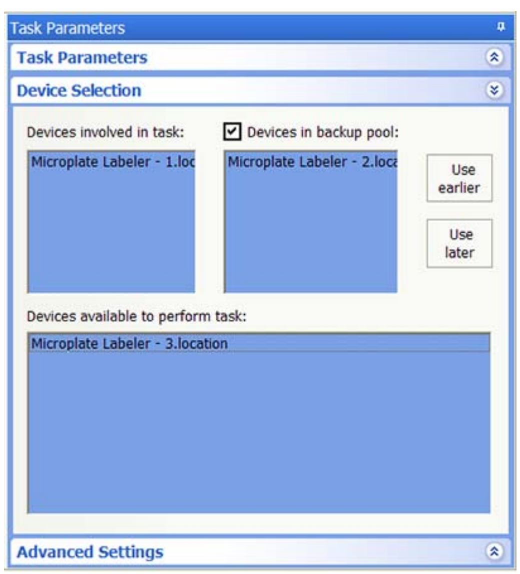
- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the primary list. However, if all of the devices in the primary list are in an error state, the software will automatically use the device in the backup pool.

IMPORTANT The multiple devices must be configured and set up identically. All formats must be identical across the multiple devices. This includes the format names, field names, and definitions.

After adding the Print and Apply task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click the desired device in the **Devices available to perform task** area to move it to the **Devices involved in task** area. If you have multiple devices of the same type, you can move them to the **Devices involved in task** area.
2. If you have multiple devices in the **Devices involved in task** area, select a device, and then click **Use earlier** or **Use later** to prioritize it.
3. *Optional.* Select backup devices to use in case all of the devices in the **Devices involved in task** area encounter an error.
   a. Select **Devices in backup pool.**
   b. Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.
   c. If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



Setting the task parameters

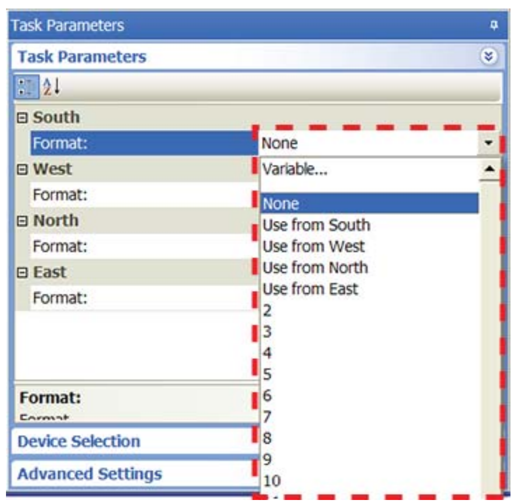
IMPORTANT Make sure the label formats are uploaded to the printer. Be sure to initialize the Microplate Labeler device before you set the task parameters.

After selecting the device to use for the Print and Apply task, you can set the parameters in the **Task Parameters** area. The area lists the four sides of a microplate (south, west, north, and east). For each side, you can select a label format and specify the data that will substitute for the text and barcode fields in the label format.

CAUTION Format selection and field information are saved with the protocol. If the formats on the printer are changed, initializing the device will overwrite the information in the protocol. For example, suppose you created a protocol and selected a format called MyFormat. Later, MyFormat was deleted from the printer. The next time you initialize the device and open the protocol, MyFormat will not appear in the protocol.

To set the task parameters:

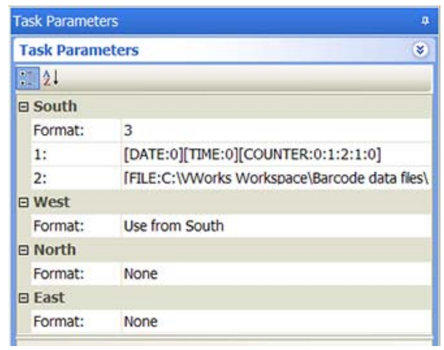
1 For the side on which the barcode label will be applied, select the barcode format in the **Format** list:



Format selection	Description
None	Indicates no label will be applied.

<table>
<tr><th>Format selection</th><th>Description</th></tr>
<tr><td>Use from <side></td><td>Uses the format from another side of the microplate to print a new label and apply it to the selected side.

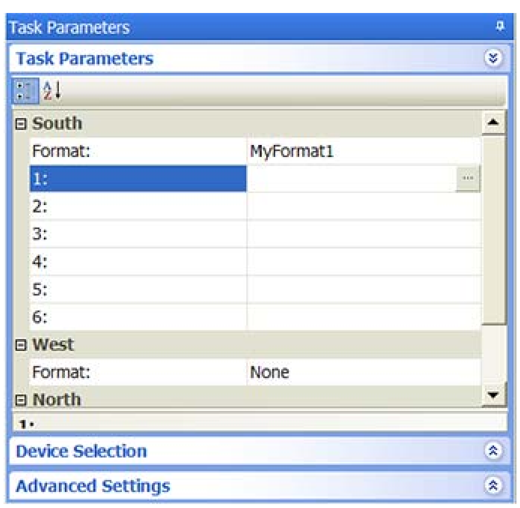
For example, suppose you have already selected a format and specified the label contents for the South side. You want to use the same format and contents on the West side without having to reselect and respecify the same information. Then in the West Format list, select Use from South.


</td></tr>
<tr><td>Format name or number</td><td>Uses a format that was set up in Microplate Labeler Diagnostics.

Note: If you do not see a list of formats, make sure the label formats are not empty (formats must contain at least one field), the formats are uploaded to the printer, and the Microplate Labeler device is initialized.</td></tr>
</table>

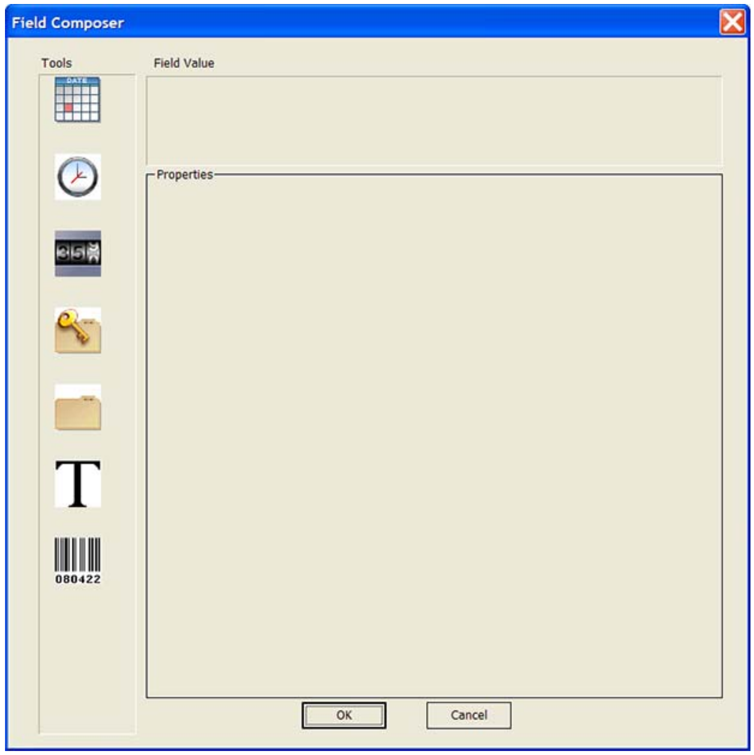
As soon as you select a format, fields appear in the Task Parameters area. You can specify the information you want to print in these fields.

Note: The number of fields that appear depends on the format you select.



2 Click a field, and then click the [...] button that appears. The Field Composer dialog box opens.

The Field Composer allows you to specify the information to print on the barcode label. For example, you can print the current date and time.



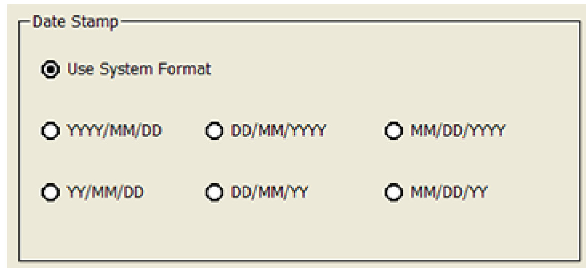
3 In the **Tools** area, double-click one or more of the following icons to specify the information to be printed on the barcode label. The selected icon appears in the **Field Value** area.

IMPORTANT For field limitations, such as the maximum number of characters permitted or symbology-dependent limitations, check the format you set up in Microplate Labeler Diagnostics. See also the *Microplate Labeler User Guide*.

<table>
<tr><th>Icon</th><th>Description</th></tr>
<tr><td></td><td>
Prints the current date.

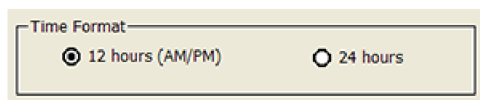
Click the icon in the **Field Value** area. In the **Properties** area, select the desired date format. YYYY is the year, MM is the month, and DD is the day.

Note: The Use System Format option uses the local computer’s date format.



</td></tr>
<tr><td></td><td>
Prints the current time.

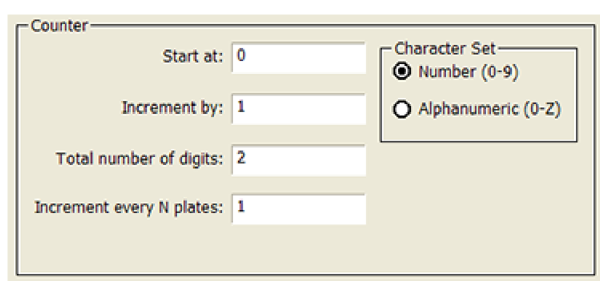
Click the icon in the **Field Value** area. In the **Properties** area, select the desired data format: **12 hours (AM/PM)** or **24 hours.**



</td></tr>
<tr><td></td><td>
Prints a numeric or alphanumeric value that can be incremented.

Click the icon in the **Field Value** area. Set the following in the **Properties** area:

- *Character Set*. The option to use either numeric or alphanumeric characters.
- *Start at*. The starting value.
- *Increment by*. The amount by which the value increments.
- *Total number of digits*. The total number of digits or characters, including leading 0s.
- *Increment every N plates*. The increment value. For example, 1 increments the value every microplate.

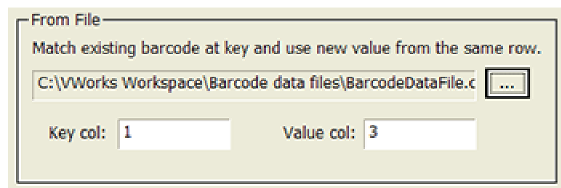


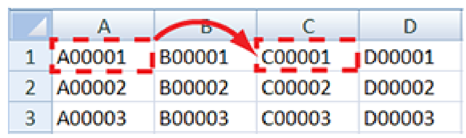
</td></tr>
</table>

<table>
<tr><th>Icon</th><th>Description</th></tr>
<tr><td></td><td>Looks up a value (key) in the specified barcode data file and prints the corresponding value found in the file.

Click the icon in the Field Value area. In the Properties area, click the ... button to select the barcode data file. Specify the column number that contains the lookup value (key) and the column number that contains the values you want to print.

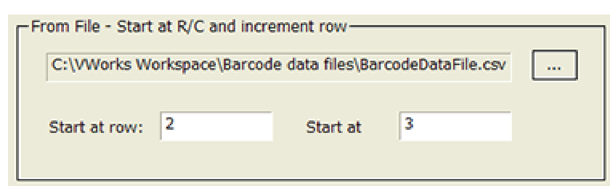
For example, Key col is set at 1, and Value col is set at 3. During the run, an A00001 barcode is scanned. The software looks up A00001 in the specified file, and prints the barcode, C00001, found in the third column.

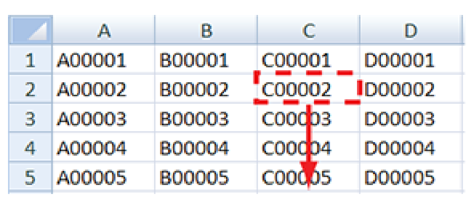


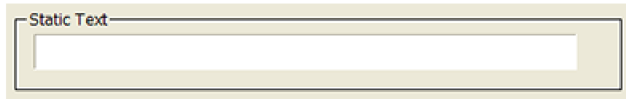

</td></tr>
<tr><td></td><td>Prints the values in the specified barcode input file.

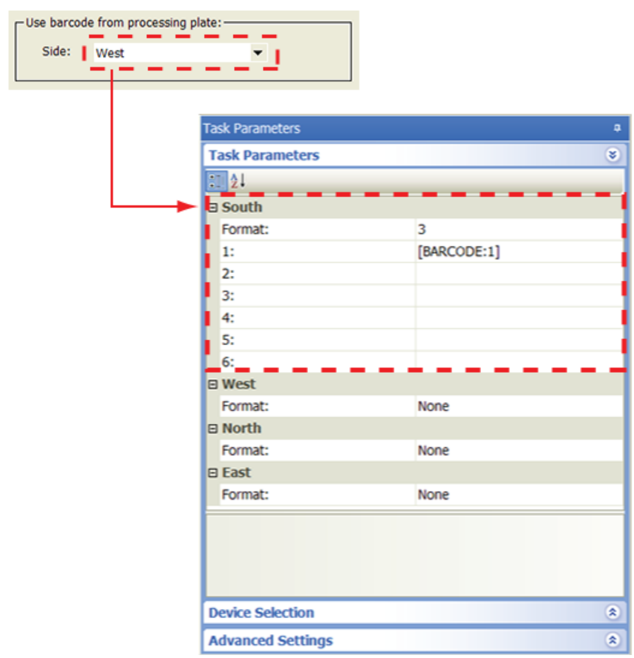
Click the icon in the Field Value area. In the Properties area, click the ... button to select the input file. Specify the row and column number of the starting value. During the run, the software automatically increments to the next row to print the next value.

For example, Start at row is set at 2, Start at col is set at 3. During the run, the software starts with the value in row 2 column 3 (C00002). Then, the software moves to row 3 column 3 (C00003), row 4 column 3 (C00004), row 5 column 3 (C00005), and so on.




</td></tr>
</table>

<table>
<tr><th>Icon</th><th>Description</th></tr>
<tr><td>T</td><td>
Prints the text you specify.
Click the icon in the Field Value area. In the Properties area, type the text you want to print on every microplate.


</td></tr>
<tr><td></td><td>
Copies the barcode data from an existing label on the selected side to print a new label.
Note: You can only copy barcode data from another label. You cannot copy human-readable text.
Click the icon in the Field Value area. In the Properties area, select the side (barcode) to be copied.
For example, an incoming microplate already has a label on the west side. You want to copy the barcode data from the west-side label to print a new label, and apply the new label to the south side. In the Task Parameters area, you specify that the south side of the labware will be labeled. In the Field Composer dialog box, you specify Use barcode from processing plate, and select West.


</td></tr>
</table>

4 When you are finished, click **OK.** The information you specified appears in the Task Parameters area.

5 Repeat the procedure if you want to print labels on other sides of the microplate.

Example 1: Copy label contents from one side of the microplate and print and apply to a new side

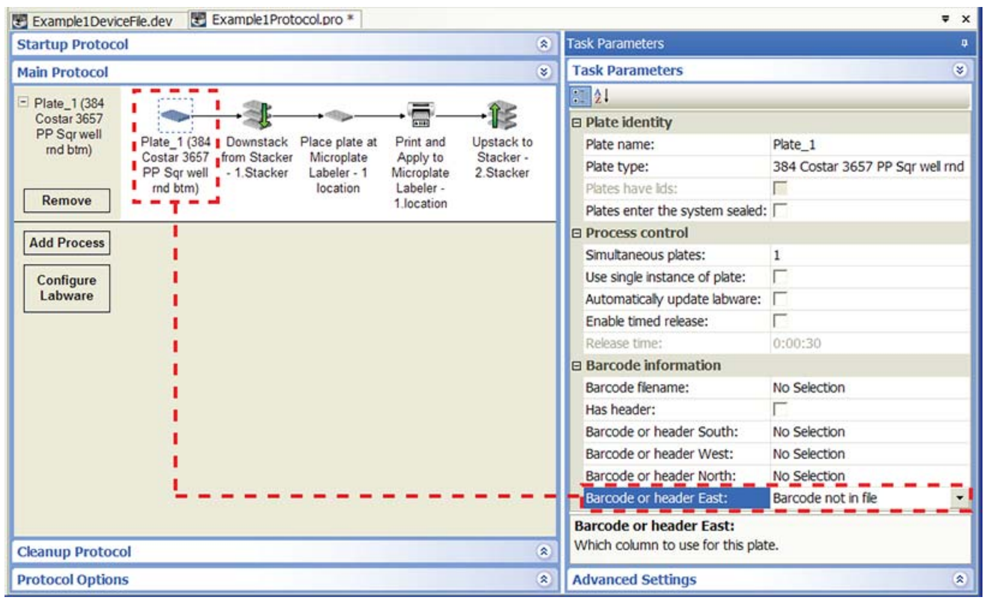
Goal

For each incoming microplate, scan the label on the east side of the microplate, print the same contents on a new label, and apply the new label to the north side of the microplate.

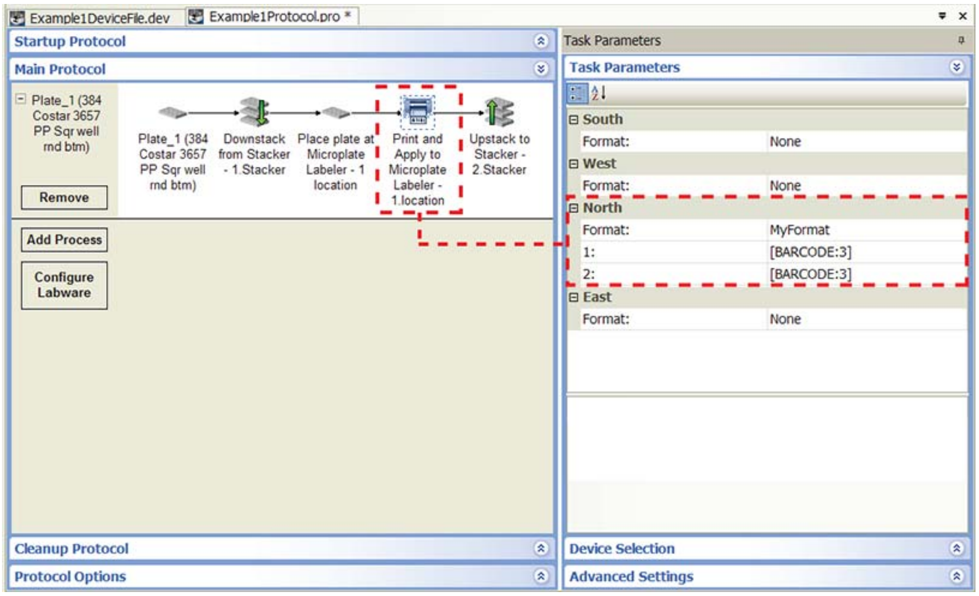
Implementation

Note: This example assumes that the Microplate Labeler is set up correctly and the format, MyFormat, is already defined and loaded to the printer. MyFormat contains two fields. Field 1 is a human-readable text field. Field 2 is a barcode field. The device file and protocol are correctly created.

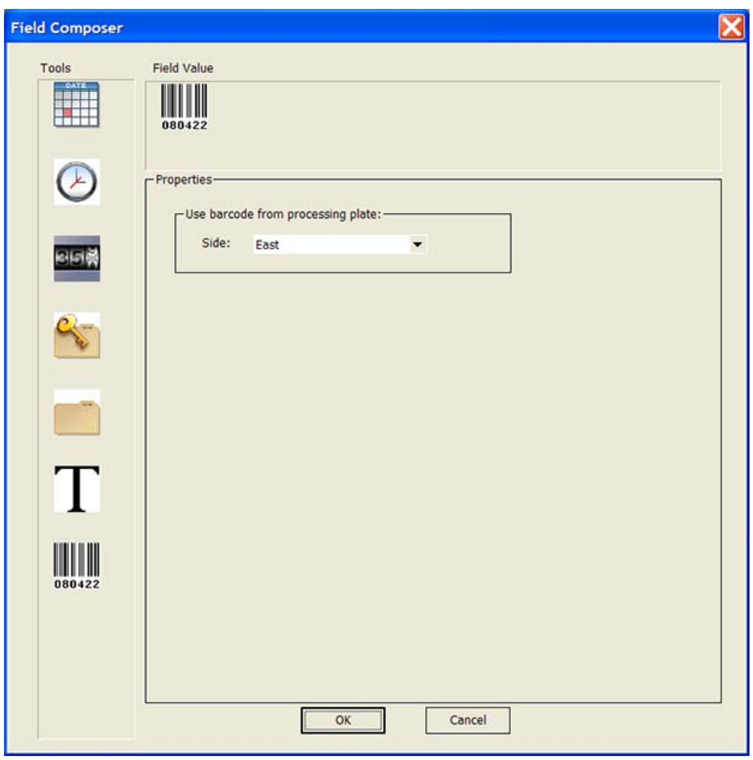
When setting the plate parameters, select **Barcode or header East** in the **Barcode information** area so the system knows to scan the east side of the incoming microplates. No barcode verification file will be used, so select **Barcode not in file.**



When setting the Print and Apply task parameter, select MyFormat for the side you want to label: **North.**



For each of the two format fields (1 and 2), open the Field Composer dialog box and double-click the barcode icon in the **Tools** area to add it to the Field Value area. Because you want to copy the barcode information from the east side of the microplate, select **East** from the **Side** list.



Example 2: Use a barcode data file to print and apply labels

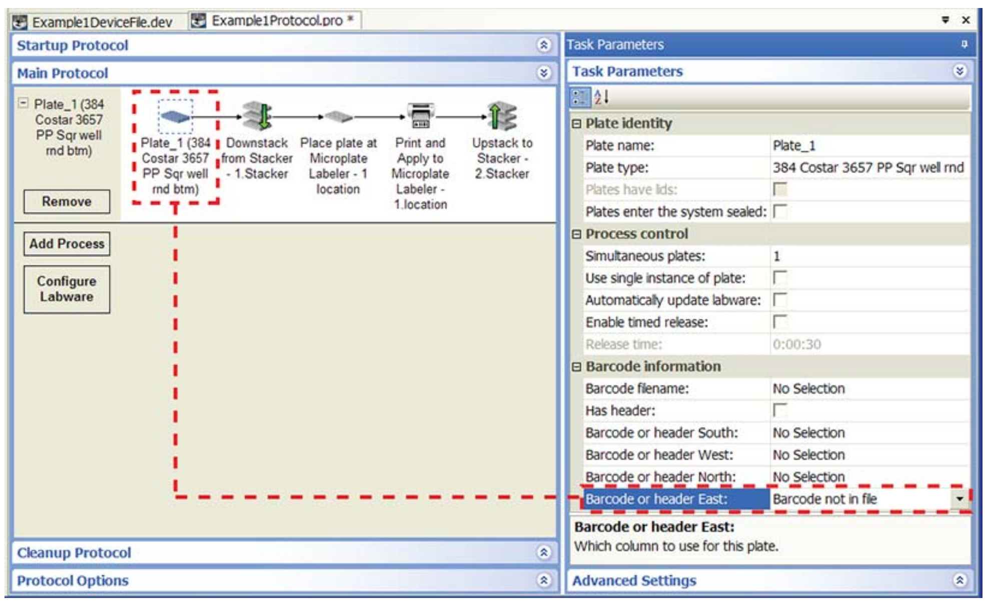
Goal

For each incoming microplate, scan the label on the east side of the microplate, locate the information (key) in the first column in the data file, print the corresponding information (value) from the third column in the file on a new label, and apply the new label to the same (east) side of the microplate.

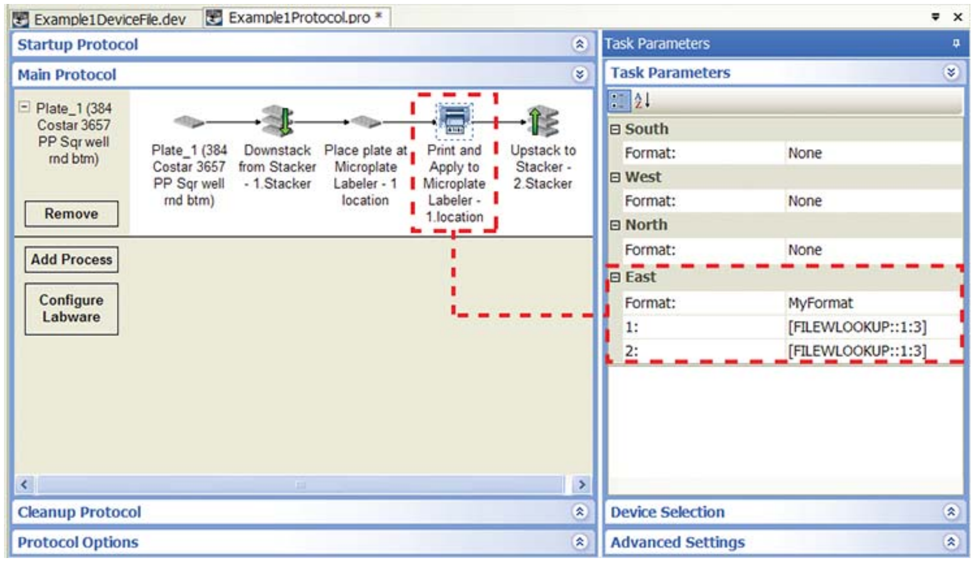
Implementation

Note: This example assumes that the Microplate Labeler is set up correctly and the format, MyFormat, is already defined and loaded to the printer. MyFormat contains two fields. Field 1 is a human-readable text field. Field 2 is a barcode field. The barcode data file is stored in C:\VWorks Workspace\Barcode data files. The device file and protocol are correctly created.

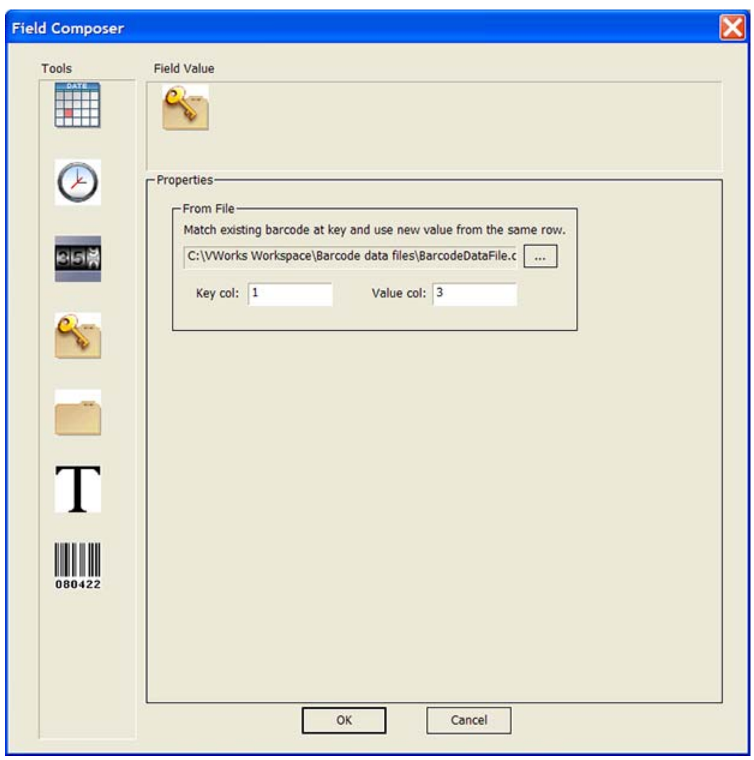
When setting the plate parameters, select **Barcode or header East** in the **Barcode information** area so the system knows to scan the east side of the incoming microplates. No barcode verification file will be used, so select **Barcode not in file.**



When setting the Print and Apply task parameter, select MyFormat for the same side: **East.**



For each of the two format fields, open the Field Composer dialog box and double-click the key-file icon in the **Tools** area to add it to the Field Value area. In the **From File** area, locate and select the data file to use. In the **Key col** box, type 1. In the **Value col** box, type 3.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • *Microplate Labeler User Guide*
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Barcode data file and barcode input file	"Tracking barcodes" on page 74
Rotate Stage (Microplate Labeler) task	"Rotate Stage (Microplate Labeler)" on page 341
Writing JavaScript for the Print and Apply task	"Using JavaScript with the CentrifugeAuto task" on page 108
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Relid

Description

The Relid task (

) places lids on the microplates using a device such as the Bravo Pipettor or a Lid Hotel Station.

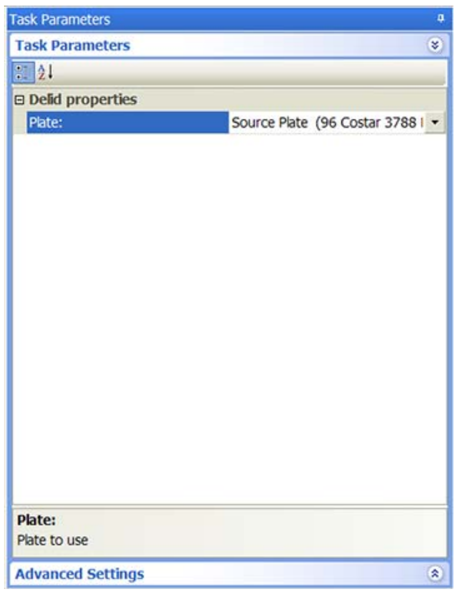
Task is available for...	Task is available in...
BenchCel Workstation	Startup Protocol Main Protocol Cleanup Protocol
Bravo Platform	Startup Protocol Main Protocol Cleanup Protocol

Task is available for...	Task is available in...
Lid Hotel Station	Startup Protocol Main Protocol Cleanup Protocol

Task Parameters

Bravo Platform

When relidding microplates on the Bravo Platform, you add the Relid task within the Bravo Subprocess. After adding the Relid task at the desired point in the Bravo Subprocess, select the microplate you want to relid in the **Task Parameters** area, as the following example shows.



Parameter	Description
Plate	The microplate that will be relidded.

Lid Hotel Station

The Relid task does not have task parameters when using the Lid Hotel Station.

Device selection

You do not need to select a device for the Relid task. The system automatically uses the correct device to relid the microplates.

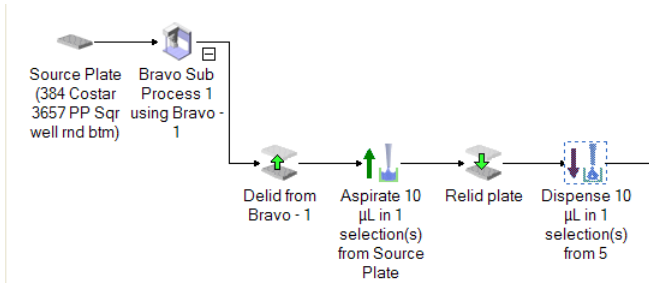
Example: Delid and relid microplates

Goal

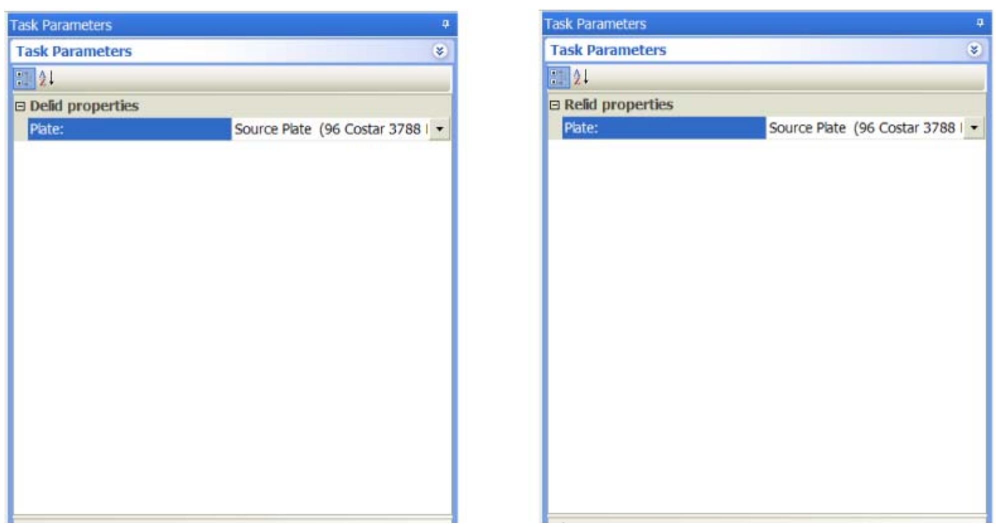
On the Bravo Platform, remove a lid from the Source Plate for an Aspirate task. After the task is finished, relid the microplate.

Implementation

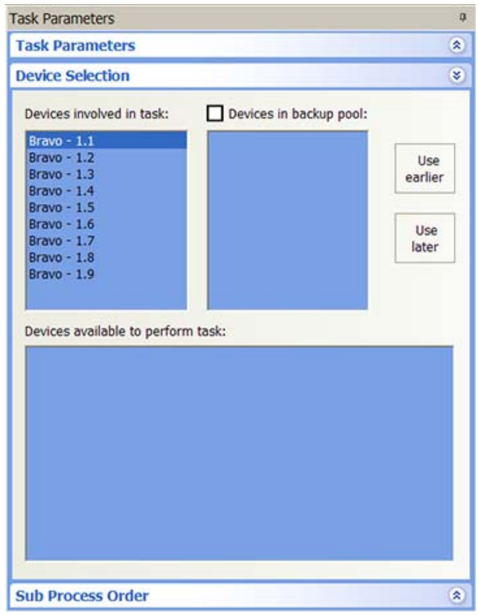
Within the Bravo Subprocess, add a Delid task before the Aspirate task to remove the lid. Add a Relid task after the Aspirate task to place the lid back on the microplate.



The task parameters for the Delid and Relid tasks are:



The device selection for the Delid task is as follows. Note that you do not need to select a device for the Relid task.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Delid task	"Delid" on page 296
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Reserve Location

Description

The Reserve Location task (Reserve Location) reserves a location on the Bravo Platform for a specified length of time. The task is typically used to incubate a microplate at that location.

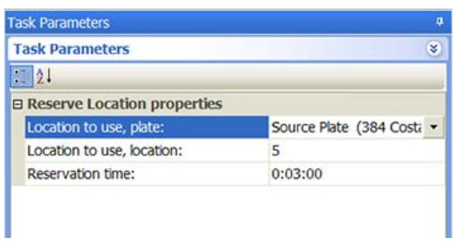
Note:

- To incubate labware on a platepad or in a Plate Hotel in a system, use the Incubate task.
- To incubate a labware in a storage device such as a Plate Hub Carousel or the LiConic StoreX incubator, use the Storage Incubate task.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

Task parameters

After adding the Reserve Location task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location to use, plate	The labware that will be placed at the specified location during the reservation period.
Location to use, location	The location that you want to reserve. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task.
Reservation time	The length of time that the location is reserved. The time starts when the desired labware arrives at the location.

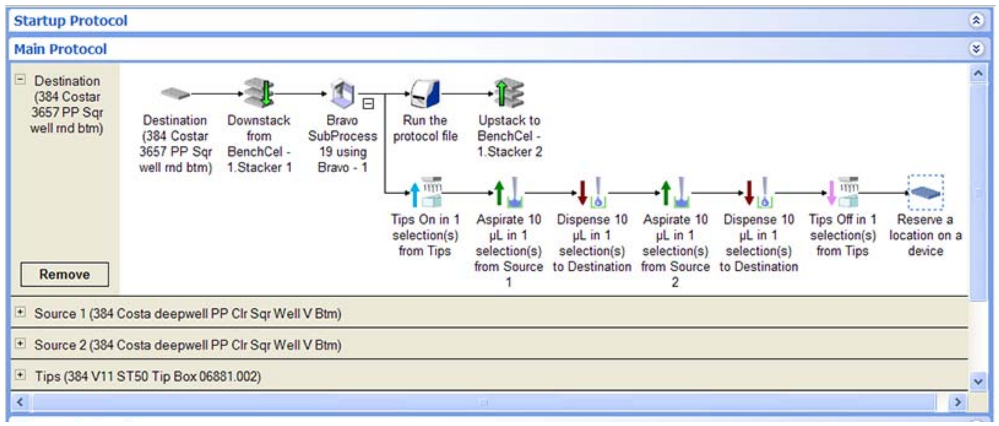
Example: Reserving a deck location for incubation

Goal

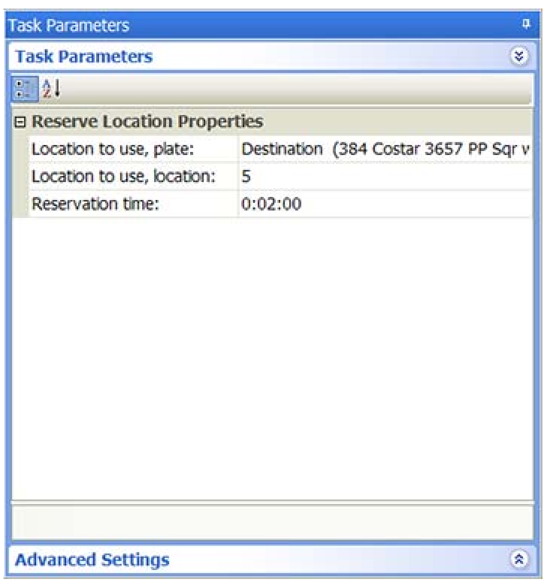
Aspirate contents from two source microplates and dispense into a destination microplate. Incubate the destination microplate at Bravo deck location 5 for 2 minutes before reading the microplate.

Implementation

In the protocol, add a Reserve Location task after the liquid-handling tasks.



The task parameters for the Reserve Location task are shown in the following example.



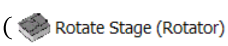
Related information

For information about...	See...
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53

For information about...	See...
Incubate task	"Incubate" on page 303
Storage Incubate task	"Storage Incubate" on page 373
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Rotate Stage (Rotator)

Description

The Rotate Stage task (

) rotates the stage on the Rotator device by 180°. You can use the task to change the orientation of microplates before they are placed in devices that require the rotated orientation.

Task is available for...	Task is available in...
Rotator	Main Protocol

Task parameters

The Rotate Stage (Rotator) task does not have task parameters.

Device selection

You must select a device for Rotate Stage (Rotator) tasks. If you have multiple devices of the same type, you can:

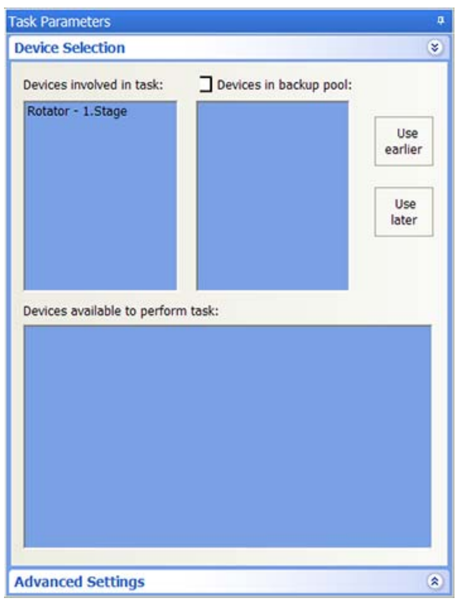
- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the list. However, if all of the devices in the list are in an error state, the software will automatically use the device in the backup pool.

IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Rotate Stage (Rotator) task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click a device in the **Devices available to perform task** area to add it to the **Devices involved in task** area.
2. To prioritize its use, select the device in the **Devices involved in task** area, and then click **User earlier** or **Use later.** The devices (deck locations) that are higher in the list are favored by the software scheduler during the protocol run.
3. *Optional.* Select backup devices to use in case all of the devices in the **Devices involved in task** area encounter an error.
   a. Select **Devices in backup pool.**
   b. Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.
   c. If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



Related information

For information about...	See...
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Rotate Stage (Microplate Labeler)

Description

The Rotate Stage (Microplate Labeler) task (Rotate Stage (Microplate Labeler)) rotates the Microplate Labeler stage 180°. You use this task to change the orientation of microplates in preparation for label applications.

Task is available for...	Task is available in...
Microplate Labeler	Main Protocol

Task parameters

The Rotate Stage (Microplate Labeler) task does not have task parameters.

Device selection

You must select a device for Rotate Stage (Microplate Labeler) tasks. If you have multiple devices of the same type, you can:

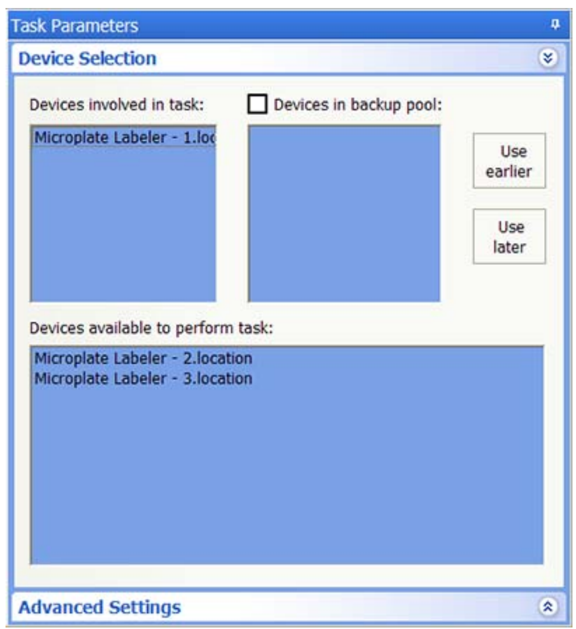
- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the list. However, if all of the devices in the list are in an error state, the software will automatically use the device in the backup pool.

IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Rotate Stage (Microplate Labeler) task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click a device in the **Devices available to perform task** area to add it to the **Devices involved in task** area.
2. To prioritize its use, select the device in the **Devices involved in task** area, and then click **User earlier** or **Use later.** The devices (deck locations) that are higher in the list are favored by the software scheduler during the protocol run.
3. *Optional.* Select backup devices to use in case all of the devices in the **Devices involved in task** area encounter an error.
   a. Select **Devices in backup pool.**
   b. Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.
   c. If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



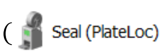
Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Print and Apply task	"Print and Apply" on page 318

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Seal (PlateLoc)

Description

The Seal (PlateLoc) task (

) seals microplates using the PlateLoc Sealer.

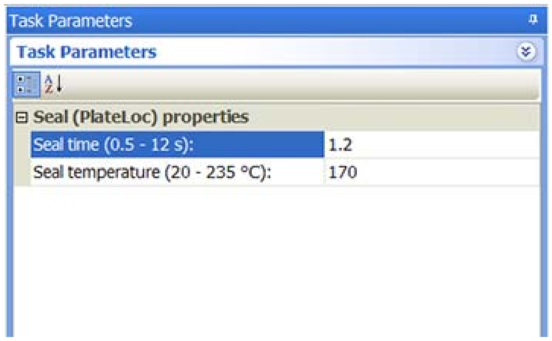
Note:

- If you require different sealing temperatures to accommodate different microplate types, Agilent Technologies recommends that you use a different PlateLoc Sealer for each sealing temperature. Using different PlateLoc Sealers that have dedicated temperature settings prevents time lost as the device heats or cools between microplate types.
- The PlateLoc Sealer immediately starts adjusting to the **Startup seal temp** defined in its profile when the profile is initialized. Initialization occurs when the protocol is first run or when you click **Initialize this profile** in the diagnostics dialog box.

Task is available for...	Task is available in...
PlateLoc Sealer	Main Protocol

Task parameters

After adding the Seal (PlateLoc) task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Seal time	The length of time the hot plate is in contact with the seal material and the microplate.
Seal temperature	The temperature of the hot plate when the seal cycle starts.

Device selection

You must select a device for Seal (PlateLoc) tasks. If you have multiple devices of the same type, you can:

- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the list. However, if all of the devices in the list are in an error state, the software will automatically use the device in the backup pool.

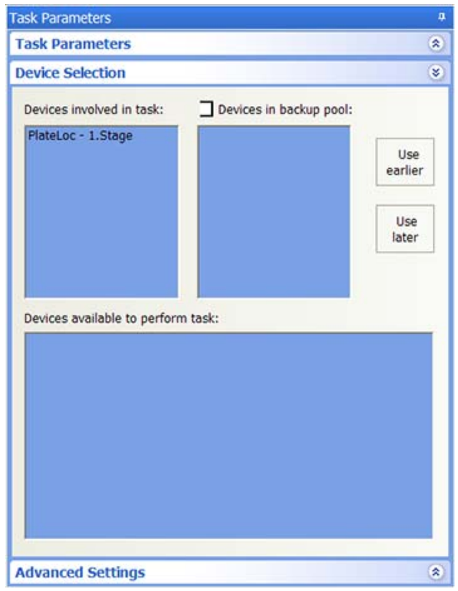
IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Seal (PlateLoc) task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click a device in the **Devices available to perform task** area to add it to the **Devices involved in task** area.
2. To prioritize its use, select the device in the **Devices involved in task** area, and then click **User earlier** or **Use later.** The devices that are higher in the list are favored by the software scheduler during the protocol run.
3. *Optional.* Select backup devices to use in case all of the devices in the **Devices involved in task** area encounter an error.
   a. Select **Devices in backup pool.**
   b. Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.

c If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



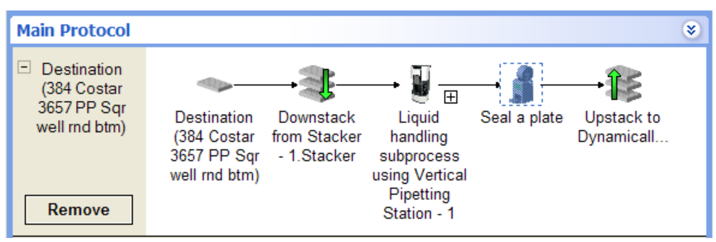
Example

Goal

Downstack microplates from a stacking device for liquid-handling tasks on the Vertical Pipetting Station, seal them, and then upstack them to any available stacker.

Implementation

Add a Seal (PlateLoc) task after the Vertical Pipetting Station Subprocess.



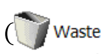
Related information

For information about...	See...
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Downstack task	“Mount” on page 307
Upstack task	“Upstack” on page 382

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

Waste

Description

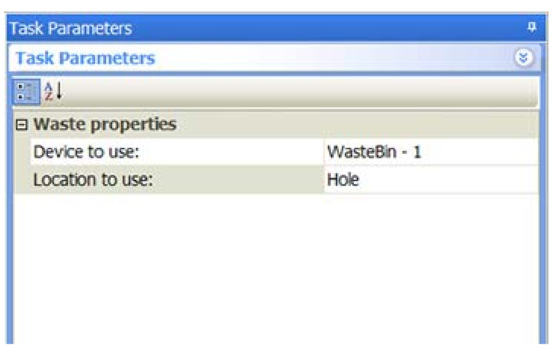
The Waste task (

) moves a lid or labware into the waste bin. For example, you can use the Waste task to discard a filter microplate after the Centrifuge task is finished.

Task is available for...	Task is available in...
Waste Bin	Main Protocol

Task parameters

After adding the Waste task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Device to use	The waste bin device you want to use for the task.
Location to use	The waste bin location.

Example: Discard filter microplates after filtering process

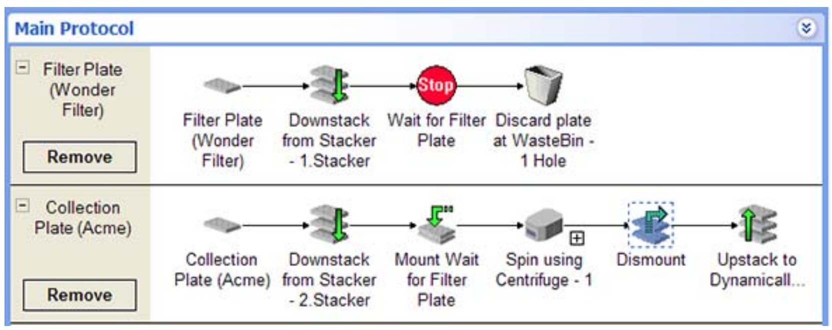
Goal

Downstack a reservoir (collection microplate) from a Labware Stacker device and place it on a platepad. Downstack a filter microplate from another Labware Stacker device and place it on top of the collection microplate. The assembly (filter microplate with collection microplate) is centrifuged. Remove the filter microplate from the collection microplate and discard it in the waste bin. Upstack the collection microplate containing the elution for future processing.

Implementation

Create one process for the collection microplate (Collection Plate) and another process for the filter microplate (Filter Plate). See “Example: Dismount a filter microplate from a collection microplate” on page 301 for details about the Mount, Dismount, and Wait For tasks.

In the Filter Plate process, add a Waste task after the Wait For task. The Waste task discards the filter microplate after the Centrifuge task is finished.



Related information

For information about...	See...
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Downstack task	“Mount” on page 307
Upstack task	“Upstack” on page 382
Mount task	“Mount” on page 307
Dismount task	“Dismount” on page 301
Microplate-handling tasks	“Setting parameters for microplate-handling tasks” on page 277
Microplate-storage tasks	“Setting parameters for microplate storage tasks” on page 349
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387

For information about...	See...
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

9
Setting parameters for microplate storage tasks

This chapter contains the following topics:

Check First Plate Orientation (Stacker)

Description

The Check First Plate Orientation (Stacker) task (Check First Plate Orientation (Stacker)) performs an orientation check on the first microplate in the specified stack. The task is primarily used in the Startup Protocol to ensure the correct loading of labware before a run starts. The task is especially useful if the labware is in a controlled-environment system and should not be taken out of the environment without operator assistance.

Note: If you want to check orientation of every microplate in the stack during the run, use the Check orientation option in the Labware Editor.

Task is available for...	Task is available in...
Labware Stacker (Stacker)	Startup Protocol Cleanup Protocol

Task parameters

The Check First Plate Orientation task does not have any task parameters.

Device selection

You must select the device for the Check First Plate Orientation task. After adding the Check First Plate Orientation task at the desired point in the Startup Protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

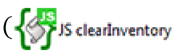
Double-click the desired device in the **Devices available to perform task** area to move it to the **Devices involved in task** area.

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Process plate properties	"Setting plate parameters" on page 46
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

clearInventory

Description

The clearInventory task (

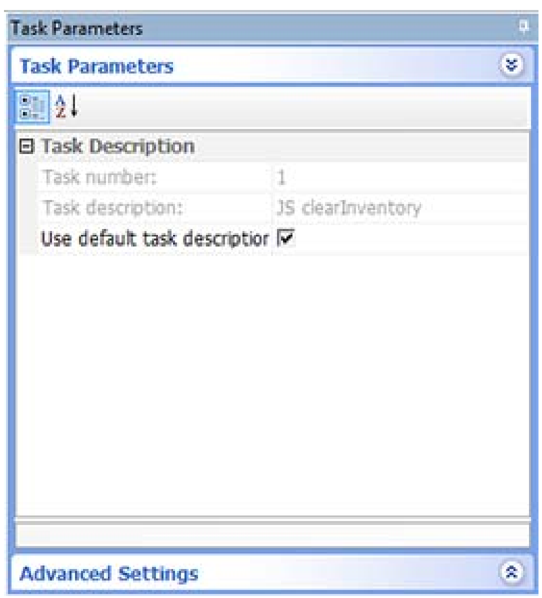
) deletes all labware data from the inventory database. Use this task to clear the inventory at the desired point in the protocol.

If you want to automatically clear the inventory every time the run starts, select **Clear inventory** in Protocol Options. See “Setting protocol options” on page 30.

Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Task parameters

After adding the clearInventory task at the desired point in the protocol, set the following **Task Parameters**:



Parameter	Description
Task number	The number that indicates the position of the task in the protocol.
Task description	The description of the task.

Parameter	Description
Use default task description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

Related information

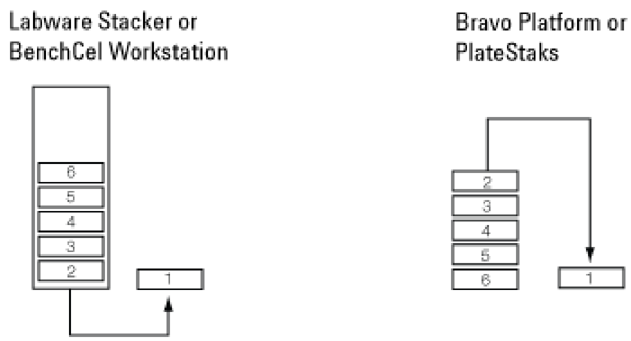
For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Process plate properties	"Setting plate parameters" on page 46
exportDatabase task	"exportDatabase" on page 358
importCsvToInventory task	"importCsvToInventory" on page 360
Load task	"Load" on page 362
Unload task	"Unload" on page 376
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Downstack

Description

The Downstack task (Downstack) moves labware as follows:

- Out of the bottom of stacking devices such as the Labware Stacker or BenchCel Workstation
- Out of the top of a stack of labware at a Bravo deck location or the Perkin Elmer PlateStaks



For example, you can use the Downstack task to move microplates out of the Stacker and onto a platepad.

You can use a single Downstack task to move labware out of multiple stacking devices. For example, after all the microplates are moved out of one stacker, the Downstack task can continue at a second stacking device.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
BenchCel Workstation	Main Protocol
Stacker	Main Protocol
Perkin Elmer PlateStaks	Main Protocol

Requirements

Bravo Platform

To use the Downstack task in a Bravo Subprocess, you must:

- Specify the maximum stack height in the Bravo deck location properties area.
- Add a Scan Stack task for each Downstack task.

BenchCel Workstation

The software always determines a location for a labware before it is downstacked from the BenchCel stacker. If you add a subprocess immediately after a Downstack task, the system will downstack the labware and place it at the location where the first subprocess task will be performed.

If you want to quarantine microplates with the wrong orientation immediately after they are downstacked from a BenchCel stacker, add a Place Plate task immediately after the Downstack task. Doing so ensures that the labware will be quarantined before proceeding to downstream tasks.

For information on how to set up quarantine criteria, see "Setting up automated error responses" on page 654.

Task parameters

After adding the Downstack task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:

Parameter	Description
Free empty stackers	The option to allow this stacker, when emptied, to become available for Upstack or Reorder tasks.

Device selection

You must select a device for Downstack tasks. If you have multiple devices of the same type, you can prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.

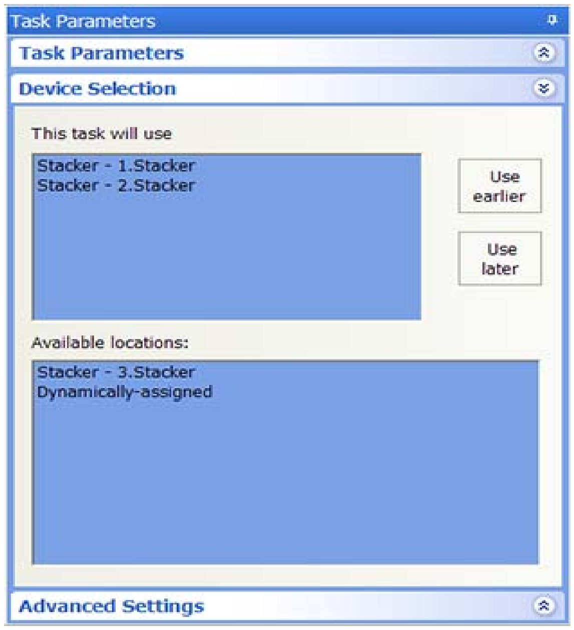
IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Downstack task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1 Double-click a device or deck location in the **Devices available to perform task** area to add it to the **Devices involved in task** area.

2 To prioritize its use, select the device in the **Devices involved in task** area, and then click **User earlier** or **Use later.** The devices that are higher in the list are favored by the software scheduler during the protocol run.



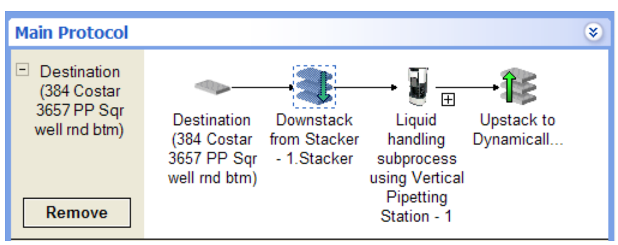
Example: Downstack microplates from Labware Stacker

Goal

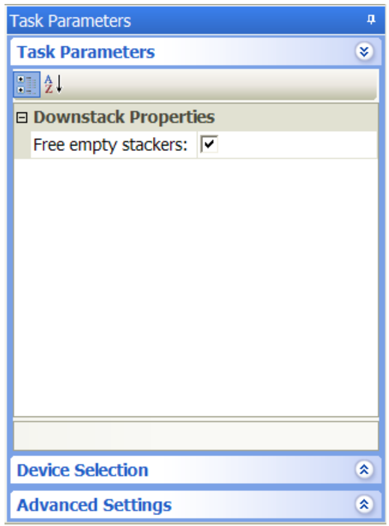
Downstack the Destination Plate from Stacker 1 for liquid-handling tasks at the Vertical Pipetting Station. After the stack is emptied, the stack can be used for Upstack and Reorder tasks.

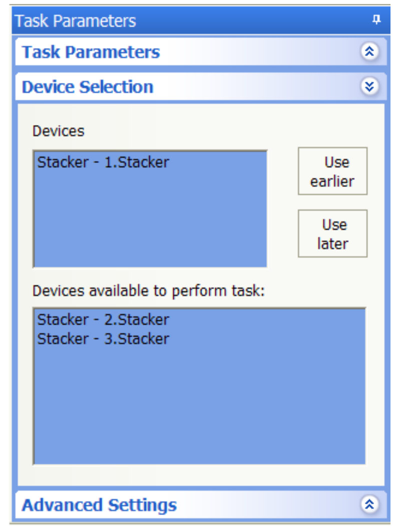
Implementation

Add a Downstack task before the Vertical Pipetting Station Subprocess task that contains the liquid-handling subroutine.



The task parameters and device selections are:





Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Scan Stack task	"Scan Stack" on page 371
Upstack task	"Upstack" on page 382
Reorder task	"Reorder" on page 367
Quarantining labware	• "Setting plate parameters" on page 46 • "Setting up automated error responses" on page 654
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

exportDatabase

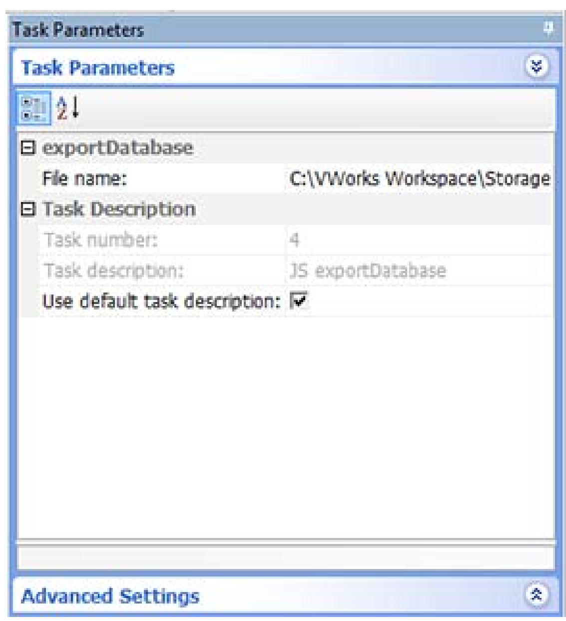
Description

The exportDatabase task () exports all data in the inventory database to the specified location. You can use this task to create a backup copy of the inventory database at the desired point in the protocol.

Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Task parameters

After adding the exportDatabase task at the desired point in the protocol, set the following **Task Parameters**:



Parameter	Description
File name	The path and name of the exported inventory file. Click the browse button to select the location and provide a name for the exported file.
Task number	The number that indicates the position of the task in the protocol.
Task description	The description of the task.

Parameter	Description
Use default task description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
clearInventory task	"clearInventory" on page 352
importCsvToInventory task	"importCsvToInventory" on page 360
Scan Stack task	"Scan Stack" on page 371
Upstack task	"Upstack" on page 382
Reorder task	"Reorder" on page 367
Quarantining labware	• "Setting plate parameters" on page 46 • "Setting up automated error responses" on page 654
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

importCsvToInventory

Description

The importCsvToInventory task (

) imports labware data in the specified CSV file into the inventory database.

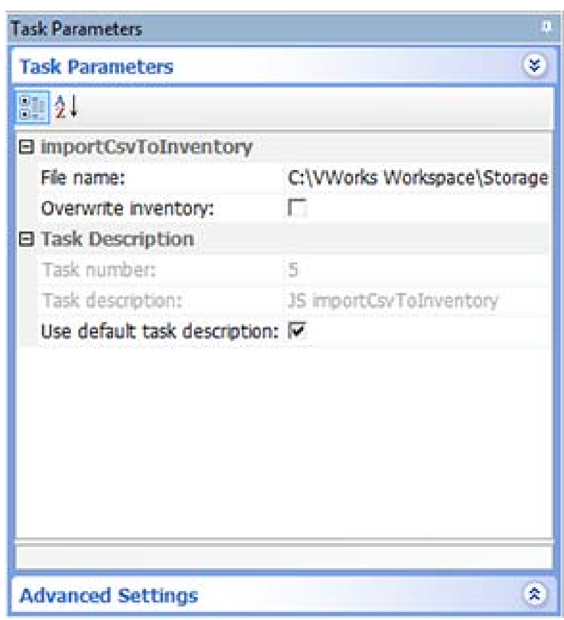
Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

For the list of CSV requirements, see the *VWorks Automation Control Setup Guide*

You can use this task to add a large amount of labware data in the inventory database. For example, in the protocol, you can add a Load task to fill a storage device with labware. If the labware do not have barcodes, or if your system does not have a barcode reader, you can add the importCsvToInventory task to enter the corresponding labware data in the inventory database.

Task parameters

After adding the importCsvToInventory task at the desired point in the protocol, set the following **Task Parameters**:



Parameter	Description
File name	The path and name of the CSV file. Click the browse button to locate and select the CSV file.
Overwrite inventory	The option to replace the existing entry in the database with the data in the CSV file.
Task number	The number that indicates the position of the task in the protocol.
Task description	The description of the task.
Use default task description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
clearInventory task	"clearInventory" on page 352
exportDatabase task	"exportDatabase" on page 358
Scan Stack task	"Scan Stack" on page 371
Upstack task	"Upstack" on page 382
Reorder task	"Reorder" on page 367
Quarantining labware	• "Setting plate parameters" on page 46 • "Setting up automated error responses" on page 654
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Load

Description

The Load task (→ Load) instructs a robot to move a defined set of labware into a storage device.

Note: If you want to move a defined set of labware from one storage device to another, use the Unload task and Load task in sequence. For information about the Unload task, see "Unload" on page 376.

Task is available for...	Task is available in...
Plate Hub Carousel	Main Protocol
Heraeus Cytomat incubator	Main Protocol
LiConic StoreX incubator	Main Protocol

Storage location selections

When you add the Load task, you can select the storage locations in one of the following ways:

- *Native locations*. Native locations are known locations in a device file. For example, if the Plate Hub Carousel is in the device file, you can select a cassette in that carousel for the Load task. When you select a cassette, all 16 slots in the cassette are available for the task. If you have only 10 microplates to load into that cassette, six of the slots will remain empty.
- *Location groups*. Location groups are device locations that are grouped together. For example, a location group can consist of 21 Plate Hub Carousel slots as follows: cassette 3 (all 16 slots) and cassette 7 slots 1 through 5 only. If you want to load exactly 21 microplates, you can select that particular location group.

Labware loading order

Under some circumstances, the original order of the labware in plate groups might not be maintained during the Load tasks:

- If the **Dynamically assign empty slot to load to storage device** option is selected in the Protocol Options, the software will find an empty slot to load labware if the original slot is occupied.
- The software always unloads the first labware in the plate group. For example, if a plate group contains microplates 1 through 5, microplate 1 is unloaded, then microplate 2, and so on. If microplate 1 is loaded back into the plate group before microplate 2 is unloaded, the software will attempt to unload the first in the sequence, microplate 1.

If keeping the original order of labware is important, do one of the following:

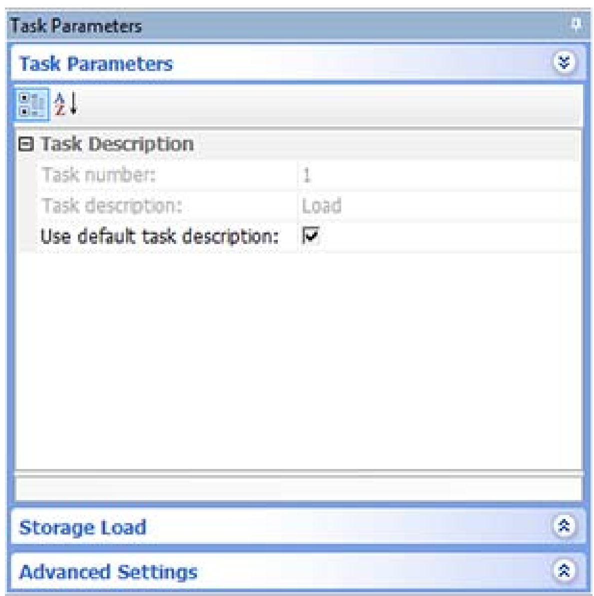
- Select the **Use original locations** option in the Load task parameters. The original storage locations will be reserved and the same labware order is preserved.

- When adding the Unload task preceding a Load task, select the **Remove plates from group when processed** option so that each labware is removed from the plate group (no longer a member of the plate group) as it is unloaded. This ensures that the labware is unloaded in the expected order. For example, a plate group consists of microplates 1 through 5. When microplate 1 is unloaded, it is removed from the plate group, so that microplate 2 becomes the first microplate in the group. If microplate 1 is loaded back into the plate group location before microplate 2 is unloaded, the software will attempt to unload microplate 2 instead of looking for microplate 1.

  When adding the Load task, add the labware to the original plate group. If you select the same plate group for the Load task, the labware will be loaded back into the plate group in the correct sequence.

Task parameters

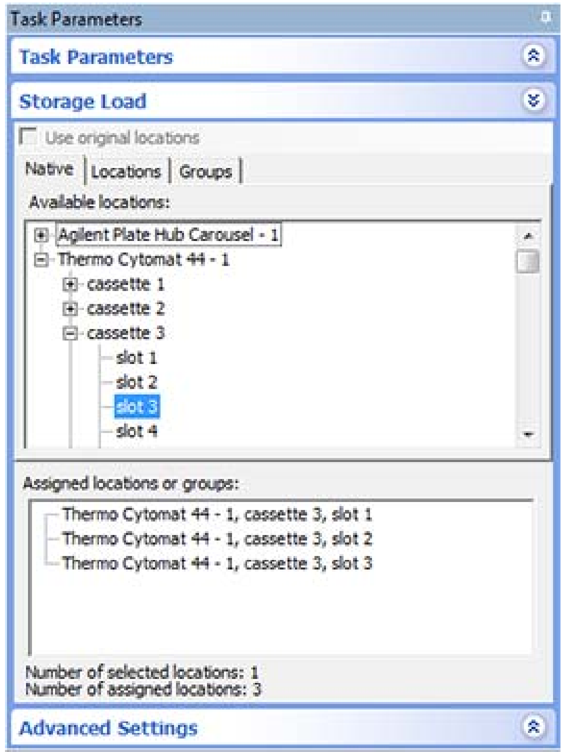
After adding the Load task at the desired point in the protocol, set the following **Task Parameters**:



Parameter	Description
Task number	The number that indicates the position of the task in the protocol.
Task description	The description of the task.
Use default task description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

Storage Load parameters

After setting the Task Parameters, click **Storage Load**, and set the following parameters:



Parameter	Description
Use original locations	The option to move a set of labware back to its original storage location and maintain the original order in storage. For example, if a set of microplates were unloaded from Plate Hub A cassette 3, selecting this option in the Load task would move the microplates back to Plate Hub A cassette 3.
Native tab	The tab that allows you to select the storage locations from a list of known locations in the device file. Double-click a location in the **Available locations** area. The selected location appears in the **Assigned locations** area.

Parameter	Description
Locations tab	The tab that allows you to select locations from a list of location groups. The location groups are created in the Inventory Editor. Double-click a location group in the **Available locations** area. The selected location group appears in the **Assigned locations** area. To view or revise existing location groups, click **Edit location groups.**
Groups tab	The tab that allows you to select a plate group into which you want to add the labware. In addition to specifying the storage location during a Load task, you can add the labware to an existing plate group. Plate groups are created in the Inventory Editor. Double-click a plate group in the **Available groups** area. The selected plate group appears in the **Assigned locations** area. To view or revise existing plate groups, click **Edit plate groups.**

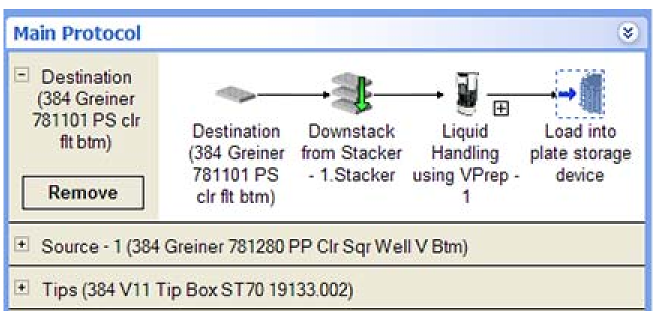
Example: Load microplates into the Plate Hub Carousel after processing

Goal

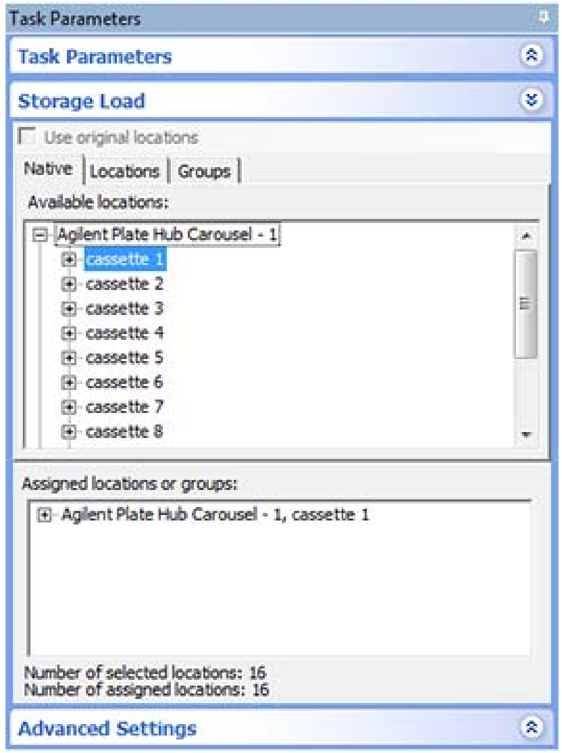
Load the destination microplates into Plate Hub Carousel cassette 1 after the specified liquid-handling tasks.

Implementation

Add the Load task after the liquid-handling tasks as shown.



In the **Storage Load** area, click the **Native** tab. Select cassette 1 in the Plate Hub Carousel to use the cassette.



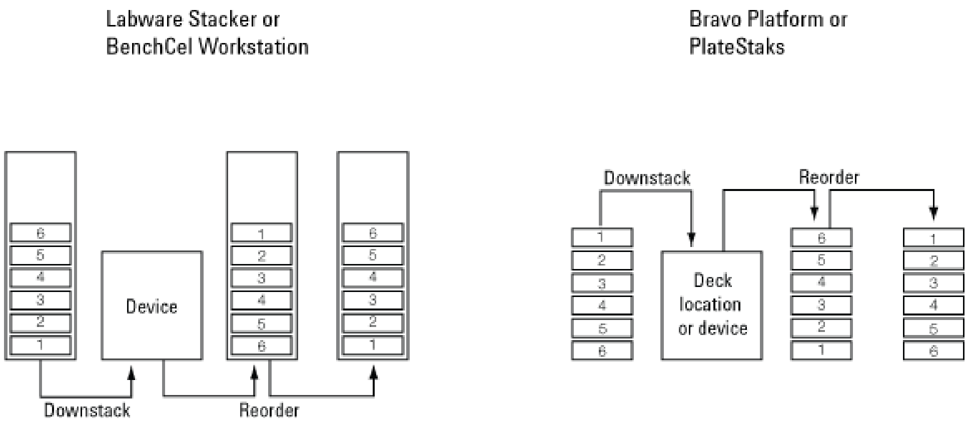
Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Setting up plate locations and plate groups in inventory manager	*VWorks Automation Control Setup Guide*
Unload task	"Unload" on page 376
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Reorder

Description

The Reorder task (Reorder) collects and incubates labware in a stack and then reverses the order of the labware in a second stack as shown in the following diagram.



You use the Reorder task to:

- Ensure liquid transfers from different source microplates occur in the same order.
- Ensure equal incubation time across all the labware in a stack.
- Minimize evaporation in the labware.

In the example shown, labware is downstacked to a device or deck location where reagents are added. The labware is then upstacked to and incubated in the first of the two reorder stacks. The labware is then moved to a second stack so that the original stack order is maintained. If the labware moves on to a downstream task, the incubation time of the first labware is the same as the last labware to leave the stack. Without the Reorder task, the first labware to move to the next task has the shortest incubation time and the last labware has the longest incubation time.

Task is available for...	Task is available in...
BenchCel Workstation	Main Protocol
Bravo Platform	Main Protocol, Bravo Subprocess
Stacker	Main Protocol
Perkin Elmer PlateStaks	Main Protocol

Requirements

General

At least two stacking locations are required for the Reorder task. For example, you need at least two Labware Stackers, two BenchCel stacker heads, or two Bravo deck locations.

Bravo Platform

To use the Reorder task in a Bravo Subprocess, you must:

- Select the desired stacking options in the Bravo profile.
- Specify the maximum stack height in the Bravo deck location properties area.
- Add two Scan Stack tasks for each pair of stacking locations used in the task.

PlateStaks

Be sure to add two Scan Stack tasks for each pair of PlateStaks locations.

Task parameters

After adding the Reorder task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:

Parameter	Description
Plate storage time	The length of time to incubate the labware in the collection stack.
Number of plates to store	The maximum number of labware allowed in the collection stack for incubation. *Note:* This value can affect the timing of the incubation. For example, if the time taken to move all labware to the first reorder stack is greater than the time specified for the incubation, the first labware cannot be moved to the next task in time. To resolve this problem, reduce the number of labware to store and add more stackers.

Device selection

You must select a device for Reorder tasks. If you have multiple devices of the same type, you can prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.

IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Reorder task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1 Double-click a device or deck location in the **Devices available to perform task** area to add it to the **Devices involved in task** area.

   Note: If you want the software to dynamically move labware to available stackers, double-click **Dynamically assigned** in the **Devices available to perform task** area. You can check the run log to determine the location of the labware in the various stackers.

2 To prioritize its use, select the device in the **Devices involved in task** area, and then click **User earlier** or **Use later.** The devices that are higher in the list are favored by the software scheduler during the protocol run.

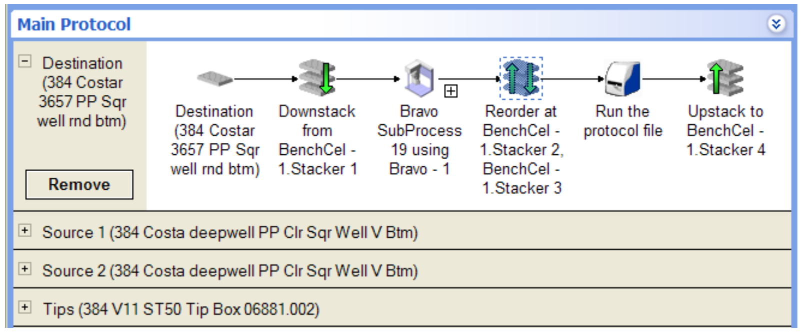
Example: Reorder microplates before reading them

Goal

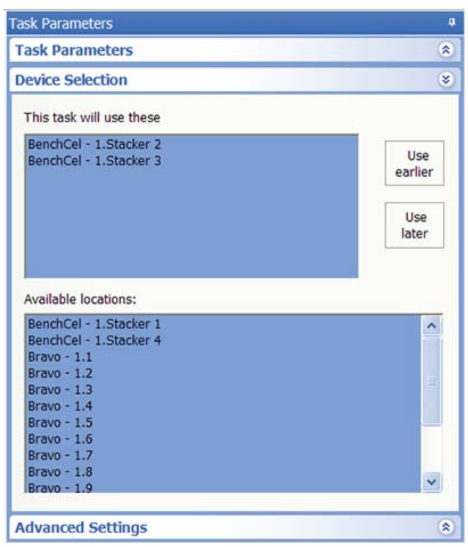
Aspirate contents from two source microplates, dispense into a destination microplate. Incubate the destination microplate for at least 20 minutes. Rearrange the microplate order to the original sequence, and then read the microplates.

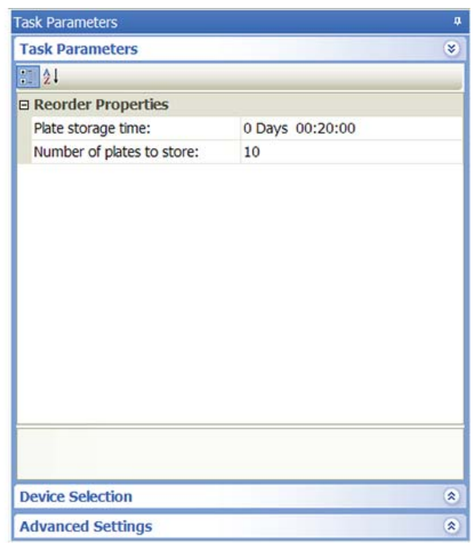
Implementation

In the protocol, add a liquid-handling subprocess to specify the tasks for adding the contents from the two source microplates into the destination microplate. Add a Reorder task after the liquid-handling tasks and before the microplate-reading task as shown in the following example.



For the Reorder task, select two stacking devices to use, and then set the Plate storage time to 20 minutes in the Task Parameters area as shown in the following examples.





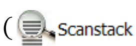
Related information

For information about...	See...
Adding devices	• “Adding devices” on page 25 • Stacking device guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Scan Stack task	“Scan Stack” on page 371
Downstack task	“Downstack” on page 354
Upstack task	“Upstack” on page 382

For information about...	See...
Microplate-handling tasks	“Setting parameters for microplate storage tasks” on page 349
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Scan Stack

Description

The Scan Stack task (

) scans a specified Bravo deck location to confirm the presence or absence of labware and determine the height of the stack. You must add one Scan Stack task for each Downstack or Upstack task, and at least two Scan Stack tasks for each Reorder task in the Bravo Subprocess.

Task is available for...	Task is available in...
Bravo Platform	Startup Protocol Cleanup Protocol
Perkin Elmer PlateStaks	Main Protocol

Task parameters

After adding the Scan Stack task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:

Parameter	Description
Device to use	The Bravo Platform on which the stacking will occur.

Parameter	Description
Location to use	The the deck location to be scanned. If the deck location will be used for a Downstack task, the robot will scan the stack to confirm the presence of the stack and determine its height. If the deck location will be used for an Upstack task, the robot will scan the location to confirm that it is empty and ready to receive labware.

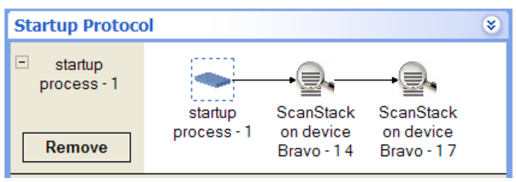
Example

Goal

Add the correct number of Scan Stack tasks to permit downstacking of microplates from deck location 4 for liquid-handling tasks. Add the correct number of Scan Stack tasks to permit uploading of those microplates to deck location 7 after processing.

Implementation

In the Startup Protocol, add two Scan Stack tasks: one to scan deck location 4 (where the microplates will be downstacked), and one to scan deck location 9 where the microplates will be upstacked.



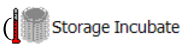
Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Downstack task	"Downstack" on page 354
Upstack task	"Upstack" on page 382
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

For information about...	See...
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Storage Incubate

Description

The Storage Incubate task (

) moves a defined set of labware into a storage device, leaves it there for a specified time period, and then moves it out of the storage device.

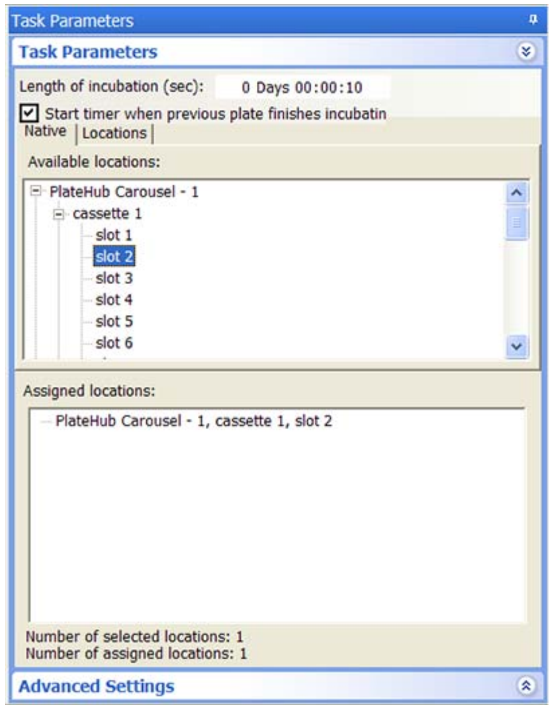
Note:

- To incubate labware on a platepad, in a Plate Hotel, or in a Vertical Pipetting Station, use the Incubate task.
- To incubate labware on the Bravo Platform, use the Reserve Location task.

Task is available for...	Task is available in...
Plate Hub Carousel	Main Protocol
Heraeus Cytomat incubator	Main Protocol
LiConic StoreX incubator	Main Protocol

Task parameters

After adding the Storage Incubate task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Length of incubation	The length of time to incubate the labware.
Start timer when previous plate finishes incubation	The option to start timing the incubation after the previous labware has finished incubating.
Native tab	The tab that allows you to select the storage locations from a list of all the known locations in the device file. Double-click a location in the **Available locations** area. The selected location appears in the **Assigned locations** area.
Locations tab	The tab that allows you to select locations from a list of location groups. The location groups are created in the Inventory Editor. Double-click a location group in the **Available locations** area. The selected location group appears in the **Assigned locations** area. To view or revise existing location groups, click **Edit location groups.**

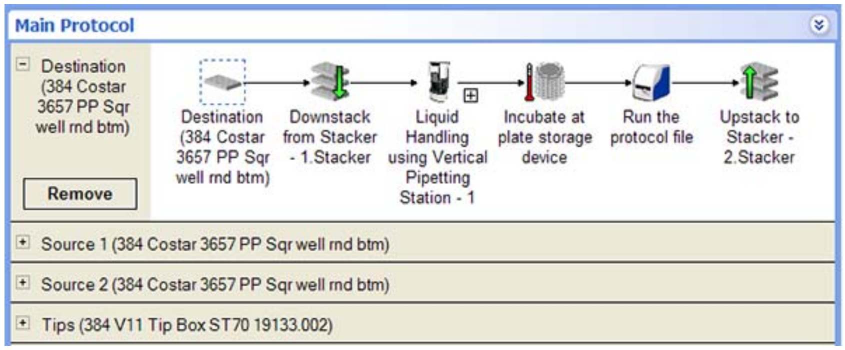
Example: Incubate destination microplates in the Plate Hub Carousel

Goal

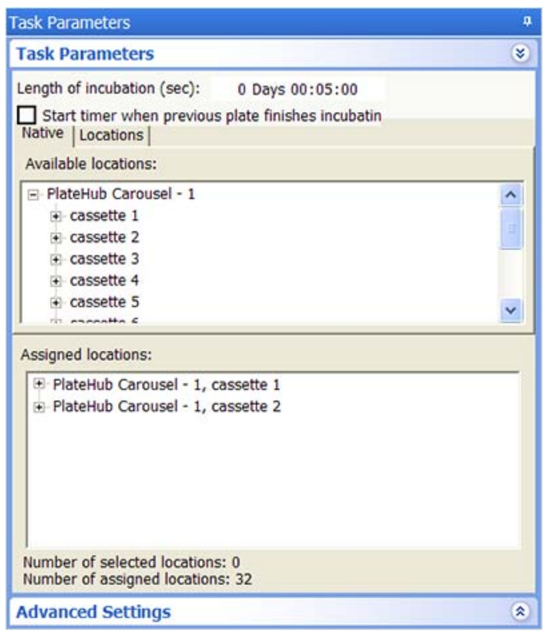
Aspirate contents from two source microplates and dispense into a destination microplate. Incubate the destination microplate in the Plate Hub Carousel for 5 minutes before reading the microplates.

Implementation

In the protocol, add a liquid-handling subprocess to specify the tasks for moving contents from the two microplates into the destination microplates. Add a Storage Incubate task after the liquid-handling subprocess and before the microplate-reading task as shown in the following example.



In the Storage Incubate Task Parameters area, set the Length of incubation at 5 minutes as shown in the following example. The example assumes that cassettes 1 and 2 are available for use.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Incubate task	"Incubate" on page 303
Reserve Location task	"Reserve Location" on page 337

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Unload

Description

The Unload task () instructs a robot to move a defined set of labware out of a storage device.

Note: If you want to move a defined set of labware from one storage device to another, use the Unload task and Load task in sequence. For information about the Load task, see "Load" on page 362.

Task is available for...	Task is available in...
Plate Hub Carousel	Main Protocol
Heraeus Cytomat incubator	Main Protocol
LiConic StoreX incubator	Main Protocol

Storage location selections

When you add the Unload task, you can select the storage locations in one of the following ways:

- *Native locations*. Native locations are known locations in a device file. For example, if the Plate Hub Carousel is in the device file, you can select a cassette in that carousel for the Load task. When you select a cassette, all 16 slots in the cassette are available for the task. If you only have 10 microplates to load into that cassette, six of the slots will remain empty.
- *Location groups*. Location groups are device locations that are grouped together. For example, a location group can consist of 21 Plate Hub Carousel slots as follows: cassette 3 (all 16 slots) and cassette 7 slots 1 through 5 only. If you want to load exactly 21 microplates, you can select that particular location group.

- *Plate groups*. Plate groups are labware that are grouped together. Each group member has location information. So when you add the Unload task and select a plate group, the software knows where to go to unload the labware.

Unload parameters that affect labware order

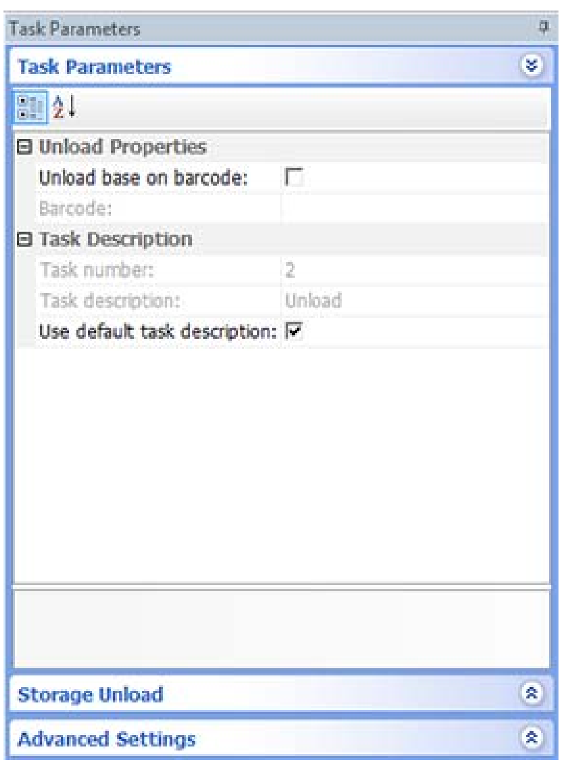
The software always unloads the first labware in the plate group. For example, if a plate group contains microplates 1 through 5, microplate 1 is unloaded, then microplate 2, and so on. If microplate 1 is loaded back into the plate group location before the labware 2 is unloaded, the software will attempt to unload the first in the sequence, labware 1.

If you plan to load any unloaded plate groups back in storage devices in the same order, do one of the following:

- Select the **Use original locations** option in the Load task parameters. The original storage locations will be reserved and the same labware order is preserved.
- When adding the Unload task, select the **Remove plates from group when processed** option. When adding the Load task, either add the labware to the original plate group, or create a new plate group and add the labware to the new group. During the unload, the labware is removed from the plate group so that the software will seek the next labware in the group. If you select the same plate group for the Load task, the labware will be loaded back into the plate group at the bottom of the sequence.

Task parameters

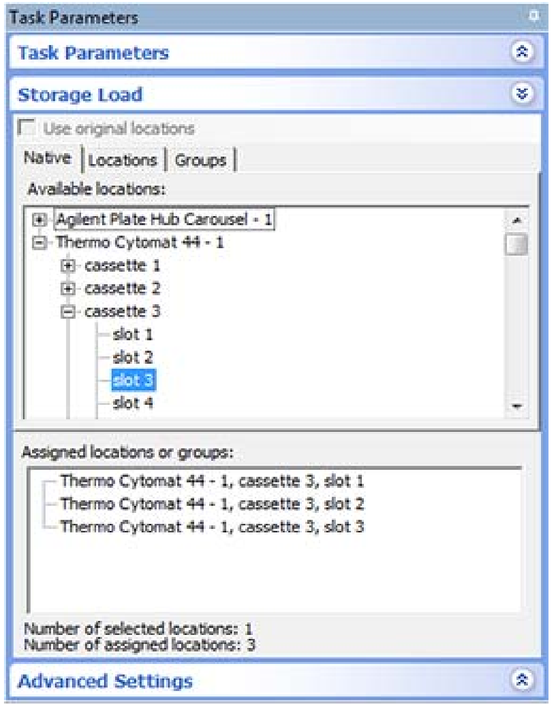
After adding the Unload task at the desired point in the protocol, set the following **Task Parameters**:



Parameter	Description
Unload based on barcode	The option to use the specified barcodes to search for and unload specific labware from storage. If you select this option, you must supply the barcodes in the **Barcode** box.
Barcode	The barcode of the labware you want to unload. You can supply one or more barcodes as follows: • Type the single barcode in the box. • Manually scan the single barcode label. • Type a JavaScript variable to supply a different barcode per labware instance. This box is only active if you selected **Unload base on barcode**.
Task number	The number that indicates the position of the task in the protocol.
Task description	The description of the task.
Use default task description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

Storage Load parameters

After setting the Task Parameters, click **Storage Load**, and set the following parameters:



Parameter	Description
Native tab	The tab that allows you to select the storage locations from a list of known locations in the device file. Double-click a location in the **Available locations** area. The selected location appears in the **Assigned locations** area.
Locations tab	The tab that allows you to select locations from a list of location groups. The location groups are created in the Inventory Editor. Double-click a location group in the **Available locations** area. The selected location group appears in the **Assigned locations** area. To view or revise existing location groups, click **Edit location groups.**
Groups tab	The tab that allows you to select a plate group that you want to move out of storage. Double-click a plate group in the **Available groups** area. The selected plate group appears in the **Assigned locations** area. To view or revise existing plate groups, click **Edit plate groups.**
Remove plates from group when processed	The option to remove the specified labware from a plate group after the labware is moved out of storage.

Note: A compiler error will appear if you try to unload from a plate group (Groups tab) and another storage location type (Native tab or Locations tab).

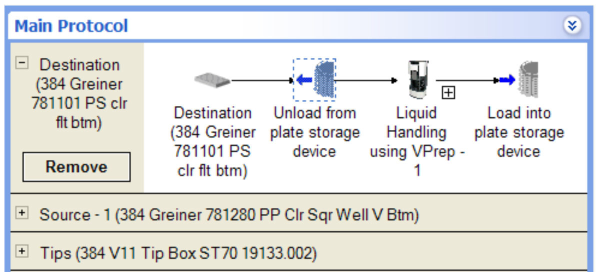
Example: Unload microplates from the Plate Hub Carousel for processing

Goal

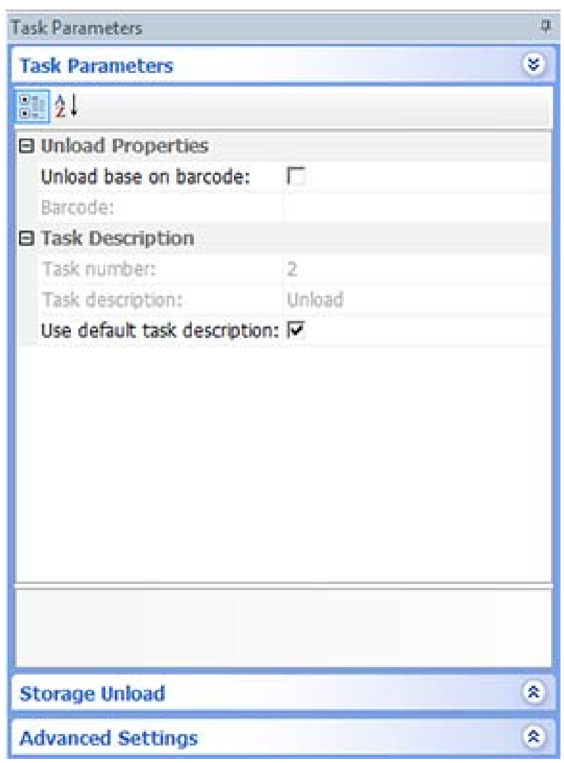
Unload the specified microplates from Plate Hub Carousel cassette 1 in preparation for liquid-handling tasks. In this example, barcodes will not be used to unload the microplate.

Implementation

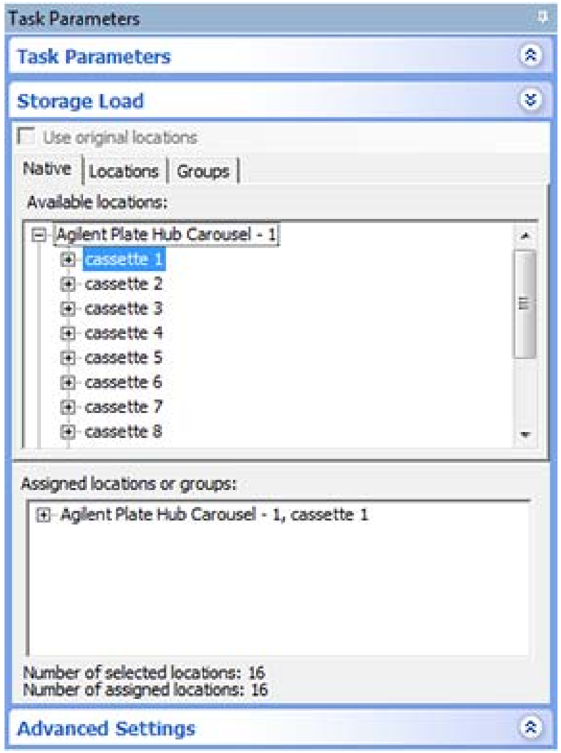
In the protocol, add the Unload task before the liquid-handling tasks as shown.



In the Task Parameters area, clear the **Unload based on barcode** check box.



In the Storage Load area, click the Native tab. Select cassette 1 in the Plate Hub Carousel to use the cassette.



Related information

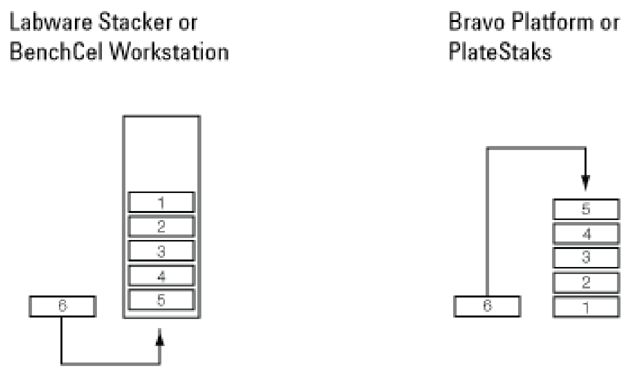
For information about...	See...
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Setting up plate locations and plate groups in inventory manager	*VWorks Automation Control Setup Guide*
Load task	“Load” on page 362
Microplate-handling tasks	“Setting parameters for microplate storage tasks” on page 349
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Upstack

Description

The Upstack task (Upstack) moves labware as follows:

- Into the bottom of stacking devices such as the Labware Stacker or the BenchCel Workstation
- Onto the top of a stack of labware at a Bravo deck location or the Perkin Elmer PlateStaks



For example, you can use the Upstack task to move a microplate from a platepad into the Labware Stacker.

You can use a single Upstack task to move labware into multiple stacking devices. For example, if three stackers are available, the upstack task can dynamically move microplates into any of the three stacking devices.

Task is available for...	Task is available in...
BenchCel Workstation	Main Protocol
Bravo Platform	Main Protocol, Bravo Subprocess
Stacker	Main Protocol
Perkin Elmer PlateStaks	Main Protocol

Requirements

Bravo Pipettor

To use the Upstack task in a Bravo Subprocess, you must:

- Select the desired stacking options in the Bravo profile.
- Specify the maximum stack height in the Bravo deck location properties area.
- Add a Scan Stack task for each Upstack task.

Task parameters

The Upstack task does not have any task parameters.

Device selection

You must select a device for Upstack tasks. If you have multiple devices of the same type, you can prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.

IMPORTANT The multiple devices must have the same setup and configuration.

After adding the Upstack task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click a device or deck location in the **Devices available to perform task** area to add it to the **Devices involved in task** area.

   Note: If you want the software to dynamically move labware to available stackers, double-click **Dynamically assigned** in the **Devices available to perform task** area. You can check the run log to determine the location of the labware in the various stackers.
2. To prioritize its use, select the device in the **Devices involved in task** area, and then click **User earlier** or **Use later.** The devices that are higher in the list are favored by the software scheduler during the protocol run.

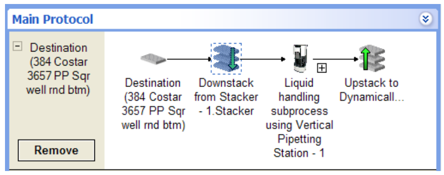
Example 1: Dynamically upstack microplates into a stacking device

Goal

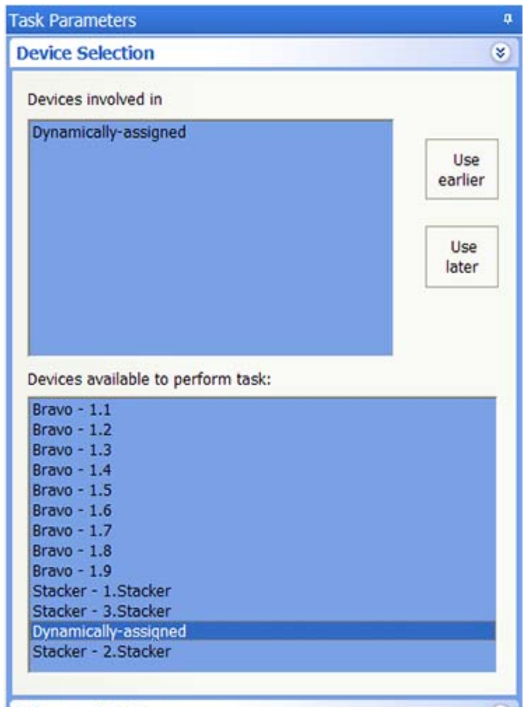
After the Destination Plate is finished with the Bravo liquid-handling process, upstack it to any available stacker.

Implementation

Add a Upstack task after the Bravo Subprocess task that contains the liquid-handling sub-routine.



The device selection is:



Example 2: Stacking on the Bravo Platform

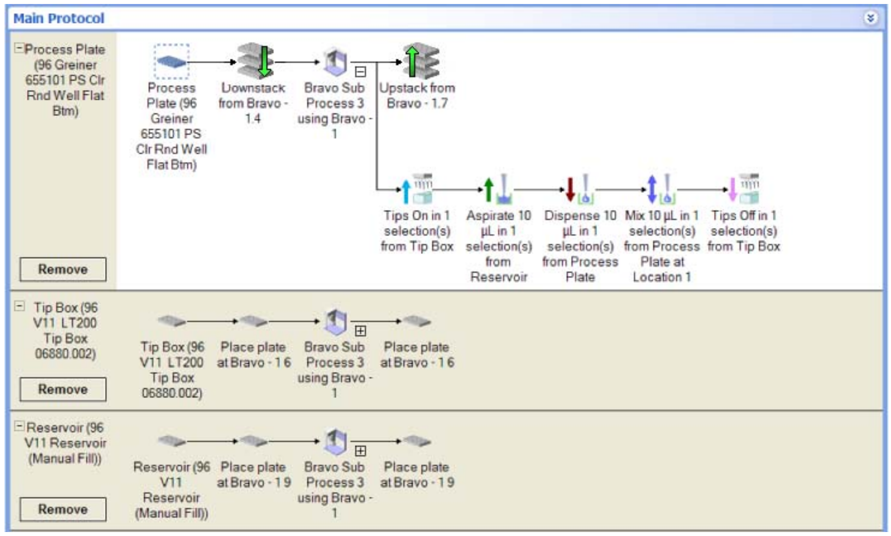
Goal

Downstack microplates for liquid-handling process, then upstack them.

Implementation

The protocol is set up as follows:

- A stack of microplates (Process Plates) is at deck location 4.
- A tip box (Tip Box) is configured at deck location 6.
- A Manual Fill Reservoir (Reservoir) is configured at deck location 9.
- The Startup Protocol (not shown) contains Scan Stack tasks for deck locations 4 (where the stack starts) and 7 (where the stack will end up after upstacking).



The protocol performs the following:

1. The robot scans deck location 4 to confirm the presence of the stack and determines the number of microplates. (The Scan Stack task in the Startup Protocol, not shown, performs this task.)
2. The robot scans deck location 7 to confirm the absence of labware. (The Scan Stack task in the Startup Protocol, not shown, performs this task.)
3. The top-most Process Plate is downstacked from deck location 4 to deck location 1.

   By default, when the <auto-select> location is selected for a process plate task, the process plates are always placed or downstacked at deck location 1. If deck location 1 is occupied, the process plate will be placed or downstacked at the next-available location.
4. Tips are installed on the pipette head at deck location 6.
5. The robot aspirates solution from the Reservoir at deck location 9.
6. The robot dispenses the solution into the Process Plate at deck location 1.
7. The robot mixes the solution in the Process Plate at deck location 1.
8. The pipette tips are removed at the Tip Box at deck location 6.
9. The Process Plate at deck location 1 is upstacked to deck location 7.
10. Steps 1 through 8 is repeated for each Process Plate stacked at deck location 4.

Related information

For information about...	See...
Adding devices	“Adding devices” on page 25
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Scan Stack task	“Scan Stack” on page 371
Downstack task	“Downstack” on page 354

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

10
Setting parameters for liquid-handling tasks

This chapter describes the Bravo and Vertical Pipetting Station liquid-handling tasks you can add in a protocol:

SubProcess (Bravo, Vertical Pipetting Station)

Description

The Subprocess (Bravo) (SubProcess (Bravo)) and Subprocess (Vertical Pipetting Station) (SubProcess (Vertical Pipetting Station)) tasks indicate the start of a protocol subroutine that employs either the Bravo Platform or the Vertical Pipetting Station. Within the subprocess, you can add tasks that are unique to the device. You can expand or collapse the subprocess to show or hide the subprocess tasks.

IMPORTANT All Bravo or Vertical Pipetting Station liquid-handling tasks must be added within a Bravo or Vertical Pipetting Station Subprocess respectively.

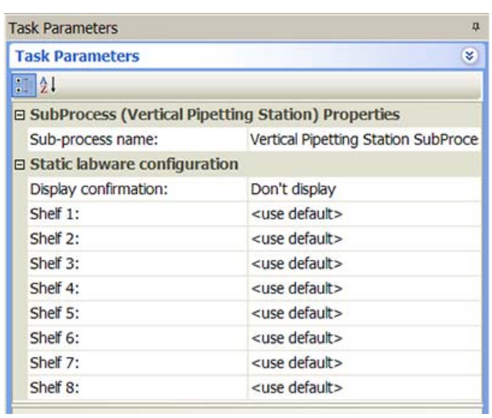
Task is available for...	Task is available in...
Bravo Platform	Main Protocol
Vertical Pipetting Station	Main Protocol

Task parameters

IMPORTANT The latest version of the VWorks software is backward-compatible with protocols created in VWorks4 version 6.2.3 or earlier and will continue to support static labware configuration procedures. However, Agilent Technologies recommends that you use the concepts and procedure in “Configuring labware” on page 40 when writing new protocols.

After adding the Subprocess task at the desired point in the protocol, you have the option of assigning static labware to locations on a device. You assign static labware in the subprocess to override the static labware configuration in the Startup Protocol. For more information about static labware, see “Planning labware use” on page 20.

To assign static labware to locations on the device, set the following parameters in the **Task Parameters** area:



Parameter	Description
Subprocess name	The name of the subprocess. Select from the list of available subprocesses currently in the protocol.
Display confirmation	The option to display a message at the beginning of the protocol run and the subprocess to remind you to verify the physical locations of the labware match what you specified in the software.
Location/Shelf n	The Bravo deck locations or Vertical Pipetting Station shelves. Select the labware for the specific locations or shelves.

Device selection

You must select a device for Subprocess tasks. If you have multiple devices of the same type, you can:

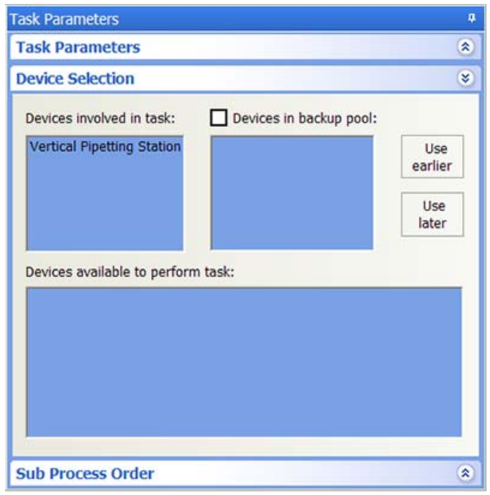
- Prioritize the list of devices for the task. If the first device in the list is busy, the software will automatically use the next device in the list. If all of the devices in the list are busy, the task that needs the device will wait until one becomes available.
- Set up a backup pool. If the primary device encounters an error, the software will automatically use the next device in the list. However, if all of the devices in the list are in an error state, the software will automatically use the device in the backup pool.

IMPORTANT The multiple devices must be configured with the same static labware and accessories, and all the accessories must have identical setups.

After adding the Subprocess task at the desired point in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

1. Double-click the desired device (deck location or shelf) in the **Devices available to perform task** area to move it to the **Devices involved in task** area. If you have multiple devices of the same type, you can move them to the **Devices involved in task** area.
2. If you have multiple devices in the **Devices involved in task** area, select a device, and then click **Use earlier** or **Use later** to prioritize it.
3. *Optional*. Select backup devices to use in case all of the devices in the **Devices involved in task** area encounter an error.
   a. Select **Devices in backup pool.**
   b. Drag one or more devices from the **Devices available to perform task** area to the **Devices in backup pool** area.
   c. If you have multiple devices in the backup pool, select a device in the **Devices in backup pool** area, and then click **Use earlier** or **Use later** to prioritize it.



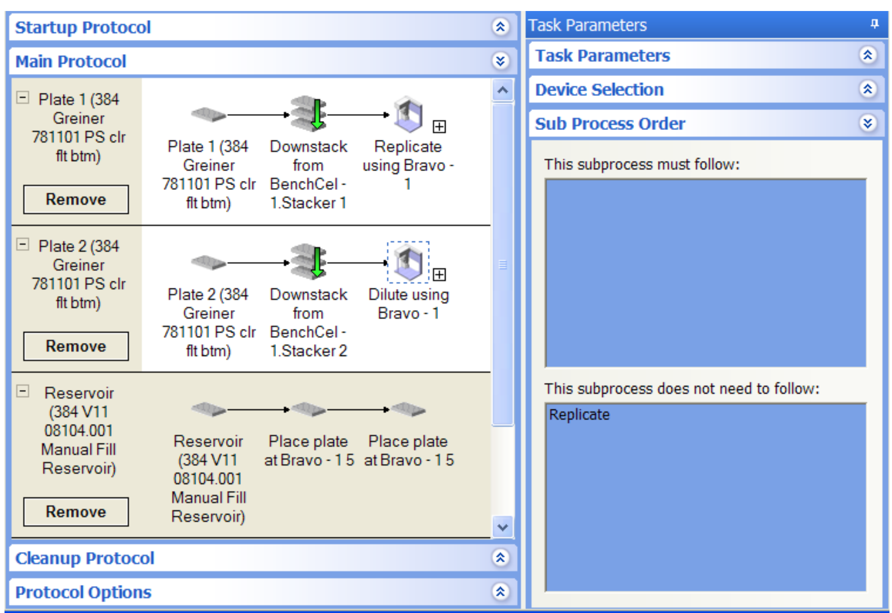
Subprocess Order

If more than one sub-process uses the same configured labware, and the sub-processes are in different protocol processes, you can specify the sequence in which the sub-processes will be performed.

To specify the sequence in which the sub-processes will be performed on the same configured labware:

1 In the protocol process, select the sub-process that contains the task that uses the configured labware.

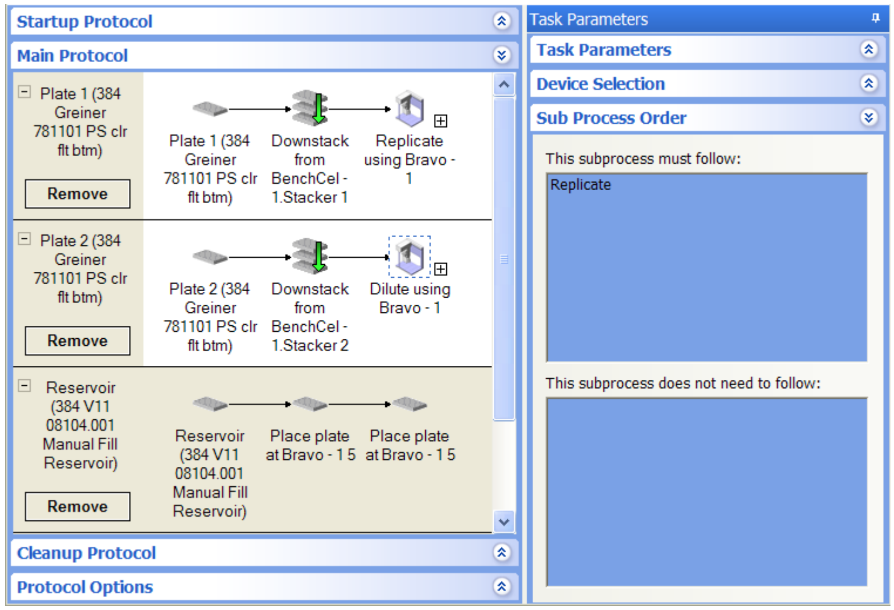
In the following example, the Bravo Subprocess named Dilute is selected.



2 In the **Task Parameters** area, click **Sub Process Order.**

3 In the **Sub Process Order** area, double-click the subprocess names to rearrange the order.

In the following example, the Dilute subprocess is selected. So the task sequence will be Replicate subprocess, followed by the Dilute subprocess.



Related information

For information about...	See...
Configured labware	"Planning labware use" on page 20
Static labware	"Planning labware use" on page 20
Startup Protocol	"Setting up Startup and Cleanup Protocol processes" on page 60
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Aspirate (Bravo, Vertical Pipetting Station)

Description

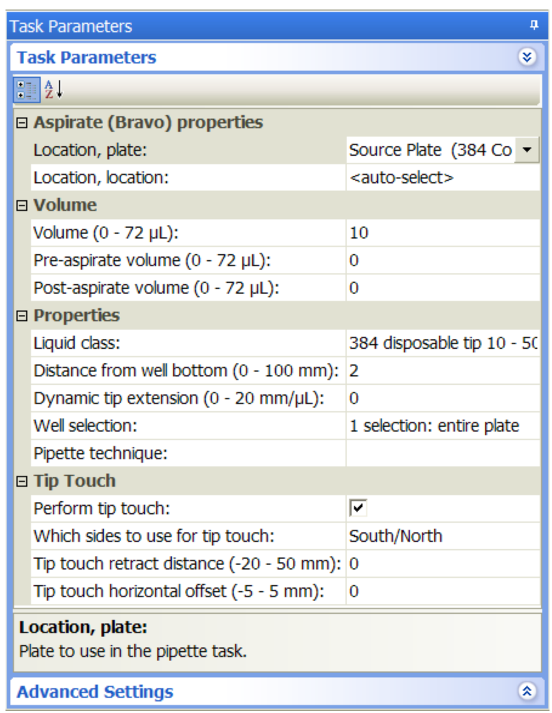
The Aspirate (Bravo) (Aspirate (Bravo)) and Aspirate (Vertical Pipetting Station) (Aspirate (Vertical Pipetting Station)) tasks draw liquid from a microplate or reservoir.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Task parameters

Note: The task parameters for Aspirate (Bravo) and Aspirate (Vertical Pipetting Station) are identical.

After adding the Aspirate task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Aspirate task.

Parameter	Description
Location, location	The location at which the Aspirate task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Volume (µL)	The volume of liquid to be drawn into each pipette tip.
Pre-aspirate volume (µL)	The volume of air to be drawn before the pipette tips enter the liquid.
Post-aspirate volume (µL)	The volume of air to be drawn after the liquid is drawn.
Liquid class	The pipetting speed and accuracy. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.
Distance from well bottom (0–100 mm)	The distance between the end of the pipette tips and the well bottoms during the Aspirate task. If you specify dynamic tip extension, this is the distance at the end of the Aspirate task. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.)

Parameter	Description
Dynamic tip extension (0–20 mm/µL)	The rate at which the pipette head moves during the Aspirate task. The software calculates the distance over which the tips will move without crashing. Use dynamic tip extension to prevent spills as the pipette tips displace the liquid. To move the tips: • *At the same rate as the volume change.* Calculate dynamic tip extension (DTE) as follows: DTE = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTE > 1/A • *Slower than the volume change.* DTE < 1/A The starting and ending positions can be calculated as follows: $(V_{aspirated} * DTE) + Distance_{well\ bottom}$ *Note:* Instead of a negative aspirated volume, the software automatically moves downward toward the well bottom with each aspirate action.
Well selection	The wells at which the Aspirate task occurs. Click in the parameter box, and then click the Browse button to select the wells in the Well Selection dialog box. Use this parameter only if the pipette head has fewer tips than the number of wells in the microplate, or if you are in single-row or single-column mode.
Pipette technique	The pipette location offset you want to use for the Aspirate task. The list of pipette techniques are defined in the Pipette Technique Editor.
Perform tip touch	The option to touch the pipette tip on one or more sides of the well.
Which sides to use for tip touch	The side or sides of the well to use during tip touch: North, South, East, West, North/South, West/East, West/East/South/North.
Tip touch retract distance (–20 to 50 mm)	The vertical distance for the pipette tips to rise before touching the sides of the wells.

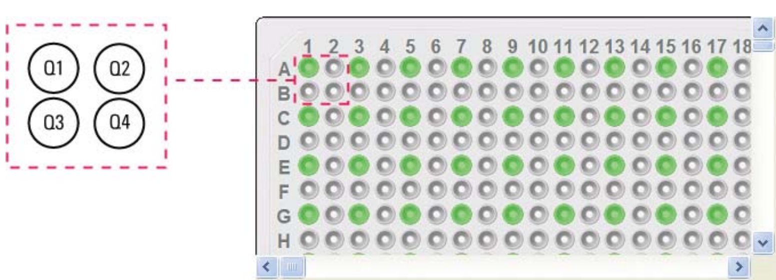
Parameter	Description
Tip touch horizontal offset (–5 to 5 mm)	The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition. For example, if you set a value of: • 0, the tips move a horizontal distance equal to the well radius • > 0, the tips attempt to move past the well radius, which results in a more forceful tip touch • < 0, the tips move a distance less than the radius of the well, resulting in a lighter tip touch

Quadrant pattern well selection

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head. The following table lists the types of pipette heads and the number of accessible quadrants in various microplates.

Pipette head channels/ pin tool pins	Microplate	Number of quadrants
96	96-well	1
	384-well	4
	1536-well	16
384	384-well	1
	1536-well	4
1536 (pin tool only)	1536-well	1

The following diagram demonstrates the concept of quadrants. The diagram shows a portion of a 384-well microplate and highlights the four quadrants (Q1, Q2, Q3, and Q4) that are accessible by the A1 tip of a 96-channel pipette head. Notice that the green color highlights all of the quadrant 1 (Q1) wells across the microplate.



Instead of a column- or row-wise pattern, you can select a quadrant pattern during well selection.

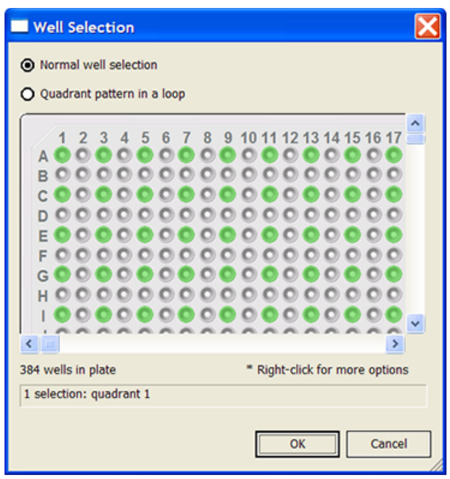
The quadrant pattern option is available only if:

- The number of channels in the pipette head (or pins in a pin tool) is fewer than the number of wells in the microplate. For example, you can use a 96-channel pipette head to dispense liquid into a 384-well microplate or 1536-well microplate.
- All the channels are selected in the Set Head Mode task when using a pipette head. (The Set Head Mode task is not an option when using a pin tool).
- The liquid-handling task is inside a loop.

IMPORTANT If you select a quadrant pattern, specifications in the Well Selection dialog box will override `task.Wellselection` values assigned in the Advanced Settings area.

To select a quadrant pattern:

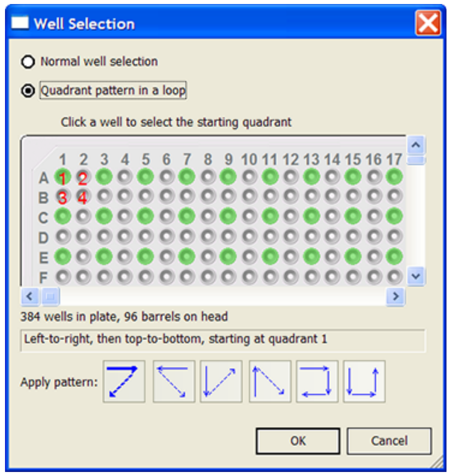
1 In the **Task Parameters** area, click the **Well selection** parameter box, and then click the Browse button. The Well Selection dialog box opens. By default, the **Normal well selection** option is selected. This option is used for column- and row-wise liquid-handling patterns.



2 Select **Quadrant pattern in a loop.** The contents of the dialog box change. Notice the following:

- Red numbers (1 through 4) appear on wells A1, A2, B1, and B2. The numbers indicate the pipetting sequence: 1 is the starting well, and 4 is the last well. In the following example, the sequence is A1, A2, B1, B2.
- Green wells indicate the starting well in the pipetting sequence.
- Pattern buttons at the bottom of the dialog box indicate the movement of the pipette channels. The movement description is provided in the text box above the buttons.

  Note: The last two patterns are unavailable if a group contains 16 wells. For example, the last two patterns are not available if you have a 96-well pipette head and a 1536-well microplate.



3 Select the starting well. The well becomes green and is labeled 1.

In the following example, the third quadrant (B1 well) is selected.

4 Click a pattern button to specify the pipette channel movement. After you click a pattern, the red numbers in the graphic are updated to show the sequence.

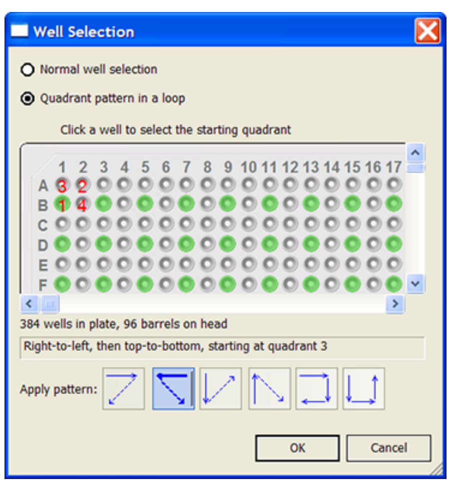
In the following example, the second pattern is selected (right-to-left, then top-to-bottom). The third quadrant (B1) is the starting well. The resulting movement is:

Quadrant 3 (B1)

Quadrant 2 (A2)

Quadrant 1 (A1)

Quadrant 4 (B2)



5 When you are finished, click **OK** to save the changes and return to the VWorks window.

Example: Aspirate from a source microplate on the Bravo Platform

Goal

Aspirate contents from a source microplate (Source 1) and dispense into a destination microplate.

Implementation

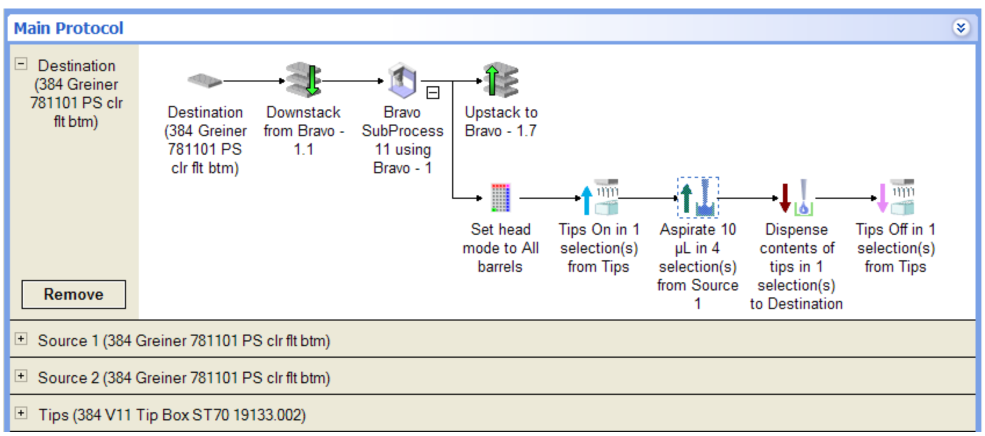
The Bravo deck is physically set up as follows:

- The destination microplates are at Bravo deck location 1.
- The source microplate is at deck location 2.
- The tipbox is at deck location 9.

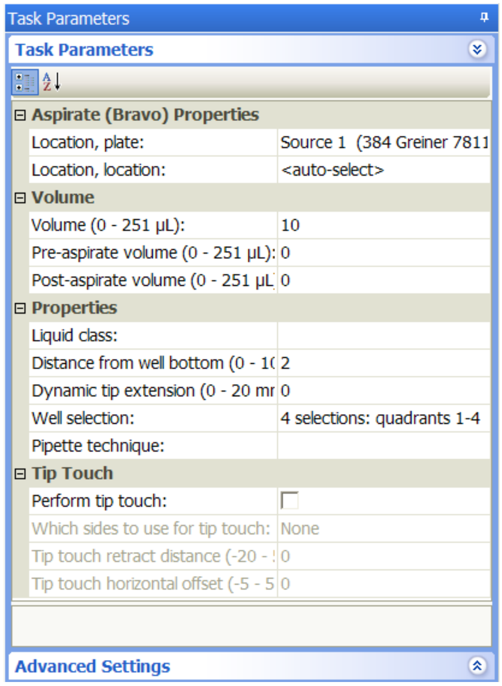
In the protocol, the following are added:

- Process for the destination microplate
- Configured labware for the source microplate
- Configured labware for the tipbox

In the Destination plate process, a Bravo subprocess is added. Within the subprocess, an Aspirate task and a Dispense task are added as shown in the following example.



In the Aspirate Task Parameters area, Source 1 is selected, because the goal is to aspirate from that microplate.



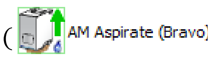
Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Liquid classes	*VWorks Automation Control Setup Guide*
Pipette techniques	"Specifying pipetting techniques" on page 613
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
Set Head Mode task	"Set Head Mode (Bravo)" on page 529
Tips On task	"Tips On (Bravo, Vertical Pipetting Station)" on page 542
Tips Off task	"Tips Off (Bravo, Vertical Pipetting Station)" on page 539
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349

For information about...	See...
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

AM Aspirate (Bravo)

Description

The AM Aspirate (Bravo) (

) task is designed for aspirating fluids through AssayMAP Bravo cartridges that are mounted on the Bravo 96AM Head.

Task is available for...	Task is available in...
Bravo Platform with Bravo 96AM Head	Main Protocol, Bravo Subprocess

The AM Aspirate (Bravo) task has the same task parameters as the Aspirate (Bravo) task, except for the following parameters:

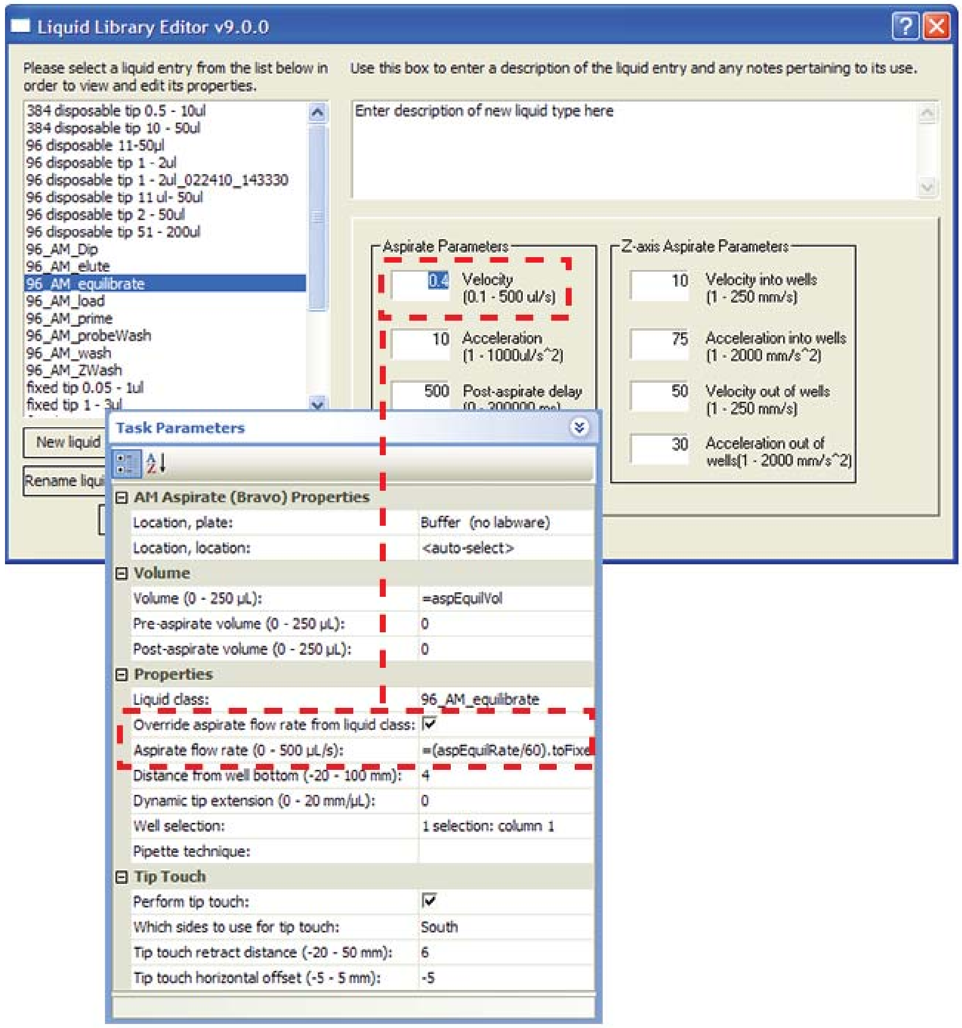
- Override aspirate flow rate from liquid class

  Precise flow-rate control is critical for the performance of the AssayMAP Bravo cartridges. The option to override the aspirate flow rate (velocity) in the liquid class enables an operator to change the flow rate from experiment to experiment for rapid method development without modifying the liquid class. Likewise, the flow rate can vary for different AM Aspirate tasks that use the same liquid class in a given protocol.

- Distance from well bottom (-20–100 mm)

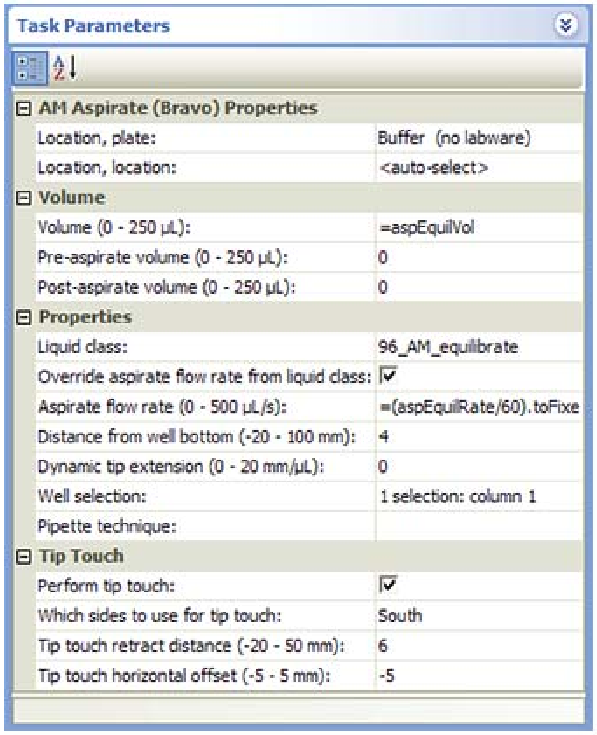
  The range includes negative values, which is critical for steps that require using the bare probes to aspirate from the top portion of unmounted cartridges.

Figure Liquid class velocity setting in the Liquid Library Editor and flow rate override option



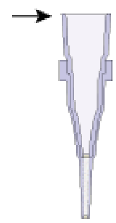
Task parameters

After adding the AM Aspirate task at the desired point in the protocol, set the following parameters in the **Task Parameters** area.



Parameter	Description
Location, plate	The labware involved in the AM Aspirate task.
Location, location	The location at which the AM Aspirate task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck, the software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task.
Volume (µL)	The volume of liquid to be drawn into each probe, cartridge, or tip.
Pre-aspirate volume (µL)	The volume of air to be drawn before the probes, cartridges, or tips enter the liquid.
Post-aspirate volume (µL)	The volume of air to be drawn after the liquid is drawn.

Parameter	Description
Liquid class	The pipetting speed and accuracy. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.
Override aspirate flow rate from liquid class	The option to override the aspirate velocity in the specified liquid class. Selecting this option enables you to specify a value for the aspirate flow rate without changing the liquid class.
Aspirate flow rate (0–500 µL/s)	The numerical value or the JavaScript variable that will override the aspirate velocity setting in the liquid class. A JavaScript variable enables the value to be assigned later. For example, using a VWorks form, an operator could easily change the flow rate for an aspirate step in increments from as low as 1 µL/min up to 2000 µL/min or more using the same liquid class. If the task is included in a VWorks macro, a JavaScript variable enables you to change the value for the task at the macro level. **IMPORTANT** The software requires that the flow rate value be in microliters per second (µL/s) at run time. If you want an operator to enter the value in microliters per minute (µL/min), you can use scripting to convert the values for the software to use.

Parameter	Description
Distance from well bottom (-20–100 mm)	The distance between the tips of the probes, cartridges, or disposable tips and the *well bottoms* during the AM Aspirate task. If you specify dynamic tip extension, this is the distance at the end of the AM Aspirate task. Use a positive number for tasks that are performed with mounted cartridges, mounted tips, or bare probes at labware other than a cartridge rack. If the task is performed in the upper cup of unmounted cartridges, you can use a negative number for the parameter value. In this case, the *well bottom* is the top of the cartridge cup, as shown in the following figure.  For example, a negative number enables a cup wash step while the cartridges are in the cartridge rack before they are mounted on the Bravo 96AM Head. **IMPORTANT** You can use a negative number for this parameter only if the AM Aspirate task is performed in cartridges that are located in a cartridge rack. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.

Parameter	Description
Dynamic tip extension (0–20 mm/µL)	The rate at which the Bravo 96AM Head moves during the AM Aspirate task. The software calculates the distance over which the tips will move without crashing. Use dynamic tip extension to prevent spills as the pipette tips displace the liquid. To move the tips: • *At the same rate as the volume change.* Calculate dynamic tip extension (DTE) as follows: DTE = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTE > 1/A • *Slower than the volume change.* DTE < 1/A The starting and ending positions can be calculated as follows: $(V_{aspirated} * DTE) + Distance_{well\ bottom}$ *Note:* Instead of a negative aspirated volume, the software automatically moves downward toward the well bottom with each aspirate action.
Well selection	The wells at which the AM Aspirate task occurs. Click in the parameter box, and then click the Browse button to select the wells in the Well Selection dialog box. Use this parameter only if the Bravo 96AM Head has fewer tips than the number of wells in the microplate, or if you are in single-row or single-column mode.
Pipette technique	The pipette location offset you want to use for the AM Aspirate task. The list of pipette techniques are defined in the Pipette Technique Editor.
Perform tip touch	The option to touch the pipette tip on one or more sides of the well.
Which sides to use for tip touch	The side or sides of the well to use during tip touch: North, South, East, West, North/South, West/East, West/East/South/North.
Tip touch retract distance (–20 to 50 mm)	The vertical distance for the pipette tips to rise before touching the sides of the wells.

Parameter	Description
Tip touch horizontal offset (–5 to 5 mm)	The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition. For example, if you set a value of: • 0, the tips move a horizontal distance equal to the well radius • > 0, the tips attempt to move past the well radius, which results in a more forceful tip touch • < 0, the tips move a distance less than the radius of the well, resulting in a lighter tip touch

Quadrant pattern well selection

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head.

IMPORTANT For the Bravo 96AM Head, the quadrant pattern option is available only if you are using 384-well microplates.

Instead of a column- or row-wise pattern, you can select a quadrant pattern during well selection.

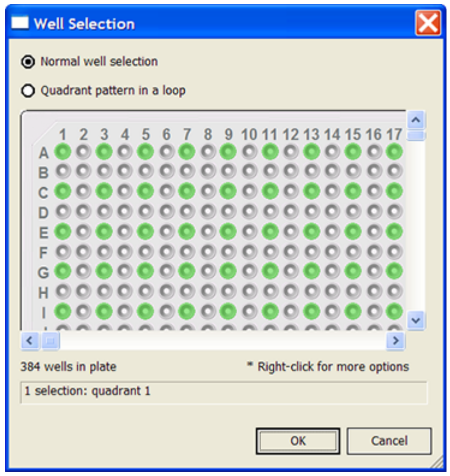
The quadrant pattern option is available only if:

- The number of channels in the pipette head (or pins in a pin tool) is fewer than the number of wells in the microplate. For example, you can use a 96-channel pipette head to dispense liquid into a 384-well microplate or 1536-well microplate.
- All the channels are selected in the Set Head Mode task when using a pipette head. (The Set Head Mode task is not an option when using a pin tool).
- The liquid-handling task is inside a loop.

IMPORTANT If you select a quadrant pattern, specifications in the Well Selection dialog box will override `task.Wellselection` values assigned in the Advanced Settings area.

To select a quadrant pattern:

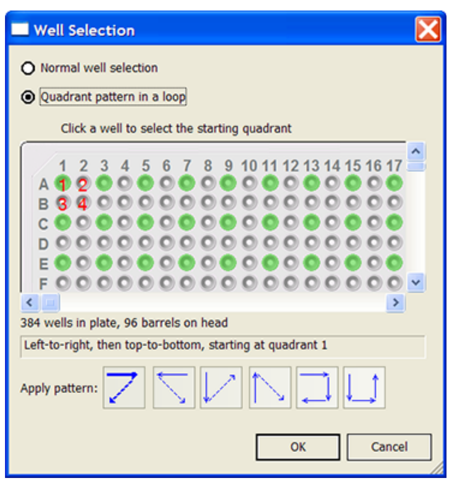
1 In the **Task Parameters** area, click the **Well selection** parameter box, and then click the Browse button. The Well Selection dialog box opens. By default, the **Normal well selection** option is selected. This option is used for column- and row-wise liquid-handling patterns.



2 Select **Quadrant pattern in a loop.** The contents of the dialog box change. Notice the following:

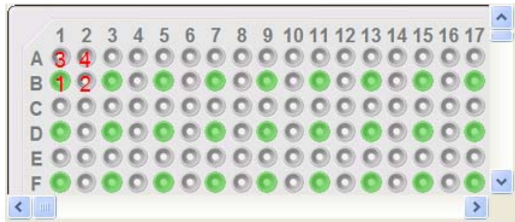
- Red numbers (1 through 4) appear on wells A1, A2, B1, and B2. The numbers indicate the pipetting sequence: 1 is the starting well, and 4 is the last well. In the following example, the sequence is A1, A2, B1, B2.
- Green wells indicate the starting well in the pipetting sequence.
- Pattern buttons at the bottom of the dialog box indicate the movement of the pipette channels. The movement description is provided in the text box above the buttons.

  Note: The last two patterns are unavailable if a group contains 16 wells. For example, the last two patterns are not available if you have a 96-well pipette head and a 1536-well microplate.



3 Select the starting well. The well becomes green and is labeled 1.

In the following example, the third quadrant (B1 well) is selected.



4 Click a pattern button to specify the pipette channel movement. After you click a pattern, the red numbers in the graphic are updated to show the sequence.

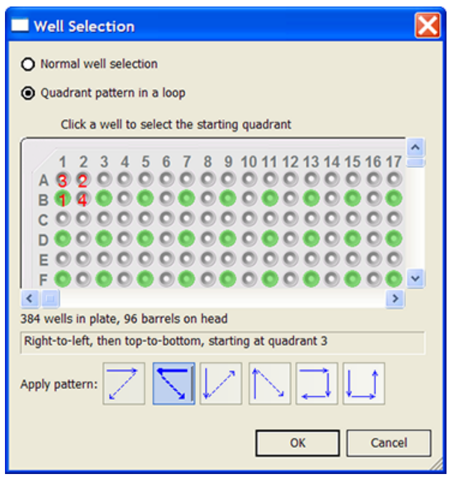
In the following example, the second pattern is selected (right-to-left, then top-to-bottom). The third quadrant (B1) is the starting well. The resulting movement is:

Quadrant 3 (B1)

Quadrant 2 (A2)

Quadrant 1 (A1)

Quadrant 4 (B2)



5 When you are finished, click **OK** to save the changes and return to the VWorks window.

Example: Allowing an operator to change the aspirate flow rate using an edit control on a VWorks form

Goal

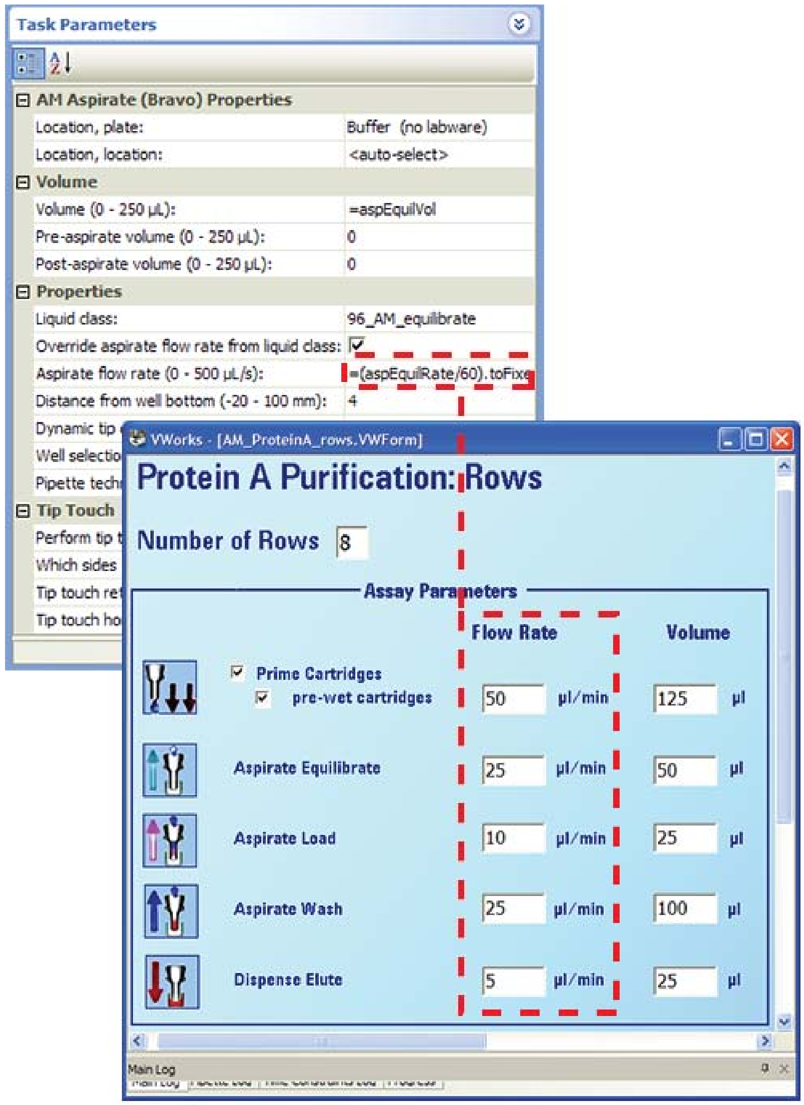
Allow an operator to change the aspirate flow rate easily from run to run without modifying the liquid class.

Implementation

In the protocol, the AM Aspirate task is added. In the task parameters, the **Override aspirate flow rate from liquid class** check box is selected, and the **Aspirate flow rate** is specified as a variable, for example `=AspFlowRate`.

In the VWorks form, an edit control is added for the aspirate flow rate using the =AspFlowRate variable. The operator can enter values within a specified range in the form and run the protocol using the form. For details on how to design VWorks forms to run protocols, see "Creating protocol forms for operators" on page 153.

Figure Flow rate parameter as JavaScript variable that appears as edit control in a VWorks form



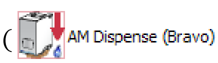
Related information

For information about...	See...
Using the Bravo 96AM Head	*Bravo Automated Liquid Handling Platform User Guide*
AM Dispense task	"AM Dispense (Bravo)" on page 410

For information about...	See...
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
AM Cartridges On	"AM Cartridges On (Bravo)" on page 431
AM Cartridges Off	"AM Cartridges Off (Bravo)" on page 429
Task macros	"Using macros to create protocols" on page 133
Protocol forms	"Creating protocol forms for operators" on page 153
Using JavaScript variables	"Using simple variables" on page 77

AM Dispense (Bravo)

Description

The AM Dispense (Bravo) (

) task is designed for dispensing fluids through AssayMAP Bravo cartridges that are mounted on the Bravo 96AM Head.

Task is available for...	Task is available in...
Bravo Platform with Bravo 96AM Head	Main Protocol, Bravo Subprocess

The AM Dispense (Bravo) task has the same task parameters as the Dispense (Bravo) task, except for the following parameters:

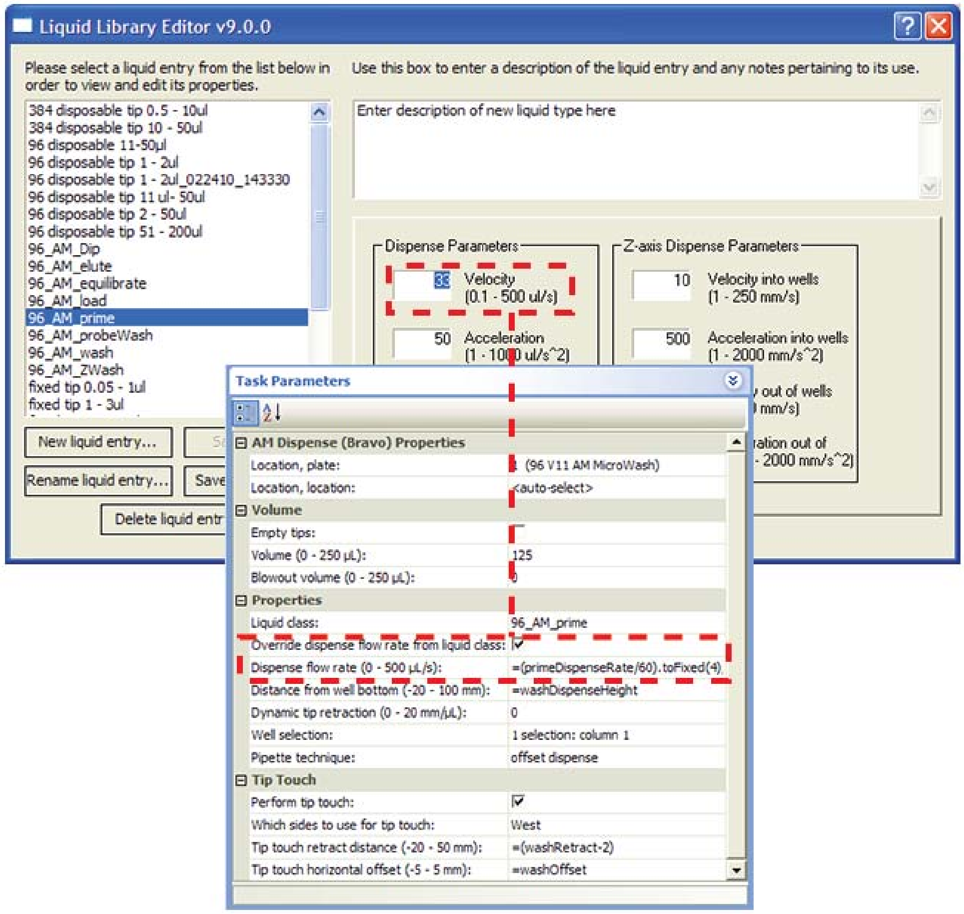
- Override dispense flow rate from liquid class

  Precise flow-rate control is critical for the performance of the AssayMAP Bravo cartridges. The option to override the dispense flow rate (velocity) in the liquid class enables an operator to change the flow rate from experiment to experiment for rapid method development without modifying the liquid class. Likewise, the flow rate can vary for different AM Dispense tasks that use the same liquid class in a given protocol.

- Distance from well bottom (-20–100 mm)

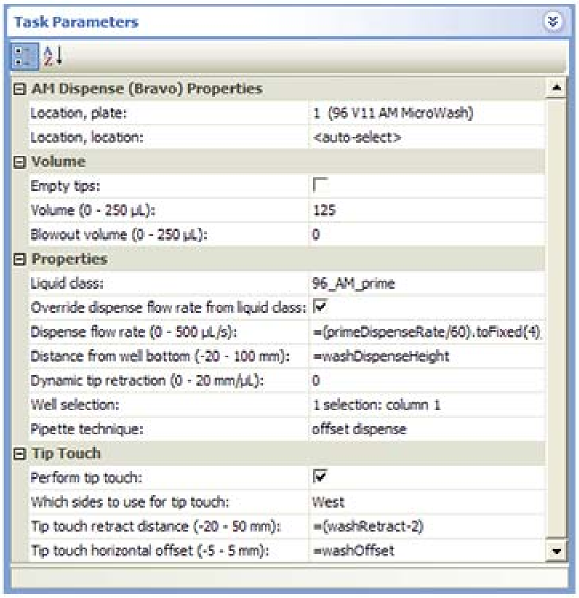
  The range includes negative values, which is critical for steps that require using the bare probes to dispense into the top portion of unmounted cartridges.

Figure Liquid class velocity setting in the Liquid Library Editor and flow rate override option



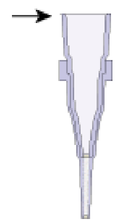
Task parameters

After adding the AM Dispense task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the AM Dispense task.
Location, location	The location at which the AM Dispense task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck, the software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task.
Empty tips	The option to empty all liquid from the probes, cartridges, or tips instead of using the dispense volume specification.
Volume (µL)	The volume of liquid to be dispensed from each probe, cartridge, or tip.
Blowout volume (µL)	Specifies the volume of air to dispense after the main volume has been dispensed while the probes, cartridges, or tips are still in the wells. Typically, the blowout volume is the same as the pre-aspirate volume. Note: Blowout occurs only in the last quadrant dispensed for a given AM Dispense task.

Parameter	Description
Liquid class	The pipetting speed and accuracy. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.
Override dispense flow rate from liquid class	The option to override the dispense velocity in the selected liquid class. Selecting this option enables you to specify a value for the dispense flow rate without changing the liquid class.
Dispense flow rate (0–500 µL/s)	The numerical value or the JavaScript variable that will override the dispense velocity setting in the liquid class. A JavaScript variable enables the value to be assigned later. For example, using a VWorks form, an operator could easily change the flow rate for an aspirate step in increments from as low as 1 µL/min up to 2000 µL/min or more using the same liquid class. If the task is included in a VWorks macro, a JavaScript variable enables you to change the value for the task at the macro level. **IMPORTANT** The software requires that the flow rate value be in microliters per second (µL/s) at run time. If you want an operator to enter the value in microliters per minute (µL/min), you can use scripting to convert the values for the software to use.

Parameter	Description
Distance from well bottom (-20–100 mm)	The distance between the tips of the probes, cartridges, or disposable tips and the *well bottoms* during the AM Dispense task. If you specify dynamic tip extension, this is the distance at the end of the AM Dispense task. Use a positive number for tasks that are performed with mounted cartridges, mounted tips, or bare probes at labware other than a cartridge rack. If the task is performed in the upper cup of unmounted cartridges, you can use a negative number for the parameter value. In this case, the *well bottom* is the top of the cartridge cup, as shown in the following figure.  For example, a negative number enables a prewetting step before mounting and priming the cartridges. **IMPORTANT** You can use a negative number for this parameter only if the AM Dispense task is performed in cartridges that are located in a cartridge rack. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.

Parameter	Description
Dynamic tip retraction (0–20 mm/µL)	The rate at which to raise the pipette head during the Dispense task. Use dynamic tip retraction to prevent spills as the pipette tips displace the liquid. To move the tips: • *At the same rate as the volume change.* Calculate dynamic tip retraction (DTR) as follows: DTR = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTR > 1/A • *Slower than the volume change.* DTR < 1/A The starting and ending positions can be calculated as follows: $(V_{dispensed} * DTR) + Distance_{well\ bottom}$
Well selection	The wells at which the AM Dispense task occurs. Click in the parameter box, and then click the Browse button to select the wells in the Well Selection dialog box. Use this parameter only if the pipette head has fewer tips than the number of wells in the microplate, or if you are in single-row or single-column mode.
Pipette technique	The pipette location offset you want to use for the AM Dispense task. The list of pipette techniques are defined in the Pipette Technique Editor.
Perform tip touch	The option to touch the probes, cartridges, or tips on one or more sides of the well.
Which sides to use for tip touch	The side or sides of the well to use during tip touch: North, South, East, West, North/South, West/East, West/East/South/North.
Tip touch retract distance (–20 to 50 mm)	The vertical distance for the probes, cartridges, or tips to rise before touching the sides of the wells.

Parameter	Description
Tip touch horizontal offset (–5 to 5 mm)	The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition. For example, if you set a value of: • 0, the tips move a horizontal distance equal to the well radius • > 0, the tips attempt to move past the well radius, which results in a more forceful tip touch • < 0, the tips move a distance less than the radius of the well, resulting in a lighter tip touch

Quadrant pattern well selection

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head.

IMPORTANT For the Bravo 96AM Head, the quadrant pattern option is available only if you are using 384-well microplates.

Instead of a column- or row-wise pattern, you can select a quadrant pattern during well selection.

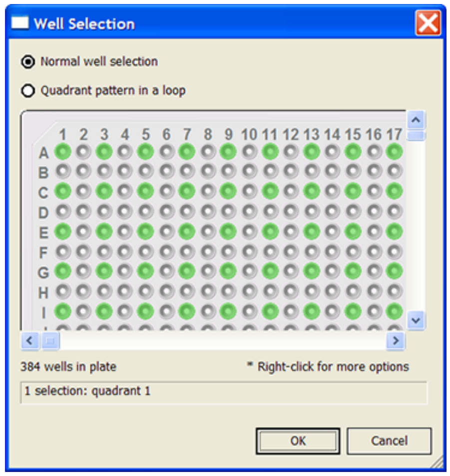
The quadrant pattern option is available only if:

- The number of channels in the pipette head (or pins in a pin tool) is fewer than the number of wells in the microplate. For example, you can use a 96-channel pipette head to dispense liquid into a 384-well microplate or 1536-well microplate.
- All the channels are selected in the Set Head Mode task when using a pipette head. (The Set Head Mode task is not an option when using a pin tool).
- The liquid-handling task is inside a loop.

IMPORTANT If you select a quadrant pattern, specifications in the Well Selection dialog box will override `task.Wellselection` values assigned in the Advanced Settings area.

To select a quadrant pattern:

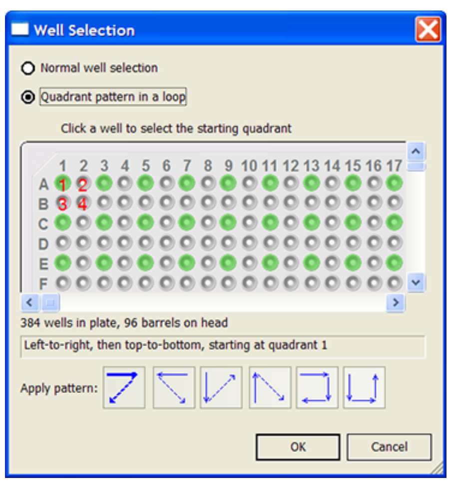
1 In the **Task Parameters** area, click the **Well selection** parameter box, and then click the Browse button. The Well Selection dialog box opens. By default, the **Normal well selection** option is selected. This option is used for column- and row-wise liquid-handling patterns.



2 Select **Quadrant pattern in a loop.** The contents of the dialog box change. Notice the following:

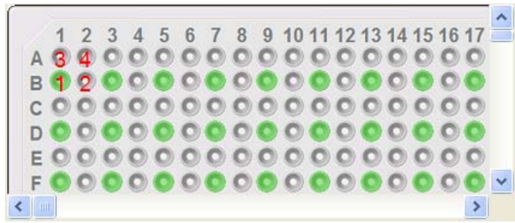
- Red numbers (1 through 4) appear on wells A1, A2, B1, and B2. The numbers indicate the pipetting sequence: 1 is the starting well, and 4 is the last well. In the following example, the sequence is A1, A2, B1, B2.
- Green wells indicate the starting well in the pipetting sequence.
- Pattern buttons at the bottom of the dialog box indicate the movement of the pipette channels. The movement description is provided in the text box above the buttons.

  Note: The last two patterns are unavailable if a group contains 16 wells. For example, the last two patterns are not available if you have a 96-well pipette head and a 1536-well microplate.



3 Select the starting well. The well becomes green and is labeled 1.

In the following example, the third quadrant (B1 well) is selected.



4 Click a pattern button to specify the pipette channel movement. After you click a pattern, the red numbers in the graphic are updated to show the sequence.

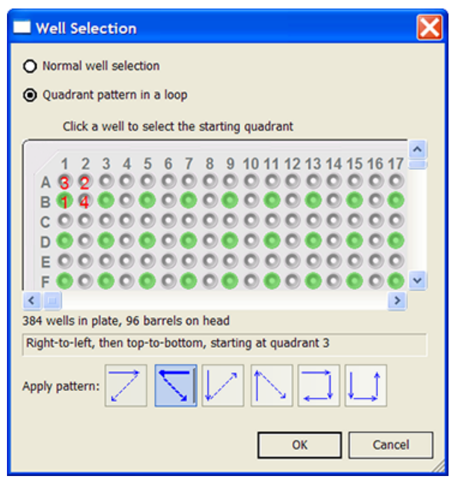
In the following example, the second pattern is selected (right-to-left, then top-to-bottom). The third quadrant (B1) is the starting well. The resulting movement is:

Quadrant 3 (B1)

Quadrant 2 (A2)

Quadrant 1 (A1)

Quadrant 4 (B2)



5 When you are finished, click **OK** to save the changes and return to the VWorks window.

Example: Allowing an operator to change the dispense flow rate using an edit control on a VWorks form

Goal

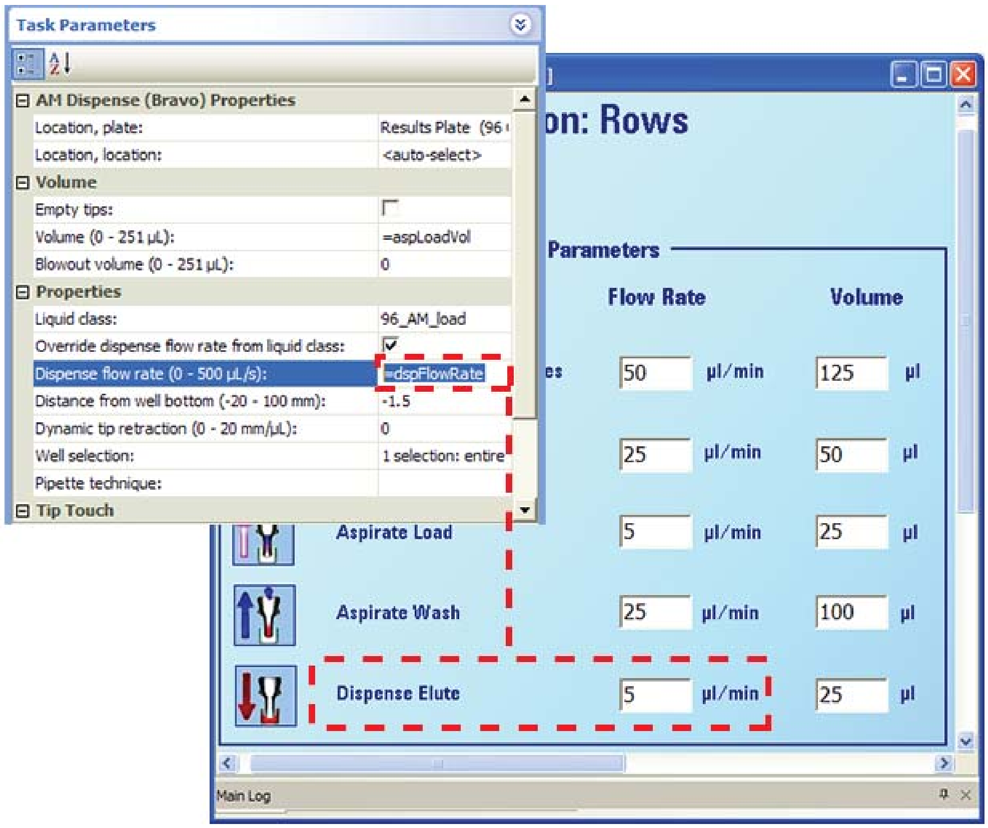
Allow an operator to change the dispense flow rate from run to run without modifying the liquid class.

Implementation

In the protocol, the AM Dispense task is added. In the task parameters, the **Override dispense flow rate from liquid class** check box is selected, and the **Dispense flow rate** is specified as a variable, for example `=dspFlowRate`.

In the VWorks form, an edit control is added for the dispense flow rate using the =dspFlowRate variable. The operator can enter values within a specified range in the form and run the protocol using the form. For details on how to design VWorks forms to run protocols, see "Creating protocol forms for operators" on page 153.

Figure Flow rate parameter as a variable that appears as an edit control in a VWorks form



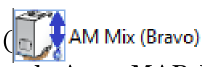
Related information

For information about...	See...
Using the Bravo 96AM Head	*Bravo Automated Liquid Handling Platform User Guide*
AM Aspirate task	"AM Aspirate (Bravo)" on page 400
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
AM Cartridges On	"AM Cartridges On (Bravo)" on page 431
AM Cartridges Off	"AM Cartridges Off (Bravo)" on page 429

For information about...	See...
Task macros	"Using macros to create protocols" on page 133
Protocol forms	"Creating protocol forms for operators" on page 153
Using JavaScript variables	"Using simple variables" on page 77

AM Mix (Bravo)

Description

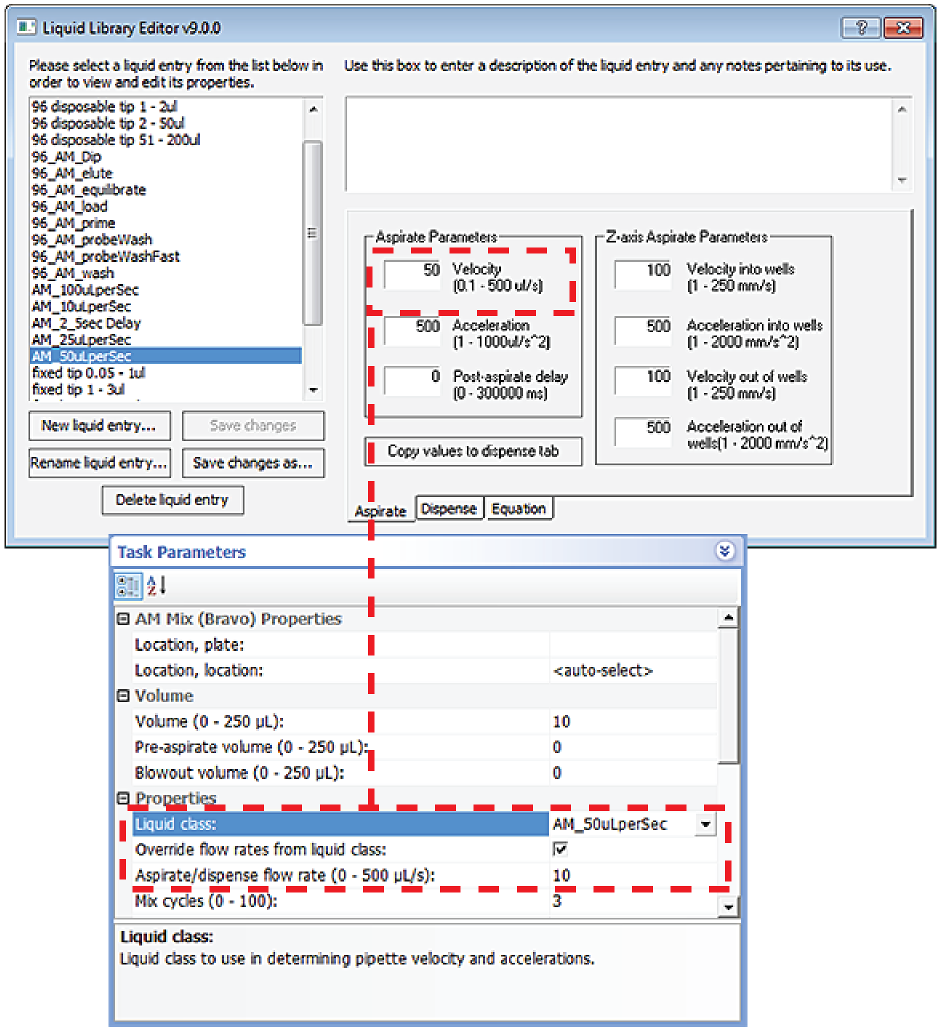
The AM Mix (Bravo) (

) task is designed for aspirating and dispensing fluids through AssayMAP Bravo cartridges that are mounted on the Bravo 96AM Head.

Task is available for...	Task is available in...
Bravo Platform with Bravo 96AM Head	Main Protocol, Bravo Subprocess

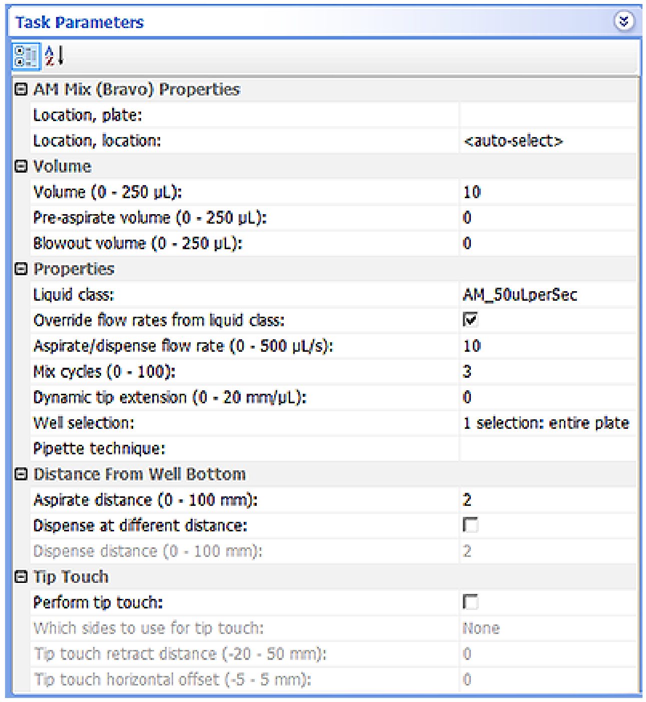
The AM Mix (Bravo) task has the same task parameters as the Mix (Bravo) task, plus an additional parameter that enables you to override the flow rate in the liquid class, if specified. Precise flow-rate control is critical for the performance of the AssayMAP Bravo cartridges. The option to override the aspirate and dispense flow rate (velocity) in the liquid class enables an operator to change the flow rate from experiment to experiment for rapid method development without modifying the liquid class. Likewise, the flow rate can vary for different AM Mix tasks that use the same liquid class in a given protocol.

Figure Liquid class velocity setting in the Liquid Library Editor and flow rate override option



Task parameters

After adding the AM Mix task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Mix task.
Location, location	The location at which the Mix task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Volume (µL)	The volume of liquid to be mixed in each well.
Pre-aspirate volume (µL)	The volume of air to be drawn before the pipette tips enter the liquid.
Blowout volume (µL)	The volume of air to dispense after the main volume has been dispensed while the tips are still in the wells. Typically, the blowout volume is the same as the pre-aspirate volume.
Liquid class	The pipetting velocity and accuracy. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.

Parameter	Description
Override flow rates from liquid class	The option to override the aspirate and dispense velocity in the selected liquid class. Selecting this option enables you to specify a value for the flow rate without changing the liquid class.
Aspirate/dispense flow rate (0–500 µL/s)	The numerical value or the JavaScript variable that will override both the aspirate and dispense velocity in the selected liquid class. A JavaScript variable enables the value to be assigned later. For example, using a VWorks form, an operator could easily change the flow rate for a mix step in increments from as low as 1 µL/min up to 2000 µL/min or more using the same liquid class. If the task is included in a VWorks macro, a JavaScript variable enables you to change the value for the task at the macro level. **IMPORTANT** The software requires that the flow rate value be in microliters per second (µL/s) at run time. If you want an operator to enter the value in microliters per minute (µL/min), you can use scripting to convert the values for the software to use.
Mix cycles (0–100)	The number of times to repeat the aspirate-and-dispense cycle.

Parameter	Description
Dynamic tip extension (0–20 mm/µL)	The rate at which the Bravo 96AM Head moves during the aspirate action. The software calculates the distance over which the tips will move without crashing. Use dynamic tip extension to prevent spills as the pipette tips displace the liquid. To move the tips: • *At the same rate as the volume change.* Calculate dynamic tip extension (DTE) as follows: DTE = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTE > 1/A • *Slower than the volume change.* DTE < 1/A The starting and ending positions can be calculated as follows: $(V_{aspirated} * DTE) + Distance_{well\ bottom}$ *Note:* Instead of a negative aspirated volume, the software automatically moves downward toward the well bottom with each aspirate action.
Well selection	The wells at which the Mix task occurs. Click in the parameter box, and then click the Browse button to select the wells in the Well Selection dialog box. Use this parameter only if the Bravo 96AM Head has fewer tips than the number of wells in the microplate, or if you are in single-row or single-column mode.
Pipette technique	The pipette location offset you want to use for the Mix task. The list of pipette techniques are defined in the Pipette Technique Editor.
Aspirate distance (0–100 mm)	The distance between the end of the pipette tips and the well botttoms during the aspirate action. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.

Parameter	Description
Dispense at different distance	The option to dispense at a pipette tip height that is different than the aspirate distance. Select the check box to enter a value for the dispense distance.
Dispense distance (0–100 mm)	The distance between the tips of the probes, cartridges, or disposable tips and the well bottoms during the dispense action.
Perform tip touch	The option to touch the tips on one or more sides of the well.
Which sides to use for tip touch	The side or sides of the well to use during tip touch: North, South, East, West, North/South, West/East, West/East/South/North.
Tip touch retract distance (–20 to 50 mm)	The vertical distance for the tips to rise before touching the sides of the wells.
Tip touch horizontal offset (–5 to 5 mm)	The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition. For example, if you set a value of: • 0, the tips move a horizontal distance equal to the well radius • > 0, the tips attempt to move past the well radius, which results in a more forceful tip touch • < 0, the tips move a distance less than the radius of the well, resulting in a lighter tip touch

Quadrant pattern well selection

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head.

IMPORTANT For the Bravo 96AM Head, the quadrant pattern option is available only if you are using 384-well microplates.

Instead of a column- or row-wise pattern, you can select a quadrant pattern during well selection.

The quadrant pattern option is available only if:

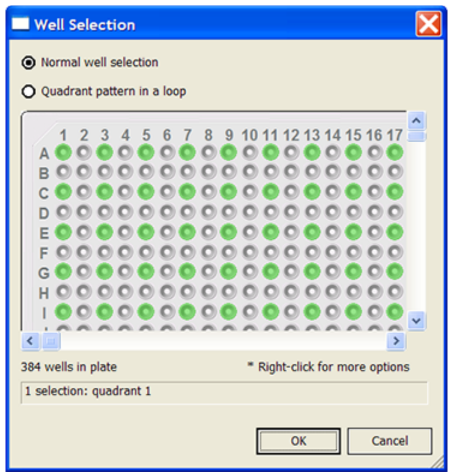
- The number of channels in the pipette head (or pins in a pin tool) is fewer than the number of wells in the microplate. For example, you can use a 96-channel pipette head to dispense liquid into a 384-well microplate or 1536-well microplate.
- All the channels are selected in the Set Head Mode task when using a pipette head. (The Set Head Mode task is not an option when using a pin tool).

- The liquid-handling task is inside a loop.

IMPORTANT If you select a quadrant pattern, specifications in the Well Selection dialog box will override `task.Wellselection` values assigned in the Advanced Settings area.

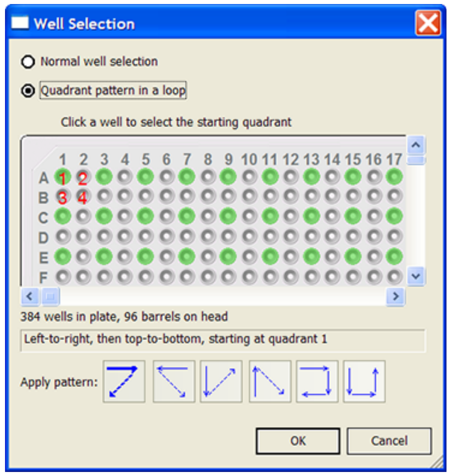
To select a quadrant pattern:

1. In the **Task Parameters** area, click the **Well selection** parameter box, and then click the Browse button. The Well Selection dialog box opens. By default, the **Normal well selection** option is selected. This option is used for column- and row-wise liquid-handling patterns.



2. Select **Quadrant pattern in a loop.** The contents of the dialog box change. Notice the following:
   - Red numbers (1 through 4) appear on wells A1, A2, B1, and B2. The numbers indicate the pipetting sequence: 1 is the starting well, and 4 is the last well. In the following example, the sequence is A1, A2, B1, B2.
   - Green wells indicate the starting well in the pipetting sequence.
   - Pattern buttons at the bottom of the dialog box indicate the movement of the pipette channels. The movement description is provided in the text box above the buttons.

     Note: The last two patterns are unavailable if a group contains 16 wells. For example, the last two patterns are not available if you have a 96-well pipette head and a 1536-well microplate.



3 Select the starting well. The well becomes green and is labeled 1.

In the following example, the third quadrant (B1 well) is selected.

4 Click a pattern button to specify the pipette channel movement. After you click a pattern, the red numbers in the graphic are updated to show the sequence.

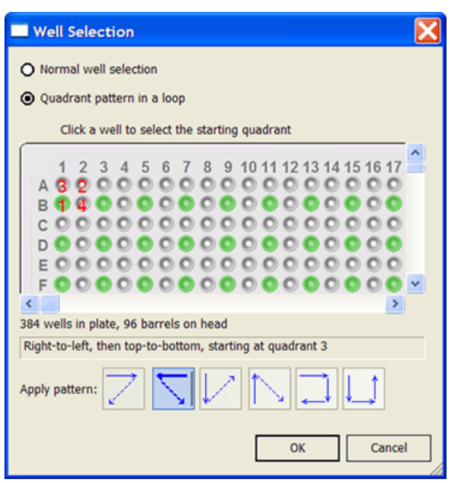
In the following example, the second pattern is selected (right-to-left, then top-to-bottom). The third quadrant (B1) is the starting well. The resulting movement is:

Quadrant 3 (B1)

Quadrant 2 (A2)

Quadrant 1 (A1)

Quadrant 4 (B2)



5 When you are finished, click **OK** to save the changes and return to the VWorks window.

Example: Allowing an operator to change the mix flow rate using an edit control on a VWorks form

Goal

Allow an operator to change the mix flow rate from run to run without modifying the liquid class.

Implementation

In the protocol, the AM Mix task is added. In the task parameters, the **Override dispense flow rates from liquid class** check box is selected, and the **Aspirate/dispense flow rate** is specified as a variable, for example `=mixFlowRate`.

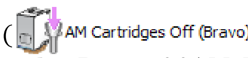
In the VWorks form, an edit control is added for the mix flow rate using the `=mixFlowRate` variable. The operator can enter values within a specified range in the form and run the protocol using the form. For details on how to design VWorks forms to run protocols, see "Creating protocol forms for operators" on page 153.

Related information

For information about...	See...
Using the Bravo 96AM Head	*Bravo Automated Liquid Handling Platform User Guide*
AM Aspirate task	"AM Aspirate (Bravo)" on page 400
AM Dispense task	"AM Dispense (Bravo)" on page 410
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
AM Cartridges On	"AM Cartridges On (Bravo)" on page 431
AM Cartridges Off	"AM Cartridges Off (Bravo)" on page 429
Task macros	"Using macros to create protocols" on page 133
Protocol forms	"Creating protocol forms for operators" on page 153
Using JavaScript variables	"Using simple variables" on page 77

AM Cartridges Off (Bravo)

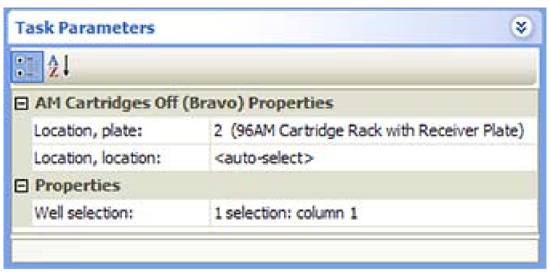
Description

The AM Cartridges Off (Bravo) () task removes the AssayMAP Bravo cartridges from the Bravo 96AM Head. The task is available only if the device profile specifies the Bravo 96AM Head. You can use the AM Cartridges Off task while the head's syringes contain liquid, and then subsequently dispense the liquid using the syringe probes.

Task is available for...	Task is available in...
Bravo Platform with a gripper and a Bravo 96AM Head installed	Main Protocol, Bravo Subprocess

Task parameters

After adding the AM Cartridges Off task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the AM Cartridges Off task. Ensure a cartridge rack is selected.
Location, location	The location at which the AM Cartridges Off task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck, the software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task.

Parameter	Description
Well selection	The well positions to use for the AM Cartridges Off task. To select fewer positions than a full head, a Set Head Mode task must precede the AM Cartridges Off task in the protocol. **IMPORTANT** See the *Bravo 96AM Head and AssayMAP Bravo Cartridges User Guide* for restrictions related to using a subset of channels in the Bravo 96AM Head.

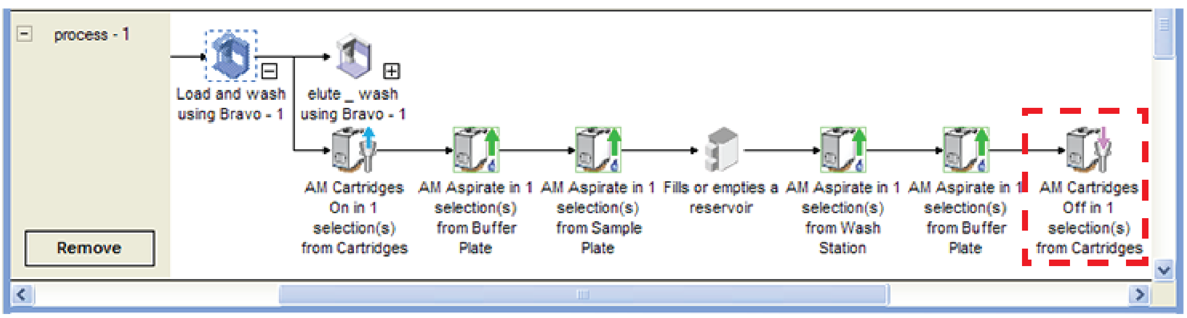
Example: Remove cartridges while the syringes contain liquid after a series of AM Aspirate tasks

Goal

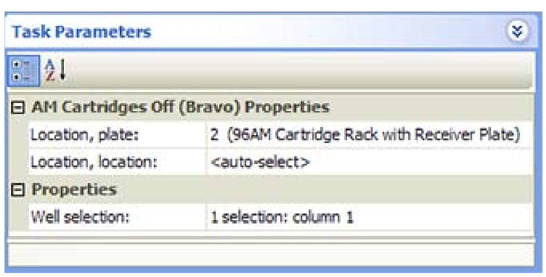
After aspirating a series of reagents through the AssayMAP Bravo cartridges to bind a target to the cartridge resin bed, remove the cartridges to wash the head's syringes before mounting the cartridges again for elution.

Implementation

In a liquid-handling subprocess, the AM Cartridges Off task is added after a series of AM Aspirate tasks and before a subsequent elute-wash subprocess.



Because the cartridges are removed at a cartridge rack, the Cartridge Rack is selected as the Location, plate in the Task Parameters area.



Related information

For information about...	See...
Using the Bravo 96AM Head	*Bravo Automated Liquid Handling Platform User Guide*
AM Cartridges On task	"AM Cartridges On (Bravo)" on page 431
AM Aspirate task	"AM Aspirate (Bravo)" on page 400
AM Dispense task	"AM Dispense (Bravo)" on page 410
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Set Head Mode task	"Set Head Mode (Bravo)" on page 529

AM Cartridges On (Bravo)

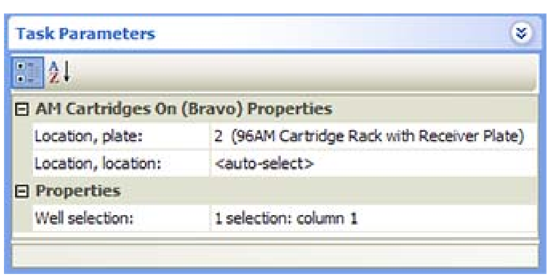
Description

The AM Cartridges On (Bravo) (AM Cartridges On (Bravo)) task presses cartridges on a Bravo 96AM Head. The task is available only if the profile specifies the Bravo 96AM Head. You can use the AM Cartridges On (Bravo) task while the head's syringes are empty or contain liquid.

Task is available for...	Task is available in...
Bravo Platform with a gripper and a Bravo 96AM Head installed	Main Protocol, Bravo Subprocess

Task parameters

After adding the AM Cartridges On task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the AM Cartridges On task. Ensure a cartridge rack is selected.
Location, location	The location at which the AM Cartridges On task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck, the software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task.
Well selection	The well positions to use for the AM Cartridges On task. To select fewer positions than a full head, a Set Head Mode task must precede the AM Cartridges On task. **IMPORTANT** See the *Bravo 96AM Head and AssayMAP Bravo Cartridges User Guide* for restrictions related to using a subset of channels in the Bravo 96AM Head.

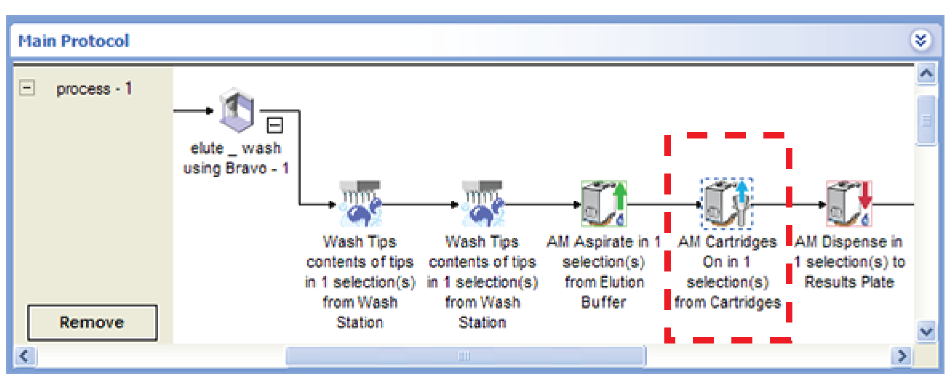
Example: Press on cartridges in order to perform a dispense through the cartridges

Goal

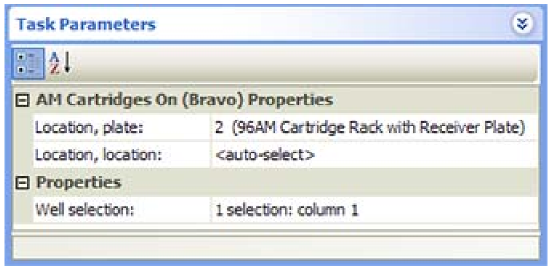
Press on the cartridges while fluid is held in the Bravo 96AM Head syringes so that the fluid can be dispensed through the cartridge resin bed for collection in a sample plate.

Implementation

In a liquid-handling subprocess, an AM Aspirate task is used to aspirate elution buffer using the bare probes into the Bravo 96AM Head syringes. The AssayMAP Bravo cartridges are pressed on using the AM Cartridges On task. An AM Dispense task is used to dispense the elution buffer through the cartridges into a results plate.



Because the cartridges are pressed on at a cartridge rack, the Cartridge Rack is selected for the Location, plate in the Task Parameters area.

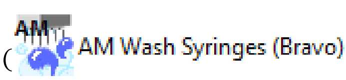


Related information

For information about...	See...
Using the Bravo 96AM Head	*Bravo Automated Liquid Handling Platform User Guide*
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
AM Cartridges Off task	"AM Cartridges Off (Bravo)" on page 429
AM Aspirate task	"AM Aspirate (Bravo)" on page 400
AM Dispense task	"AM Dispense (Bravo)" on page 410
Set Head Mode task	"Set Head Mode (Bravo)" on page 529

AM Wash Syringes (Bravo)

Description

The AM Wash Syringes (Bravo) (

) task enables the following types of washes using the Bravo 96AM Head:

- Internal washing of the Bravo 96AM Head syringes
- External washing of the syringe probes
- External washing of the AssayMAP Bravo cartridges while mounted on the probes

Task is available for...	Task is available in...
Bravo Platform with a Bravo 96AM Head and an autofilling station	Main Protocol, Bravo Subprocess

Requirements

In Bravo Diagnostics, ensure that the 96AM Wash Station is set up at the correct deck location and associated with a Pump Module.

Ensure that the protocol has a configured labware for the 96AM Wash Station in addition to the other processes and configured labware required by the protocol.

Task parameters

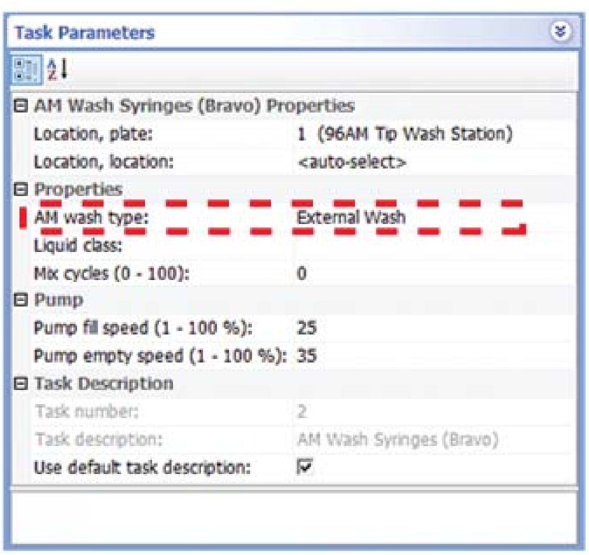
After adding the AM Wash Syringes task at the desired point in the protocol, you set the parameters in the **Task Parameters** area.

CAUTION Ensure that all the syringes have bare probes or that all syringes have mounted cartridges for this task. Otherwise, a potential collision can occur, which can damage the syringes and the 96AM Wash Station.

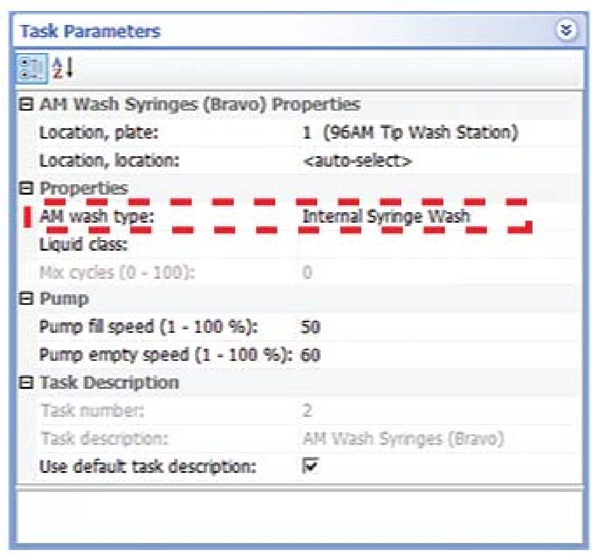
CAUTION Ensure that the labware definition for the 96AM Wash Station is accurate and the teachpoint for the location is precise so that the system can position the syringe probes or cartridge tips at the correct location. Otherwise, a potential collision can occur, which can damage the syringes and the 96AM Wash Station.

The software automatically selects all the chimneys in the wash station (entire plate) for the well selection and sets the volume and other properties. The following figure shows the **AM wash type** parameter: (**A**) **External Wash** or (**B**) **Internal Syringe Wash**.

A



B



The task actions vary depending on what you select for the **AM wash type** parameter:

- **External Wash**. Moves the tips of the probes or cartridges into the 96AM Wash Station chimneys. This wash type does not aspirate or dispense any volume. The liquid class can be used to control the duration of the wash.

Select the **External Wash** to wash the tips of the syringe probes or the cartridges mounted on the probes. See "Example External Wash" on page 436.

- **Internal Syringe Wash**. Dispenses to waste any contents remaining in the syringes, and then aspirates 250-µL wash liquid from the 96AM Wash Station chimneys into the syringes. The task dispenses the 250-µL volume to waste outside the chimneys, and then performs a tip touch on the chimneys' exterior.

  Select the **Internal Syringe Wash** to flush the syringe barrels with wash liquid. See "Example Internal Syringe Wash" on page 437.

  CAUTION Ensure that all syringe probes are bare (cartridges are off) for the Internal Syringe Wash. Otherwise, a potential collision can occur, resulting in equipment damage.

For each wash you select the following parameters.

Parameter	Description
Location, plate	The labware involved in the AM Wash Syringes task. Ensure that you use a 96AM Wash Station.
Location, location	The location at which the AM Wash Syringes task occurs. <auto-select> The software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task. This should be the deck location where the 96AM Wash Station is installed.
AM wash type	The type of wash: External Wash (default) or Internal Syringe Wash.
Liquid class	The liquid class for this wash liquid. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks. *Note:* Although the External Wash does not include aspirate and dispense actions, you can use a liquid class with a delay to control the duration of the wash.
Mix cycles (0–100)	*Internal Syringe Wash only.* The number of times to repeat the aspirate-and-dispense cycle for the wash task. Each mix cycle consists of one aspirate action and one dispense action.

Parameter	Description
Pump fill speed (%)	The speed, in percent of maximum, at which liquid flows into the 96AM Wash Station inlet ports and up into the chimneys. The Pump fill speed should be fast enough for the wash liquid to just bubble over the tops of the chimneys. Initially, you should use the following values, and then determine if you need to make adjustments: • **External Wash** – Pump fill speed 25% – Pump empty speed 35% • **Internal Syringe Wash** – Pump fill speed 50% – Pump empty speed 60%
Pump empty speed (%)	The speed, in percent of maximum, at which the waste liquid is pumped out of the 96AM Wash Station outlet ports. You should use the default values initially, and then determine if you need to adjust the value. **CAUTION** The Pump empty speed should be slightly faster than the Pump fill speed to prevent an overflow.
Use default task description	The option to change the label that appears with the task icon in the Main Protocol area. • To use the default text, ensure that the check box is selected. • To create your own label, clear the check box, and then type the desired text in the field next to **Task description**.

Example External Wash

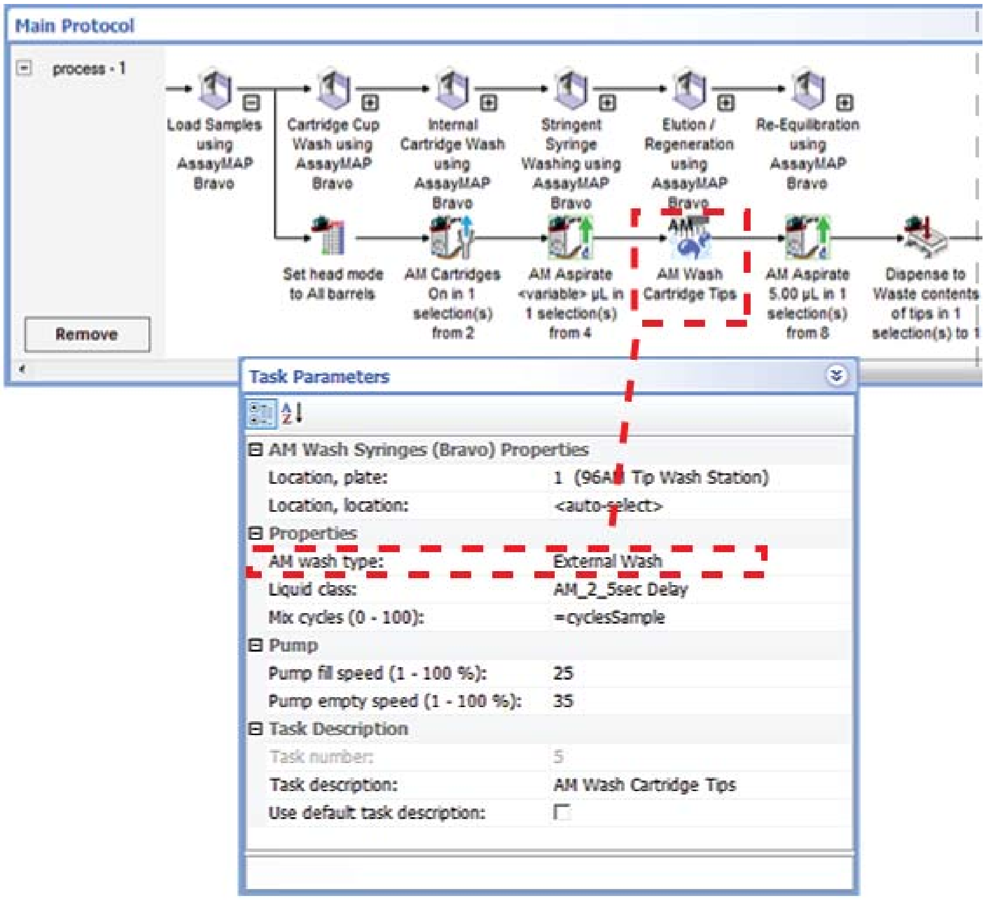
Goal

After aspirating samples or other reagents through the AssayMAP cartridges, perform an External Wash to remove the residual liquid that can cling to the external surfaces of the cartridges and prevent carryover to future steps.

Implementation

In the following example, the configured wash station at deck location 1 is called 96AM Tip Wash Station. In the Load Samples subprocess, an External Wash (AM Wash Cartridge Tips) task is added between an AM Aspirate task to load the sample on the mounted cartridges and an AM Aspirate task that aspirates 5 µL buffer as a sample chase. In the External Wash parameter settings, the liquid class specifies a delay that defines the time duration that the tips remain in the chimneys.

Figure External Wash example in an AssayMAP subprocess to load samples



Example Internal Syringe Wash

Goal

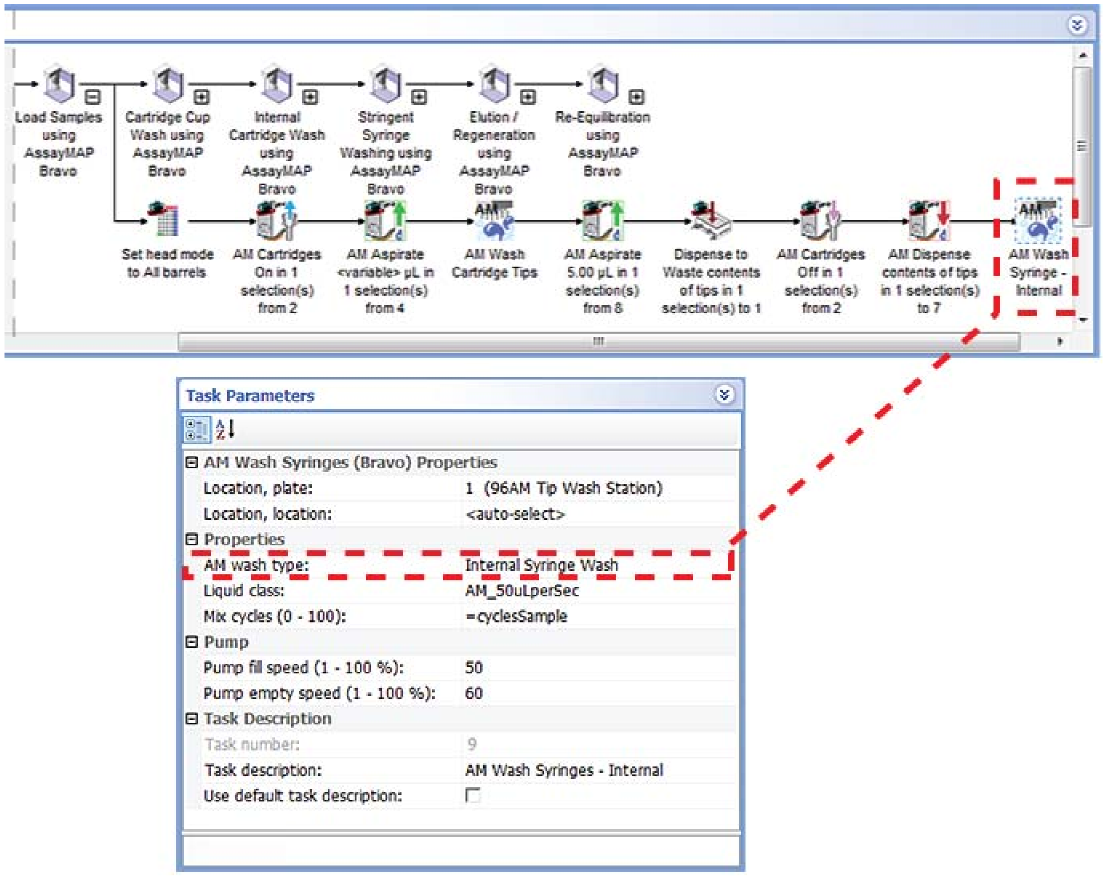
After aspirating samples or other reagents through the AssayMAP cartridges and into the syringes, eject the cartridges, and perform an Internal Syringe Wash to remove unwanted residual reagents from the syringes.

Implementation

In the following example, the configured wash station at deck location 1 is called 96AM Tip Wash Station. In the Load Samples subprocess, an Internal Syringe Wash task is added at the end of the subprocess after the cartridges are removed at deck location 2 and the syringe contents have been dispensed for collection at deck location 7.

Note: The Internal Syringe Wash dispenses to waste any contents remaining in the syringes when the task begins.

Figure Internal Syringe Wash example in an AssayMAP subprocess to load samples



Related information

For information about...	See...
Using the Bravo 96AM Head	*Bravo Automated Liquid Handling Platform User Guide*
AM Aspirate task	“AM Aspirate (Bravo)” on page 400
AM Dispense task	“AM Dispense (Bravo)” on page 410
Aspirate task	“Aspirate (Bravo, Vertical Pipetting Station)” on page 392
Dispense task	“Dispense (Bravo, Vertical Pipetting Station)” on page 451
AM Cartridges On	“AM Cartridges On (Bravo)” on page 431
AM Cartridges Off	“AM Cartridges Off (Bravo)” on page 429
Task macros	“Using macros to create protocols” on page 133

For information about...	See...
Protocol forms	"Creating protocol forms for operators" on page 153
Using JavaScript variables	"Using simple variables" on page 77

Assemble Vacuum (Bravo)

Description

The Assemble Vacuum task (Assemble Vacuum (Bravo)) directs the robot to pick up the Vacuum Filtration Station components from designated deck locations and stack them in the order you specify.

The task is used in conjunction with the following tasks:

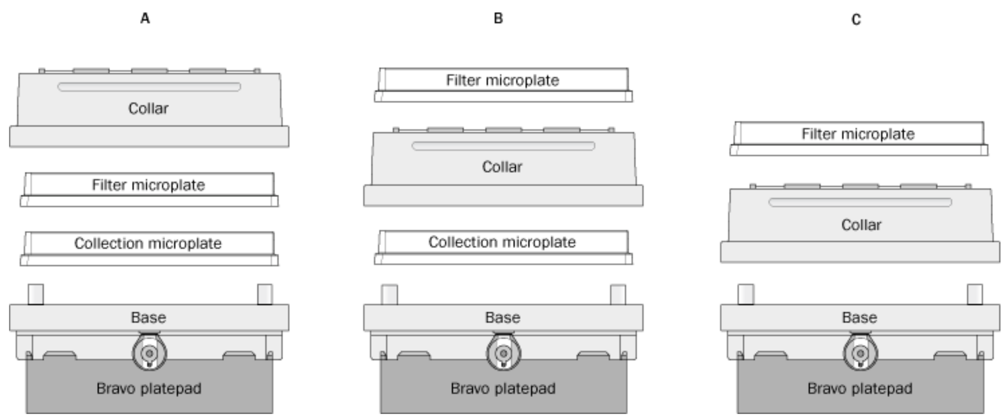
- Disassemble Vacuum
- Move and Filter Plate, or Toggle Vacuum

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

Supported configurations

To accommodate different assay types, the Bravo Platform supports the Vacuum Filtration Station configurations shown in the diagram. Note the following:

- *Configuration A*. The filter microplate is part of the station assembly. The assembly process can be automated during the protocol run. The filtrate in the collection microplate is retained.
- *Configuration B*. The filter microplate is not part of the station assembly. The assembly process can be automated during the protocol run. The robot will move the filter microplate to the station after the assembly process is finished. The filtrate in the collection microplate is retained.
- *Configuration C*. The basic configuration that must be assembled before a protocol run (assembly is not automated, so the Assembly and Disassembly tasks are not used). The filter microplate is not part of the station assembly. The robot will move the filter microplate to the station during the protocol run. The filtrate is discarded.



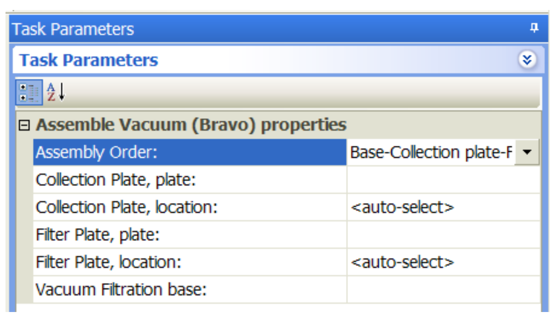
Requirements

To use the Assemble Vacuum (Bravo) task, you must first:

- Configure the Vacuum Filtration Station in Bravo Diagnostics.
- Set the robot gripper offset for labware that will be placed on the Vacuum Filtration Station during a protocol run. You set the offset in the Labware Editor.

Task parameters

After adding the Assemble Vacuum (Bravo) task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Assembly order	The order, from bottom to top, in which you want to stack the station components. The selections are: • Base-Collection plate-Filter plate-Collar (configuration A) • Base-Collection plate-Collar (configuration B) • Base-Collar (configuration C)
Collection Plate, plate	The type of microplate for collecting the filtrate.

Parameter	Description
Collection Plate, location	The location of the collection microplate. Select the deck location of the collection microplate. Select <auto-select> only if the software knows of the collection microplate location (for example, if you used a Place Plate task earlier in the protocol to specify its location).
Filter Plate, plate	The type of microplate used for the filtering process.
Filter Plate, location	The location of the filtering microplate. Select the deck location of the filter microplate. Select <auto-select> only if the software knows of the filter microplate location (for example, if you used a Place Plate task earlier in the protocol to specify its location).
Vacuum Filtration base	The location of the Vacuum Filtration Station base.

Examples:

See the example in:

- "Move and Filter Plate (Bravo)" on page 505
- "Toggle Vacuum (Bravo, Vertical Pipetting Station)" on page 545

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • *Bravo Automated Liquid Handling Platform User Guide*
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Configuring the Vacuum Filtration Station	*Bravo Automated Liquid Handling Platform User Guide*
Setting the robot gripper offset in the Labware Editor	*VWorks Automation Control Setup Guide*
Move and Filter Plate task	"Move and Filter Plate (Bravo)" on page 505
Toggle task	"Toggle Vacuum (Bravo, Vertical Pipetting Station)" on page 545
Disassemble Vacuum (Bravo) task	"Disassemble Vacuum (Bravo)" on page 449

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Dilute to Final Volume (Bravo)

Description

The Dilute to Final Volume task (Dilute To Final Volume (Bravo)) allows you to transfer liquid from a reagent labware to a destination microplate. The task is available only if you specified dilution series parameters within the Format Wizard in the Hit Pick Replication task.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

You can add the Dilute to Final Volume task before or after the Hit Pick Replication task.

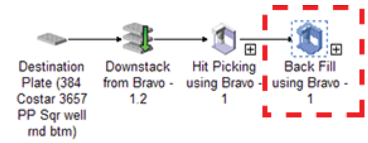
- *Before hit-picking*. Prefills the destination microplate with reagent. If the destination microplates are empty when the task starts, you only need to change the pipette tip with every destination microplate. You do not have to change the pipette tip between each reagent well, thus reducing the number of tips used.
- *After hit-picking*. Backfills the destination microplate with reagent. In this case, you must change the pipette tip between each reagent well to prevent contamination, thus using more tips.

Adding the Dilute to Final Volume task

To add the Dilute to Final Volume task:

1. Add another Bravo Subprocess in the destination microplate process. (You cannot add the Dilute to Final Volume task in the subprocess that contains the Hit Pick Replication task.)

   In the following example, the first Bravo subprocess (called Hit Picking) contains the Hit Pick Replication task. The second Bravo subprocess, Back Fill, contains the Dilute to Final Volume task. Notice that both subprocesses are in the Destination Plate process.



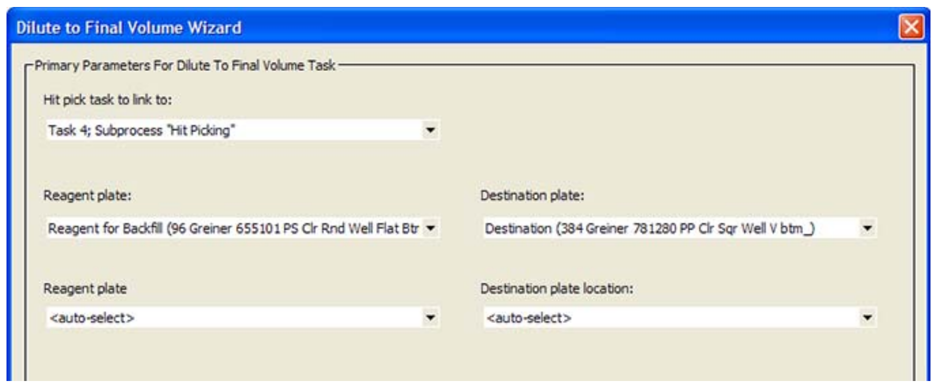
When you add the Dilute to Final Volume (Bravo) task, the software automatically starts the Dilute to Final Volume Wizard.

2 On the first page of the wizard:

a Select the source and destination labware to use.

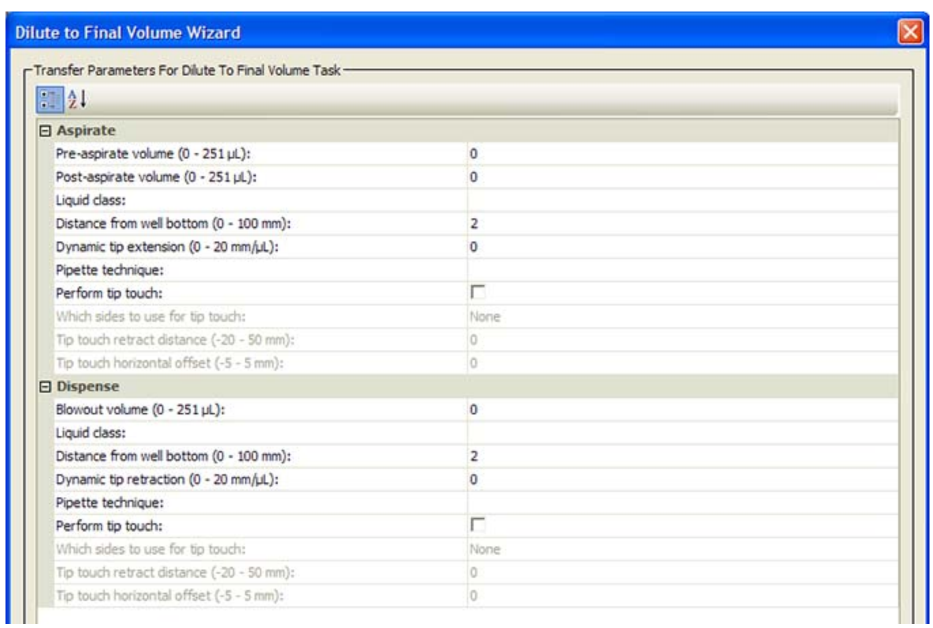
Note: The source labware is the labware that contains the reagent for back filling. It is not the source microplate used for the Hit Pick Replication task.

b Specify the locations of the reagent and destination labware.



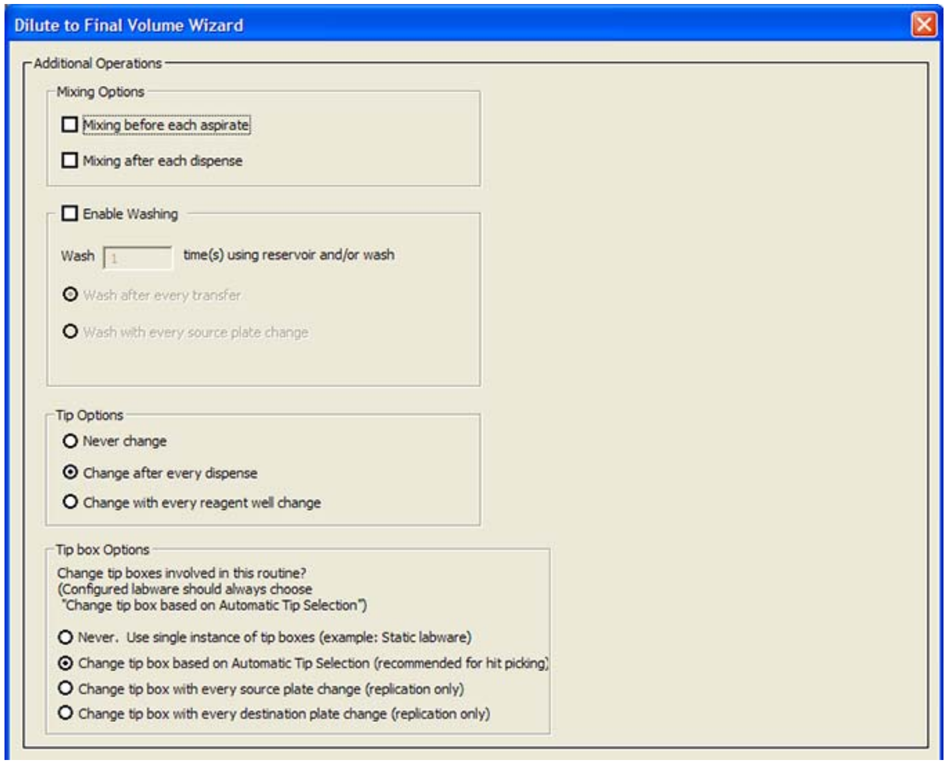
When you are finished, click **Next.**

3 Specify the Aspirate and Dispense task parameters. For detailed parameter descriptions, see "Aspirate (Bravo, Vertical Pipetting Station)" on page 392 and "Dispense (Bravo, Vertical Pipetting Station)" on page 451.



When you are finished, click **Next.**

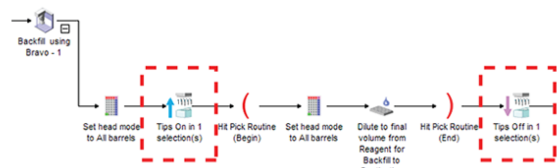
4 Select additional operations, if applicable.



Additional operation	Description
Mixing Options	Select where you want to add a Mix task: • Before each aspirate task • After each dispense task You will need to set the Mix parameters later in the wizard. If you do not want to mix, clear both check boxes.
Enable Washing	Select the check box if you want to add one or more Wash Tips tasks. You can specify the number of tip washes and whether you want to: • Wash after every transfer • Wash with every source microplate change You will need to set the Wash Tips parameters later in the wizard. Clear the check box if you do not want to add any Wash Tips tasks.

<table>
<tr><th>Additional operation</th><th>Description</th></tr>
<tr><td>Tip Options</td><td>Select the tip-change option:

Never change. If you select option, you must add the Set Head Mode, Tips On, and Tips Off tasks in the subprocess, outside of the Dilute to Final Volume routine, after you are finished with the wizard.



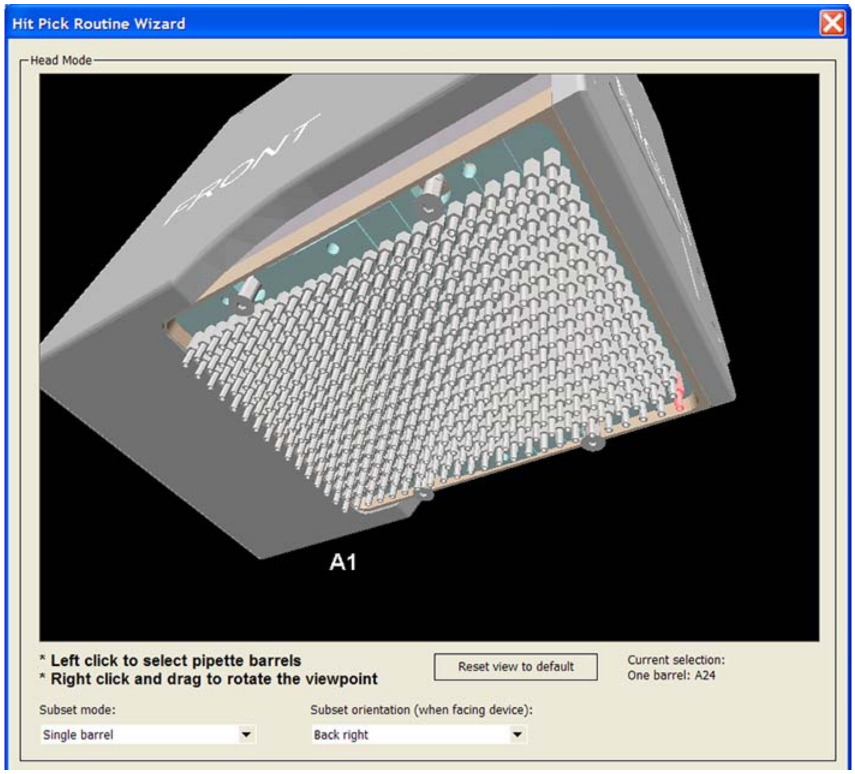
Change after every dispense. Changes the tips after every dispense within the routine.
Change with every reagent well change. Changes the tips after every reagent well change during the routine.
</td></tr>
<tr><td>Tip Box Options</td><td>Select when to change the tipbox:

Never. Use single instance of tip boxes (example: Static labware). Does not change the tipbox within the routine.
Change tip box based on Automatic Tip Selection. Uses tip-tracking within the routine, and changes the tipbox when the box if fully used. Select this option if the tipboxes you are using are configured labware.

Note: Tip-tracking continues in subsequent tasks in the process.
Change tipbox with every source plate change (replication only). Changes the tipbox every time the source microplate changes. This option is available only when a replication format file is selected.
Change tipbox with every destination plate change (replication only). Changes the tipbox every time the destination microplate changes. This option is available only when a replication format file is selected.
</td></tr>
</table>

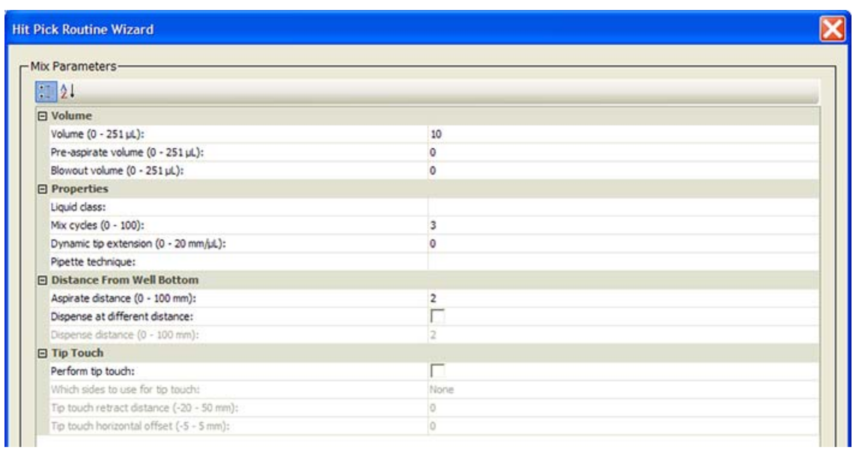
When you are finished, click **Next.**

5 Select the pipette channels to use. For a description of the selections, see "Set Head Mode (Bravo)" on page 529.



When you are finished, click **Next.**

6 *If you selected mixing options in step 4*. Set the mixing parameters. For a description of the parameters, see “Mix (Bravo, Vertical Pipetting Station)” on page 497.



When you are finished, click **Next.**

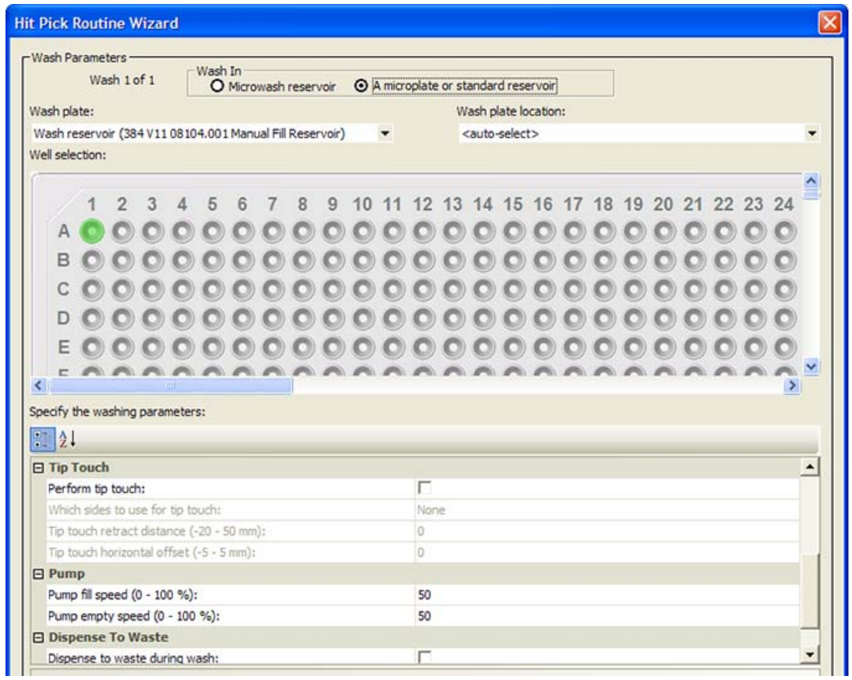
7 *If you selected wash options in step 4*. Set the wash parameters.

a In the **Wash In** area, select the type of labware to be used for the wash task:

- *Tip Wash Station (MicroWash Reservoir)*. Adds Wash Tips tasks and will use either configured or static labware.
- *A microplate or standard reservoir*. Adds Mix tasks.

b In the **Wash plate** list, select the desired labware and location.

c Set wash-tip parameters. For a description of the parameters, see "Wash Tips (Bravo, Vertical Pipetting Station)" on page 550.



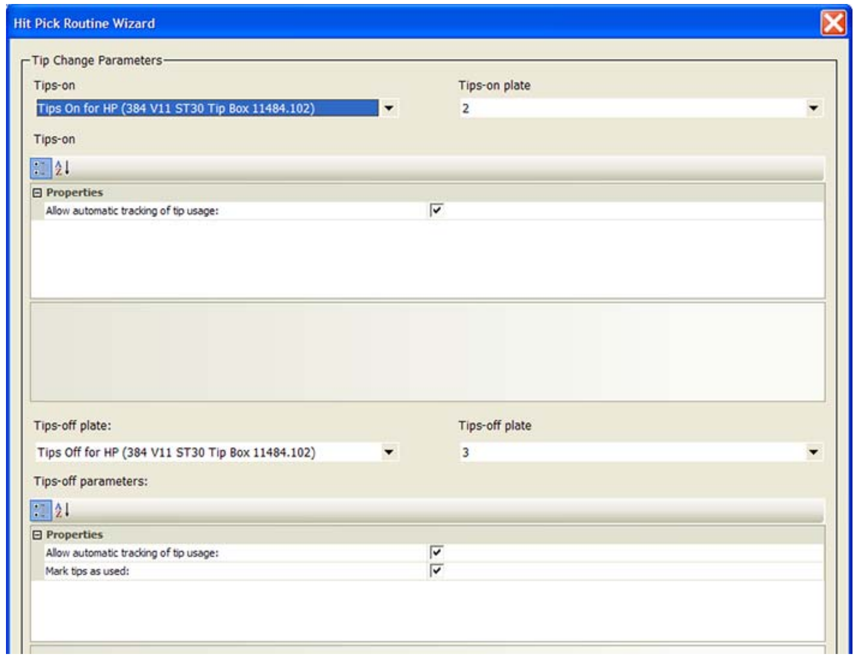
When you are finished, click **Next.**

8 *If you selected a tip-change option in step 4*. Select the tipbox options.

a Select the tipboxes you want to use.

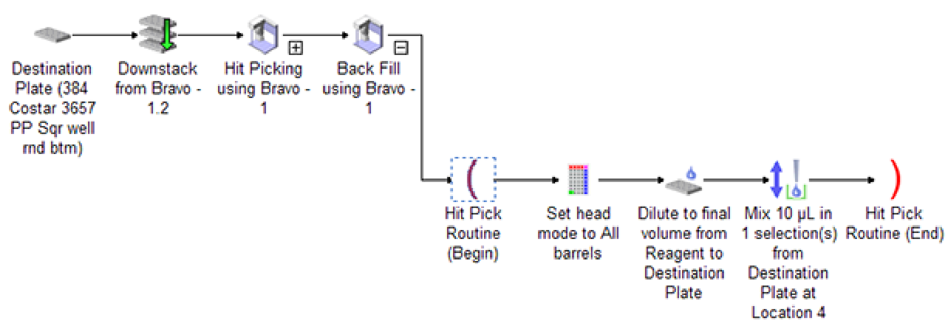
b Select the tip-tracking options you want to use:

- *Allow automatic tracking of tip usage*. Allows the software to track pipette tip usage during the protocol run or across different protocol runs. If you select the option in this task, you must also select the option in the Tips Off task. In general, use the default selections displayed.
- *Mark tips as used*. The option to use only new pipette tips during the protocol run. Select the option so that the software counts the number of tips used during the protocol run. The tips that have been used once are marked as used so that they cannot be reused. Clear the check box so that during the next Tips On task, the same tips can be reused. In general, use the default selections displayed.



9 Click **Finish.**

Based on the information you provide in the wizard, the software adds the Aspirate, Dispense, Mix, tip-washing, and other tasks that are necessary to produce the desired results.



Notice the following:

- The Hit Pick Routine (Begin) and Hit Pick Routine (End) tasks mark the beginning and end of the dilute-to-final-volume tasks.
- You can modify any of the task parameters directly in the Task Parameters area.
- You can return to the Dilute to Final Volume Wizard. To do this, select either the Hit Pick Routine (Begin) or Hit Pick Routine (End) task. In the Task Parameters area, click **Launch hit pick routine wizard.**

Related information

For information about...	See...
Adding tasks in a protocol	“Adding and deleting tasks” on page 53

For information about...	See...
Hit Pick Replication task	"Hit Pick Replication (Bravo)" on page 466
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Disassemble Vacuum (Bravo)

Description

The Disassemble Vacuum task (Disassemble Vacuum (Bravo)) directs the robot to remove components from the Vacuum Filtration Station and place them back at the locations specified in the Assemble Vacuum task. If a filter microplate was placed on top of the station, the Disassemble Vacuum task directs the robot to move the microplate to a specified location before disassembly.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

The task is used in conjunction with the following tasks:

- Assemble Vacuum
- Move and Filter Plate, or Toggle Vacuum

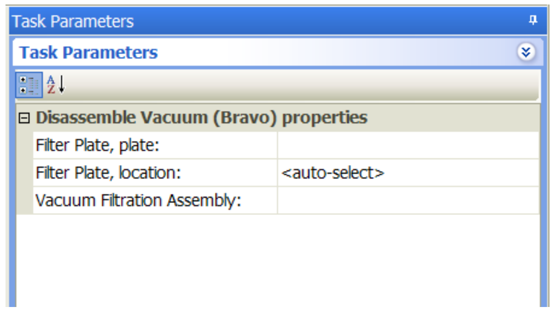
Requirements

To use the Disassemble Vacuum (Bravo) task, you must first:

- Configure the Vacuum Filtration Station in Bravo Diagnostics.
- Set the robot gripper offset for labware that will be placed on the Vacuum Filtration Station during a protocol run. You set the offset in the Labware Editor.
- Add an Assemble Vacuum (Bravo) task.

Task parameters

After adding the Disassemble Vacuum (Bravo) task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Filter Plate, plate	The type of microplate for collecting the filtrate.
Filter Plate, location	The location at which you want to place the filter microplate during the disassembly process. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Vacuum Filtration Assembly	The location of the Vacuum Filtration Station.

Examples:

See the example in:

- "Move and Filter Plate (Bravo)" on page 505
- "Toggle Vacuum (Bravo, Vertical Pipetting Station)" on page 545

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Configuring the Vacuum Filtration Station	*Bravo Automated Liquid Handling Platform User Guide*

For information about...	See...
Setting the robot gripper offset in the Labware Editor	*VWorks Automation Control Setup Guide*
Move and Filter Plate	“Move and Filter Plate (Bravo)” on page 505
Toggle	“Toggle Vacuum (Bravo, Vertical Pipetting Station)” on page 545
Assemble Vacuum (Bravo) task	“Assemble Vacuum (Bravo)” on page 439
Microplate-handling tasks	“Setting parameters for microplate-handling tasks” on page 277
Microplate-storage tasks	“Setting parameters for microplate storage tasks” on page 349
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557

Dispense (Bravo, Vertical Pipetting Station)

Description

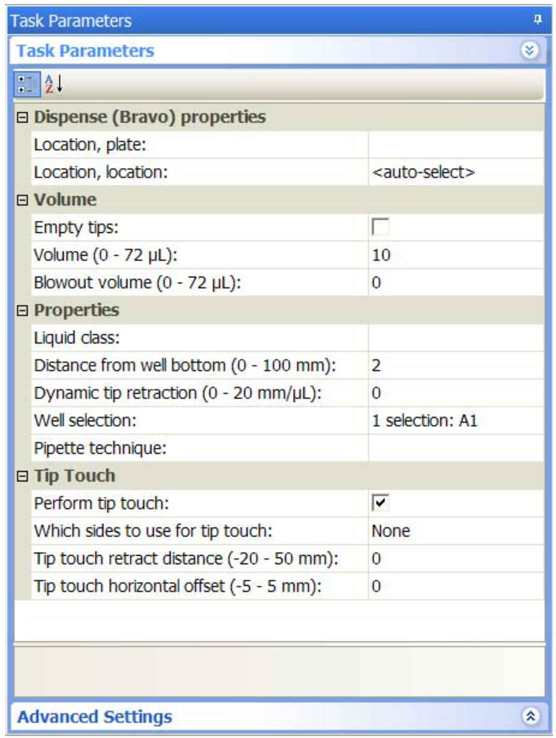
The Dispense (Bravo) (Dispense (Bravo)) and Dispense (Vertical Pipetting Station) (Dispense (Vertical Pipetting Station)) tasks dispense liquid into a microplate, reservoir, or tubes in a tube rack.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Task parameters

Note: The task parameters for Dispense (Bravo) and Dispense (Vertical Pipetting Station) are identical.

After adding the Dispense task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Dispense task.
Location, location	The location at which the Dispense task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Empty tips	The option to empty all liquid from the tips instead of using the dispense volume specification.
Volume (µL)	The volume of liquid to be dispensed from each pipette tip.
Blowout volume (µL)	Specifies the volume of air to dispense after the main volume has been dispensed while the tips are still in the wells. Typically, the blowout volume is the same as the pre-aspirate volume. Note: Blowout only occurs in the last quadrant dispensed for a given Dispense task.

Parameter	Description
Liquid class	The pipetting speed and accuracy. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.
Distance from well bottom (0–100 mm)	The distance between the end of the pipette tips and the well bottoms during the Dispense task. If you specify dynamic tip retraction, this is the starting distance. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.
Dynamic tip retraction (0–20 mm/µL)	The rate at which to raise the pipette head during the Dispense task. Use dynamic tip retraction to prevent spills as the pipette tips displace the liquid. To move the tips: • *At the same rate as the volume change.* Calculate dynamic tip retraction (DTR) as follows: DTR = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTR > 1/A • *Slower than the volume change.* DTR < 1/A The starting and ending positions can be calculated as follows: $(V_{dispensed} * DTR) + Distance_{well\ bottom}$
Well selection	The wells at which the Dispense task occurs. Click in the parameter box, and then click the Browse button to select the wells in the Well Selection dialog box. Use this parameter only if the pipette head has fewer tips than the number of wells in the microplate, or if you are in single-row or single-column mode.
Pipette technique	The pipette location offset you want to use for the Dispense task. The list of pipette techniques are defined in the Pipette Technique Editor.
Perform tip touch	The option to touch the pipette tip on one or more sides of the well.

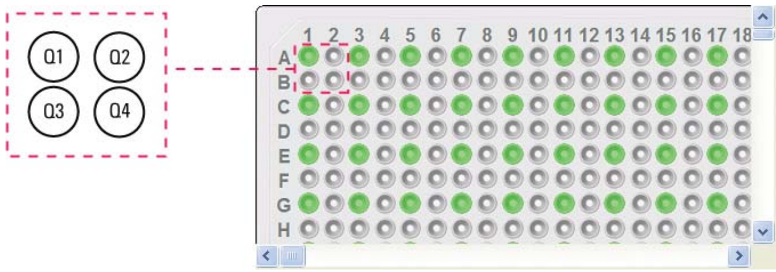
Parameter	Description
Which sides to use for tip touch	The side or sides of the well to use during tip touch: North, South, East, West, North/South, West/East, West/East/South/North.
Tip touch retract distance (–20 to 50 mm)	The vertical distance for the pipette tips to rise before touching the sides of the wells.
Tip touch horizontal offset (–5 to 5 mm)	The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition. For example, if you set a value of: • 0, the tips move a horizontal distance equal to the well radius • > 0, the tips attempt to move past the well radius, which results in a more forceful tip touch • < 0, the tips move a distance less than the radius of the well, resulting in a lighter tip touch

Quadrant pattern well selection

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head. The following table lists the types of pipette heads and the number of accessible quadrants in various microplates.

Pipette head channels/ pin tool pins	Microplate	Number of quadrants
96	96-well	1
	384-well	4
	1536-well	16
384	384-well	1
	1536-well	4
1536 (pin tool only)	1536-well	1

The following diagram demonstrates the concept of quadrants. The diagram shows a portion of a 384-well microplate and highlights the four quadrants (Q1, Q2, Q3, and Q4) that are accessible by the A1 tip of a 96-channel pipette head. Notice that the green color highlights all of the quadrant 1 (Q1) wells across the microplate.



Instead of a column- or row-wise pattern, you can select a quadrant pattern during well selection.

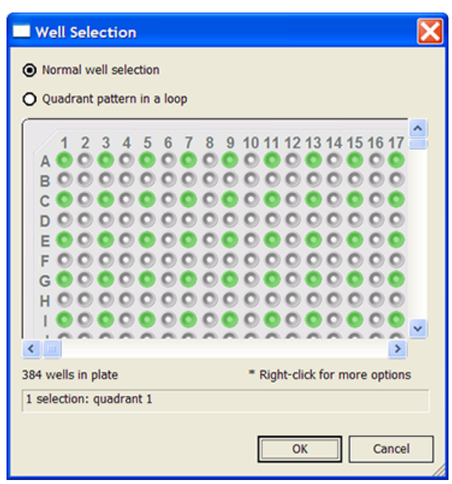
The quadrant pattern option is available only if:

- The number of channels in the pipette head (or pins in a pin tool) is fewer than the number of wells in the microplate. For example, you can use a 96-channel pipette head to dispense liquid into a 384-well microplate or 1536-well microplate.
- All the channels are selected in the Set Head Mode task when using a pipette head. (The Set Head Mode task is not an option when using a pin tool).
- The liquid-handling task is inside a loop.

IMPORTANT If you select a quadrant pattern, specifications in the Well Selection dialog box will override `task.Wellselection` values assigned in the Advanced Settings area.

To select a quadrant pattern:

1 In the **Task Parameters** area, click the **Well selection** parameter box, and then click the Browse button. The Well Selection dialog box opens. By default, the **Normal well selection** option is selected. This option is used for column- and row-wise liquid-handling patterns.

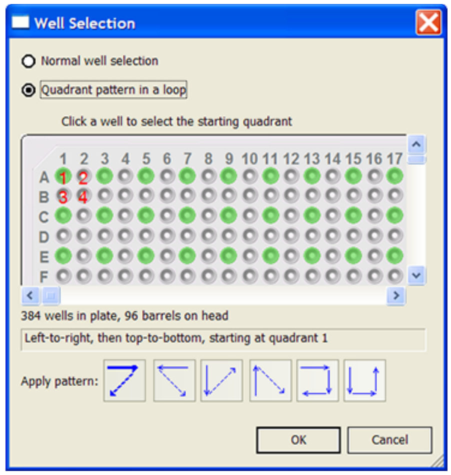


2 Select **Quadrant pattern in a loop.** The contents of the dialog box change. Notice the following:

- Red numbers (1 through 4) appear on wells A1, A2, B1, and B2. The numbers indicate the pipetting sequence: 1 is the starting well, and 4 is the last well. In the following example, the sequence is A1, A2, B1, B2.
- Green wells indicate the starting well in the pipetting sequence.

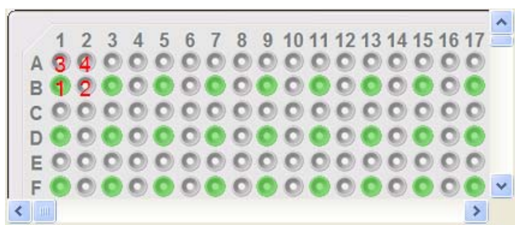
- Pattern buttons at the bottom of the dialog box indicate the movement of the pipette channels. The movement description is provided in the text box above the buttons.

  Note: The last two patterns are unavailable if a group contains 16 wells. For example, the last two patterns are not available if you have a 96-well pipette head and a 1536-well microplate.



3 Select the starting well. The well becomes green and is labeled 1.

In the following example, the third quadrant (B1 well) is selected.



4 Click a pattern button to specify the pipette channel movement. After you click a pattern, the red numbers in the graphic are updated to show the sequence.

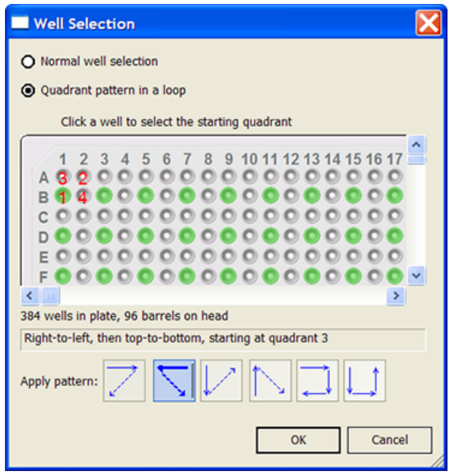
In the following example, the second pattern is selected (right-to-left, then top-to-bottom). The third quadrant (B1) is the starting well. The resulting movement is:

Quadrant 3 (B1)

Quadrant 2 (A2)

Quadrant 1 (A1)

Quadrant 4 (B2)



5 When you are finished, click **OK** to save the changes and return to the VWorks window.

Example: Dispense into a microplate on the Bravo Platform

Goal

Aspirate contents from a source microplate (Source 1) and dispense into a destination microplate.

Implementation

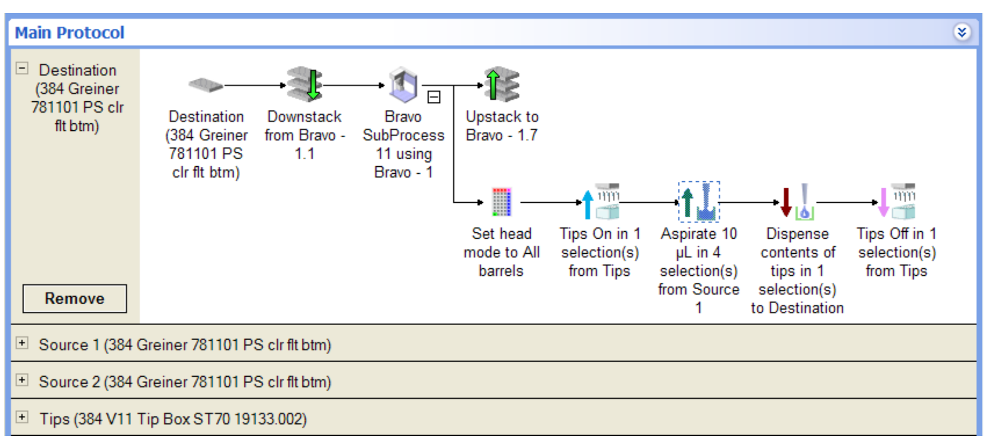
The Bravo deck is physically set up as follows:

- The destination microplates are at Bravo deck location 1.
- The source microplate is at deck location 2.
- The tipbox is at deck location 9.

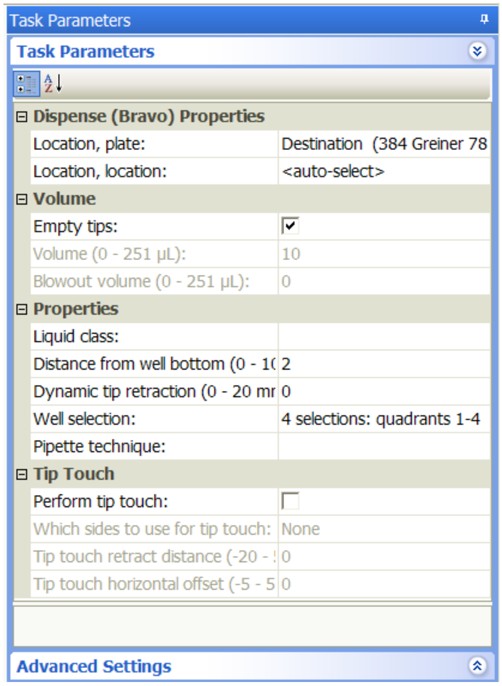
In the protocol, the following are added:

- Process for the destination microplate
- Configured labware for the source microplate
- Configured labware for the tipbox

In the Destination plate process, a Bravo subprocess is added. Within the subprocess, an Aspirate task and a Dispense task are added as shown in the following example.



In the Dispense Task Parameters area, Destination is selected, because the goal is to dispense into that microplate.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Liquid classes	*VWorks Automation Control Setup Guide*
Pipette techniques	"Specifying pipetting techniques" on page 613
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Set Head mode task	"Set Head Mode (Bravo)" on page 529
Tips On task	"Tips On (Bravo, Vertical Pipetting Station)" on page 542
Tips Off task	"Tips Off (Bravo, Vertical Pipetting Station)" on page 539
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349

For information about...	See...
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

Dispense to Waste (Bravo)

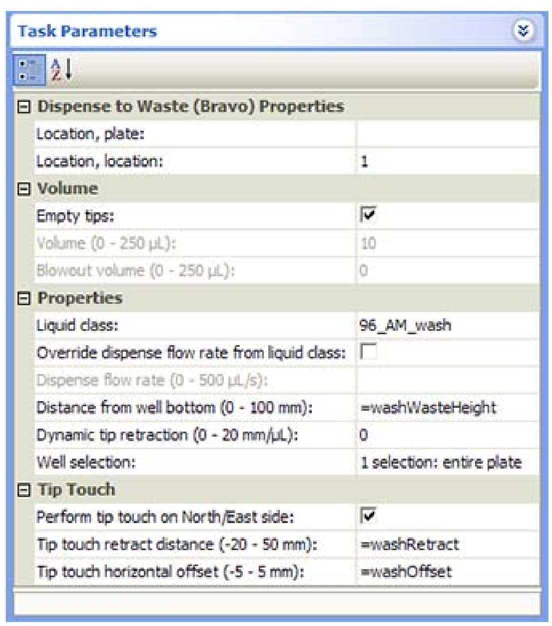
Description

The Dispense to Waste (Bravo) (Dispense to Waste (Bravo)) task moves the pipettes by a horizontal offset and then dispenses used fluid in between the chimneys in the Tip Wash Station. The software calculates the horizontal offset automatically based on the labware definition for the Tip Wash Station.

Task is available for...	Task is available in...
Tip Wash Station (MicroWash Reservoir) on Bravo Platform	Main Protocol, Bravo Subprocess

Task parameters

After adding the Dispense to Waste task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The Tip Wash Station involved in the Dispense to Waste task. *Note:* The Tip Wash Station is also known as the Microwash Reservoir.
Location, location	The location of the Tip Wash Station at which the Dispense to Waste task occurs. <auto-select> If accessories are installed on the deck, the software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task.
Empty tips	The option to empty all the liquid from the head into the Tip Wash Station outside the chimneys.
Volume (µL)	The volume of liquid to be dispensed from each pipette.
Blowout volume (µL)	Specifies the volume of air to dispense after the main volume has been dispensed while the tips are still in the Tip Wash Station.
Liquid class	The pipetting speed and accuracy. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.
Override dispense flow rate from liquid class	The option to override the dispense velocity specified in the liquid class. Selecting this option enables you to specify a value for the dispense flow rate without changing the liquid class.

Parameter	Description
Dispense flow rate (0–500 µL/s)	The numerical value or the JavaScript variable that will override the dispense velocity setting in the liquid class. A JavaScript variable enables the value to be assigned later, for example through a VWorks form. Using a VWorks form, an operator could easily change the flow rate for a dispense step in increments from as low as 1 µL/min up to 1001 µL/min using the same liquid class. If the task is included in a VWorks macro, a JavaScript variable enables you to change the value for the task at the macro level. **IMPORTANT** The software requires that the flow rate value be in microliters per second (µL/s) at run time. If you want an operator to enter the value in microliters per minute (µL/min), you can use scripting to convert the values for the software to use.
Distance from well bottom (mm)	The absolute distance between the end of the pipette tips and the bottom of the Tip Wash Station during the Dispense to Waste task. If you specify dynamic tip retraction, this is the starting distance. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.
Dynamic tip retraction (mm/µL)	The rate at which to raise the pipette head during the Dispense to Waste task. Use dynamic tip retraction to prevent spills as the pipette tips displace the liquid. To move the pipettes: • *At the same rate as the volume change.* Calculate dynamic tip retraction (DTR) as follows: DTR = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTR > 1/A • *Slower than the volume change.* DTR < 1/A The starting and ending positions can be calculated as follows: $(V_{dispensed} * DTR) + Distance_{well\ bottom}$

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Well selection</td><td>The chimney locations at which the Dispense to Waste task occurs.

Click in the parameter box, and then click the Browse button to select the locations in the Well Selection dialog box.

Use this parameter only if the pipette head has fewer tips or cartridges than the number of locations in the wash station, or if you are in single-row or single-column mode.</td></tr>
<tr><td>Perform tip touch on North/ East side</td><td>The option to touch the tip on an outer side of the adjacent northeast chimney or wall in the Tip Wash Station after performing the dispense.</td></tr>
<tr><td>Tip touch retract distance (–20 to 50 mm)</td><td>The vertical distance for the pipette tips to rise before performing the tip touch.</td></tr>
<tr><td>Tip touch horizontal offset (–5 to 5 mm)</td><td>The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition.

For example, if you set a value of:
• 0, the tips move a horizontal distance equal to the well radius
• > 0, the tips attempt to move past the well radius, which results in a more forceful tip touch
• < 0, the tips move a distance less than the radius of the well, resulting in a lighter tip touch</td></tr>
</table>

Example: Dispense into a Tip Wash Station on the Bravo Platform using a Bravo 96AM Head

Goal

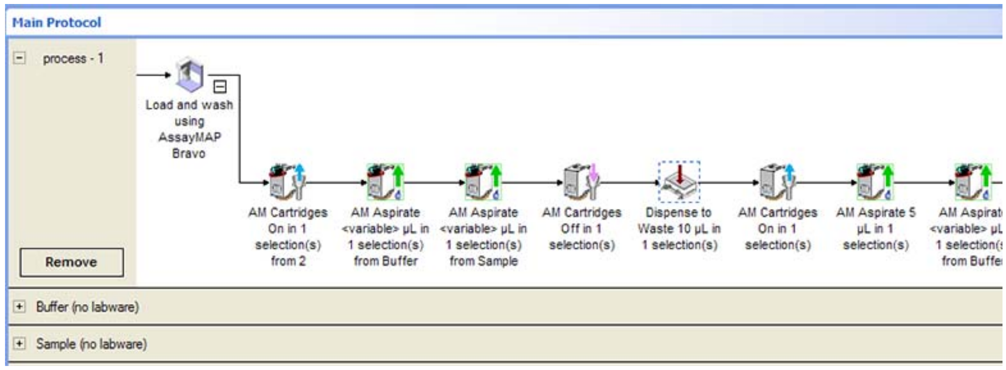
After aspirating fluids up through mounted AssayMAP Bravo cartridges, remove the cartridges, dispense the contents of the Bravo 96AM Head syringes at a Tip Wash Station, remount the cartridges for additional aspiration steps.

Implementation

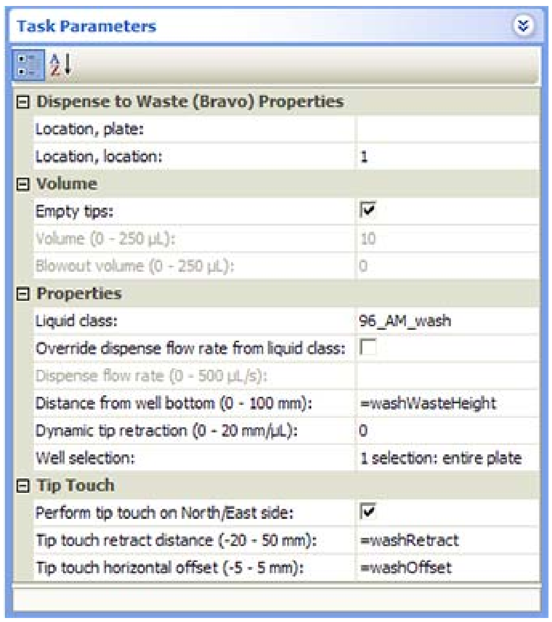
In Bravo Diagnostics, set up the Tip Wash Station on a deck location. Associate the station with one or more pump modules.

In the protocol, add a configured labware for the Tip Wash Station in addition to the other processes and configured labware required by the protocol. In the example shown, the configured reservoir is called Wash.

In the Bravo subprocess where the liquid-handling tasks are specified, a Dispense to Waste task is added after a series of aspiration tasks.



In the Dispense to Waste Task Parameters area, the Tip Wash Station is selected so that the task is performed in this labware.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Liquid classes	*VWorks Automation Control Setup Guide*
Pipette techniques	"Specifying pipetting techniques" on page 613
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392

For information about...	See...
Set Head mode task	"Set Head Mode (Bravo)" on page 529
Tips On task	"Tips On (Bravo, Vertical Pipetting Station)" on page 542
Tips Off task	"Tips Off (Bravo, Vertical Pipetting Station)" on page 539
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

Evaporate (Bravo)

Description

The Evaporate (Bravo) task (Evaporate (Bravo)) places labware on the Evaporator accessory and blows air over the labware to remove solvent or dry the sample.

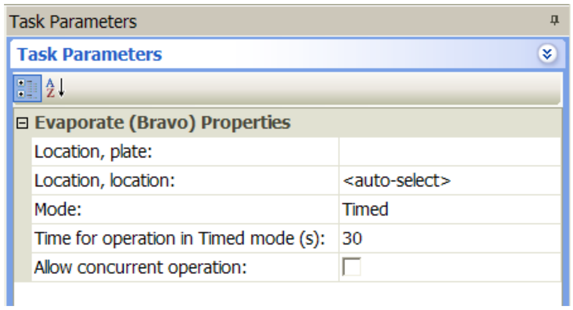
Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

Requirements

To use the Evaporate (Bravo) task, you must first configure the Evaporator in Bravo Diagnostics.

Task parameters

After adding the Evaporate (Bravo) task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Evaporate task.
Location, location	The location at which the Evaporate task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics to determine the correct location for the task.
Mode	The action of the task: • *On*. Turns on the Evaporator. • *Off*. Turns off the Evaporator. • *Timed*. Turns on the Evaporator timer. You must specify the length of time the Evaporator must remain on. If you did not select Timed, add a second Evaporate task to turn off the Evaporator.
Time for operation in Timed mode	The length of time, in seconds, you want to leave the Evaporator on. At the end of the period, the Evaporator will turn off.
Concurrent operation	The option to permit the accessory to operate simultaneously with other tasks.

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • *Bravo Automated Liquid Handling Platform User Guide*
Adding tasks in a protocol	"Adding and deleting tasks" on page 53

For information about...	See...
Configuring the Evaporator	*Bravo Automated Liquid Handling Platform User Guide*
Microplate-handling tasks	“Setting parameters for microplate-handling tasks” on page 277
Microplate-storage tasks	“Setting parameters for microplate storage tasks” on page 349
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Hit Pick Replication (Bravo)

Description

The Hit Pick Replication (Bravo) (Hit Pick Replication (Bravo)) task allows you to:

- Hit pick, or transfer contents, from selected wells in a source microplate to a destination microplate.
- Replicate a microplate by transferring contents from a source microplate into multiple destination microplates.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

Workflow

The following table presents the steps for using the Hit Pick Replication task.

Step	For this task...	See...
1	Set up labware on the Bravo deck.	“Setting up labware on the Bravo deck” on page 467
2	Create an input file.	“Creating input files” on page 468
3	Create the protocol.	“Creating a protocol: basic procedure” on page 13
4	Add the Hit Pick Replication task, and select or create a format file.	“Adding the Hit Pick Replication task” on page 470 “Creating format files” on page 480

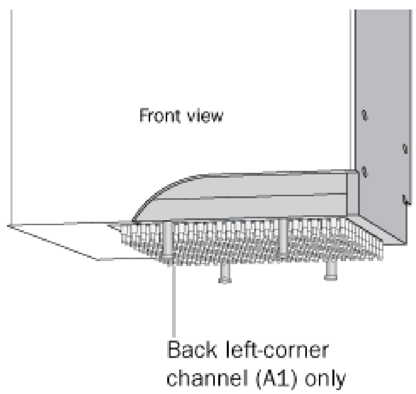
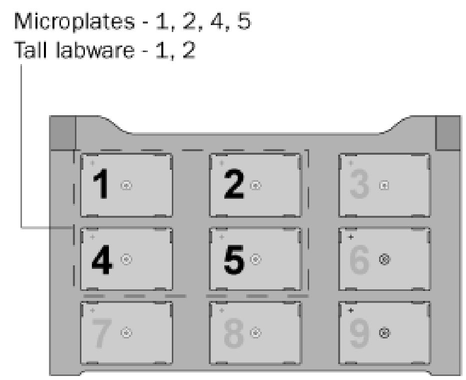
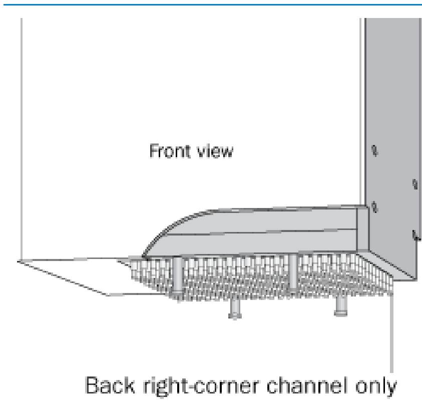
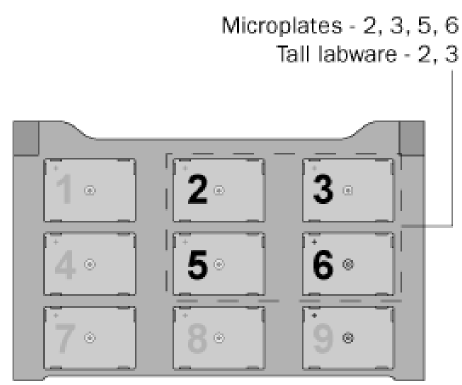
Step	For this task...	See...
5	Verify source-destination transfers.	"Verifying source-destination transfers" on page 496

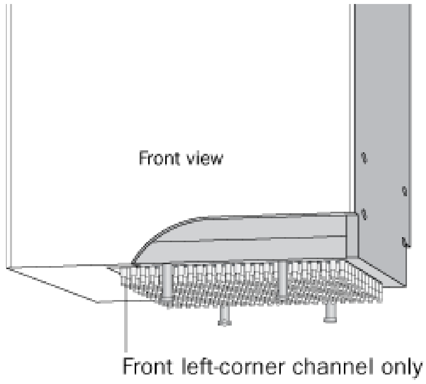
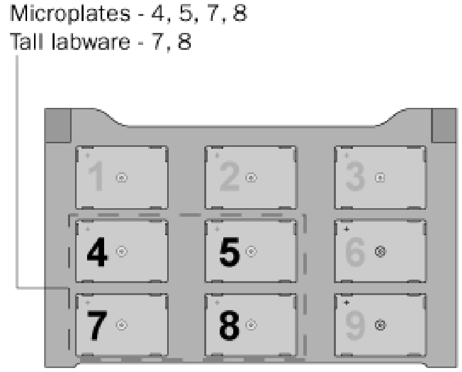
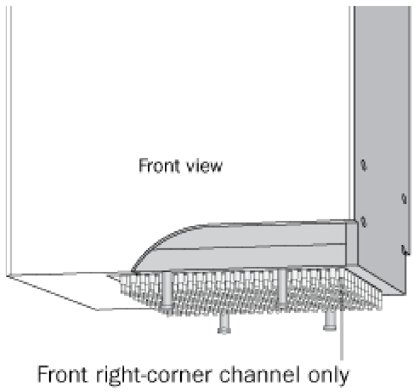
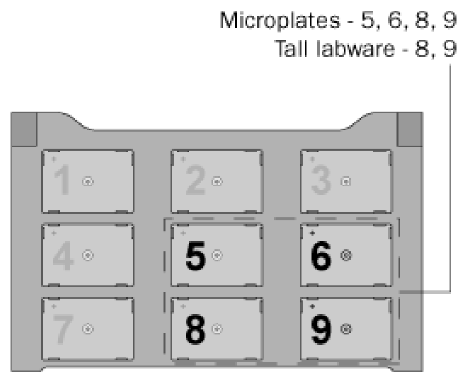
Setting up labware on the Bravo deck

For hit-picking transfers, the deck locations you can access depends on the pipette head channel you select. In addition, because tipboxes and reservoirs are tall, you must consider the locations of the these labware relative to the source and destination microplates.

The following table shows the channel selection and corresponding deck location limits for both microplates and tall labware. Use this information when setting up labware for the hit-picking task.

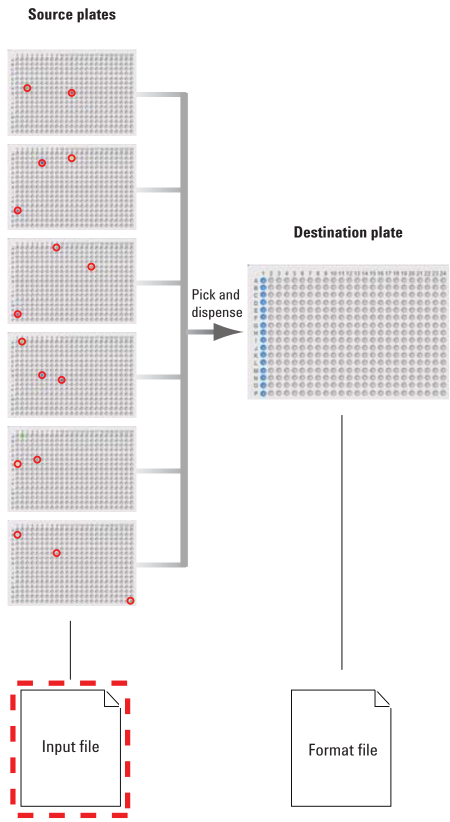
For example, if you select the A1 channel, you can place microplates at locations 1, 2, 4, and 5. If you have tall labware, such as a tipbox and a reservoir, place them at locations 1 and 2. Doing so permits the pipette head to access the microplates at locations 4 and 5 without colliding with the tall labware.

<table>
<tr><th>Pipette channel selection</th><th>Accessible Bravo deck locations</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

<table>
<tr><th>Pipette channel selection</th><th>Accessible Bravo deck locations</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Creating input files

Both the hit-picking and microplate replication tasks require an input file. An input file contains information about the source microplates.



You must have the required input file before adding the Hit Pick Replication task or before you start the protocol run. The input file can be from a LIMS or created manually. The file must meet the following requirements:

- The file must be in the comma-separated value (CSV) format.
- The file can contain a heading in the first row, or the first few consecutive rows.
- *Microplate replication*. The input file must contain a column that lists source microplate barcodes or IDs.
- *Hit picking*. The input file must contain the following columns, in any order:
  - Source microplate barcode or ID
  - Well locations (in a single column or across two columns)
  - Variable dilution factors (if dilution series will be created)
  - Transfer-volume information (if variable-volume transfer is desired)

IMPORTANT Hits from each source microplate must be in consecutive rows in the input file. In addition, the order in which the microplates are processed in protocol runs must match the order of the microplates in the input files.

The following examples show two input files (**1** and **2**) displayed in Microsoft Excel. The heading is in the first row in both files. Microplate IDs are used instead of barcodes. Notice that the well locations are displayed differently in the two files. The file on the right (**2**) also contains dilution factor information.

1

	A	B	C
1	PlateID	Well	Volume
2	Plate 1	H12	20
3	Plate 1	G3	30
4	Plate 2	C6	20
5	Plate 2	B12	15
6	Plate 2	M1	20
7	Plate 3	A9	20
8	Plate 3	E16	30
9	Plate 3	O1	15
10	Plate 4	H6	20
11	Plate 4	I10	30
12	Plate 5	A2	20
13	Plate 5	F5	20
14	Plate 5	G1	20
15	Plate 6	F9	15
16	Plate 6	B1	30
17	Plate 6	P24	20

2

	A	B	C	D
1	PlateID	Well row	Well col	Dil factor
2	Plate 1	H	12	2
3	Plate 1	G	3	2
4	Plate 2	C	6	3
5	Plate 2	B	12	4
6	Plate 2	M	1	3
7	Plate 3	A	9	1
8	Plate 3	E	16	2
9	Plate 3	O	1	3
10	Plate 4	H	6	4
11	Plate 4	I	10	2
12	Plate 5	A	2	4
13	Plate 5	F	5	3
14	Plate 5	G	1	2
15	Plate 6	F	9	3
16	Plate 6	B	1	1
17	Plate 6	P	24	4

Adding the Hit Pick Replication task

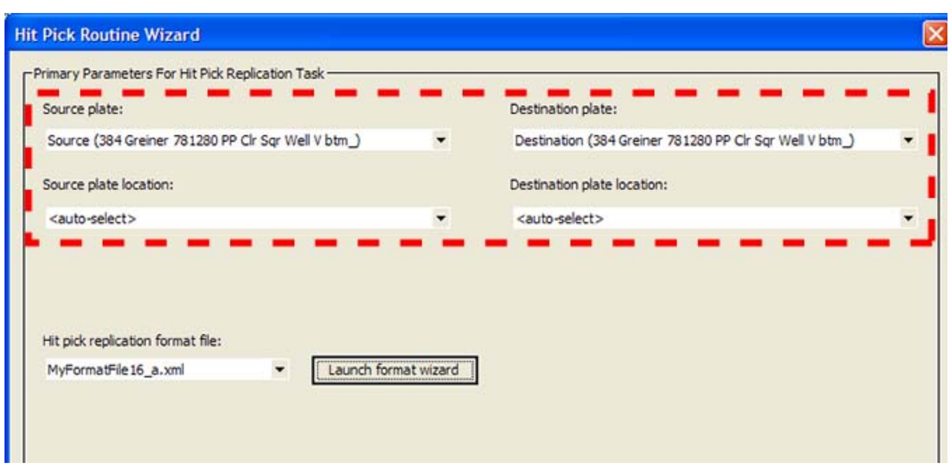
When you add the Hit Pick Replication (Bravo) task, the software automatically starts the Hit Pick Routine Wizard.

In the Hit Pick Routine Wizard:

1 Select the source and destination microplates. You must also select their locations.

When setting up hit-picking transfers, make sure the source microplate and destination microplate are at one of the accessible deck locations. See "Setting up labware on the Bravo deck" on page 467.

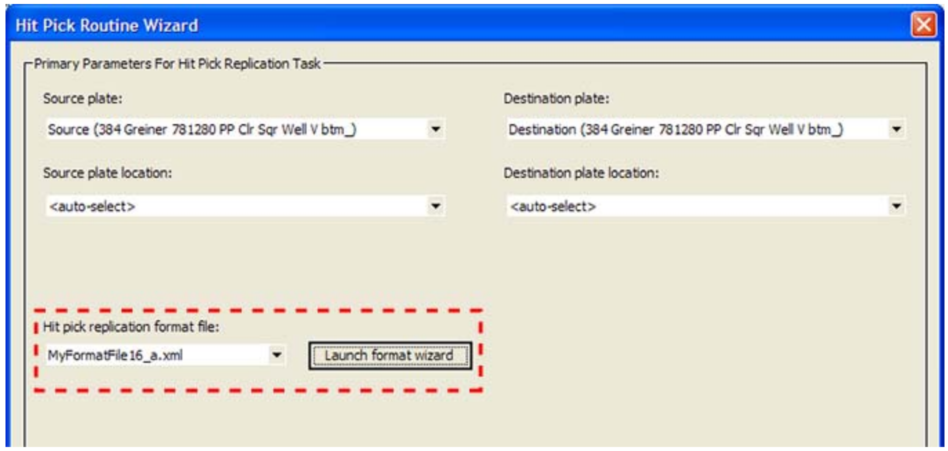
For example, if you are using the A24 (back right-corner channel) pipette-tip channel for hit picking, the accessible deck locations are 2, 3, 5, and 6. If the source microplate will be at deck location 5 and the destination microplate will be at deck location 6, you must provide this information on the first page of the Hit Pick Routine Wizard as shown.



2 Do one of the following:

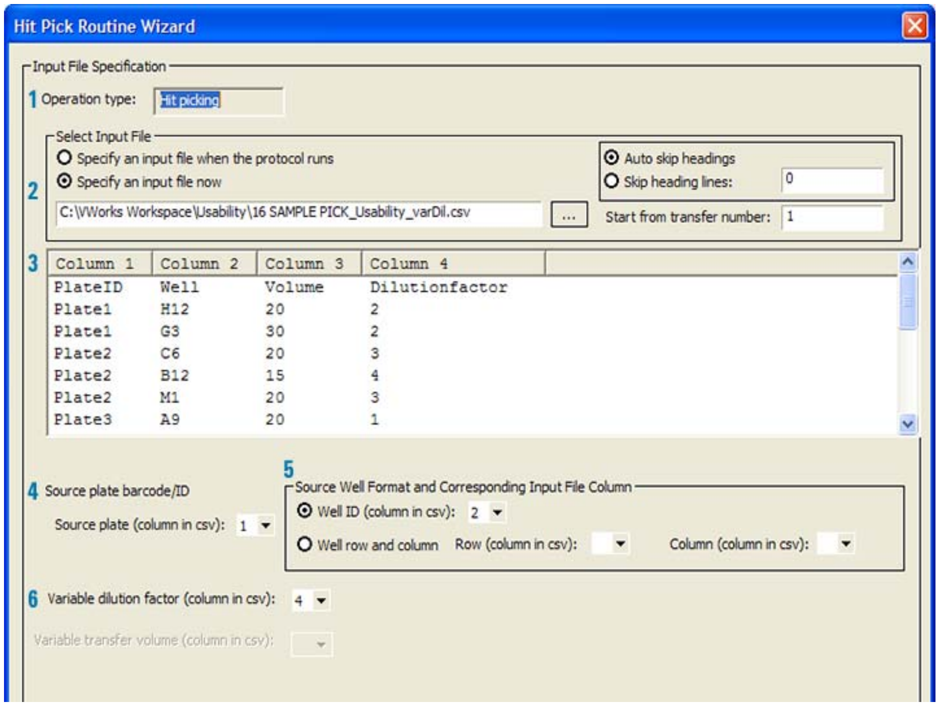
- Select an existing format file from the **Hit pick replication format** list.

- Click **Launch format wizard** to create a new format file. See “Creating format files” on page 480 for detailed instructions.



When you are finished, click **Next.**

3 Supply the input file information. If you do not have an input file, see "Creating input files" on page 468.

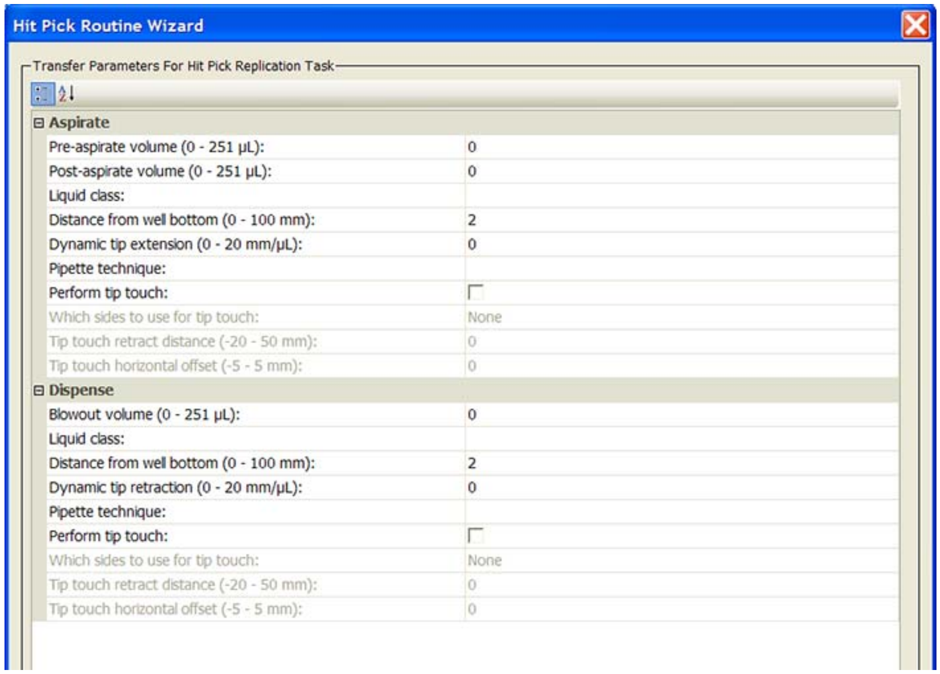


Item	Input file specification	Description
1	Operation type	Displays the type of operation: Hit picking, or replication. The selection is made in the format file and cannot be changed on this page. For information about the format file, see "Creating format files" on page 480.

Item	Input file specification	Description
2	Select input file	Allows you to select one of the following: • **Specify an input file when the protocol runs** • **Specify an input file now** If you select **Specify an input file now**, use the browse button to locate and select the desired input file.
		In either case, you can select one of the following: • *Auto skip heading*. Enables the software to automatically find and skip rows that contain heading information. • *Skip heading lines*. Skips the specified number of rows at the top of the input file. Select this option if you want to manually specify the number of heading rows.
		In addition, you can specify the row to start the transfer from in the **Start from transfer number** box. 1 indicates starting the first transfer from the first row after the heading rows.
3	Input file contents	Displays the contents of the input file.
4	Source plate barcode/ID	Allows you to select the column that contains the source plate barcode or ID.
5	Source well format and corresponding input file column	*Hit-picking only*. Allows you to indicate whether the well location is specified as: • *Well ID*. The well location is in a single column (for example, A24). Select the column that contains the well ID. • *Well row and column*. The well location is in two columns. Select the columns that contain the well row and well column information. The row information can be a letter or a number.
6	Variable dilution factor and Variable transfer volume	*Hit-picking only*. Allows you to select the column that contains the dilution factors and transfer volume.

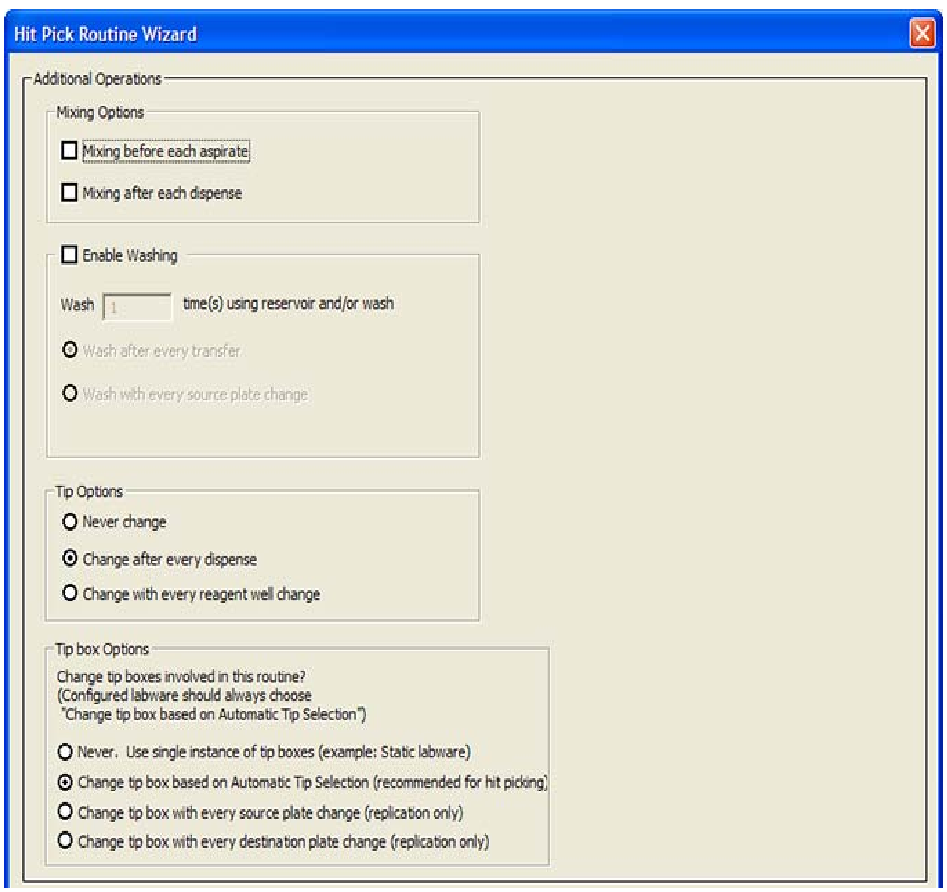
When you are finished, click **Next.**

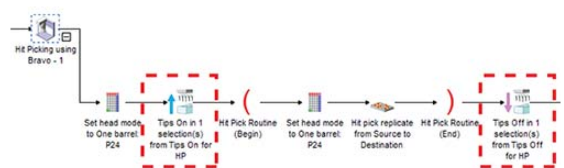
7 Specify the Aspirate and Dispense task parameters. For detailed parameter descriptions, see "Aspirate (Bravo, Vertical Pipetting Station)" on page 392 and "Dispense (Bravo, Vertical Pipetting Station)" on page 451.



When you are finished, click **Next.**

8 Select additional operations, if applicable.

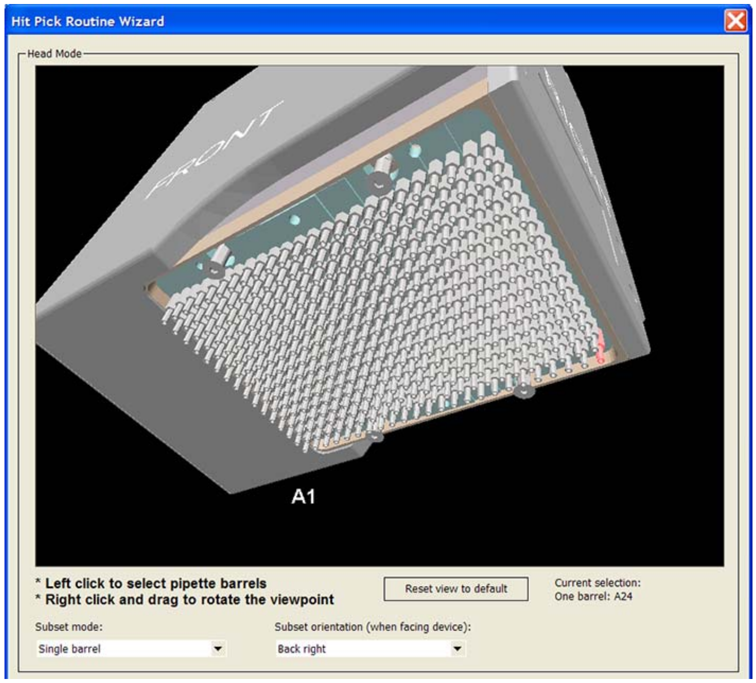


Additional operation	Description
Mixing Options	Select where you want to add a Mix task: • Before each aspirate task • After each dispense task You will need to set the Mix parameters later in the wizard. If you do not want to mix, clear both check boxes.
Enable Washing	Select the check box if you want to add one or more Wash Tips tasks. You can specify the number of tip washes and whether you want to: • Wash after every transfer • Wash with every source microplate change You will need to set the Wash Tips parameters later in the wizard. Clear the check box if you do not want to add any Wash Tips tasks.
Tip Options	Select the tip-change option: • *Never change*. Does not change the tips within the hit-picking routine. *Note:* If you select Never change, you must add the Set Head Mode, Tips On, and Tips Off tasks in the subprocess, outside of the hit-picking routine, after you are finished with the wizard.  • *Change after every dispense*. Changes the tips after every dispense within the hit-picking routine. • *Change with every source well change*. Changes the tips after every source well change during the hit-picking routine.

Additional operation	Description
Tip Box Options	Select when to change the tipbox: • *Never. Use single instance of tip boxes (example: Static labware)*. Does not change the tipbox within the hit-picking routine. • *Change tip box based on Automatic Tip Selection (recommended for hit picking)*. Uses tip-tracking within the hit-picking routine, and changes the tipbox when the box is fully used. Select this option if the tipboxes you are using for hit-picking are configured labware. *Note:* Tip-tracking continues in the Dilute to Final Volume routine. • *Change tip box with every source plate change (replication only)*. Changes the tipbox every time the source microplate changes. This option is available only when a replication format file is selected. • *Change tipbox with every destination plate change (replication only)*. Changes the tipbox every time the destination microplate changes. This option is available only when a replication format file is selected.

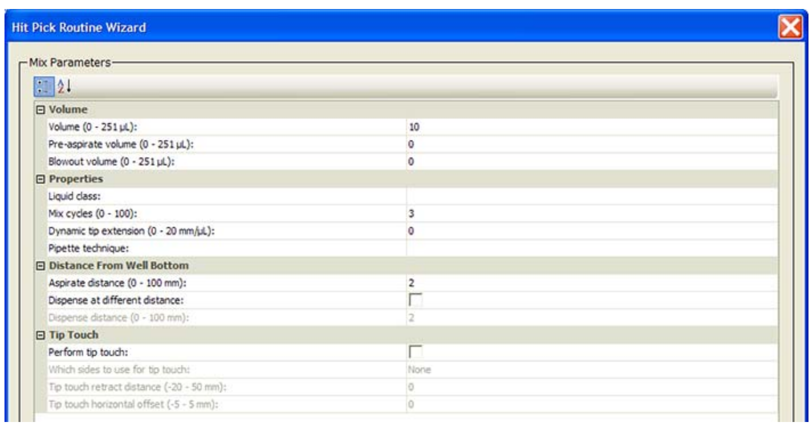
When you are finished, click **Next.**

9 Select the pipette channels to use. For a description of the selections, see "Set Head Mode (Bravo)" on page 529.



When you are finished, click **Next.**

10 *If you selected mixing options in step 8*. Set the mixing parameters. For a description of the parameters, see "Mix (Bravo, Vertical Pipetting Station)" on page 497.



When you are finished, click **Next.**

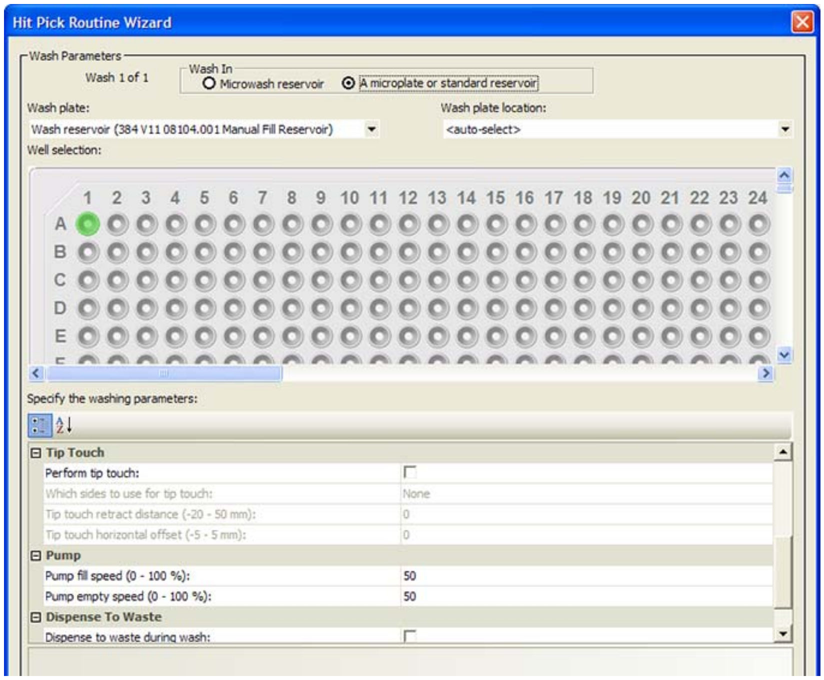
11 *If you selected wash options in step 8*. Set the wash parameters.

a In the **Wash In** area, select the type of labware to be used for the wash task:

- *Tip Wash Station (also known as MicroWash Reservoir)*. Adds Wash Tips tasks and will use either configured or static labware.
- *A microplate or standard reservoir*. Adds Mix tasks.

b In the **Wash plate** list, select the desired labware and location.

c Set wash-tip parameters. For a description of the parameters, see “Wash Tips (Bravo, Vertical Pipetting Station)” on page 550.



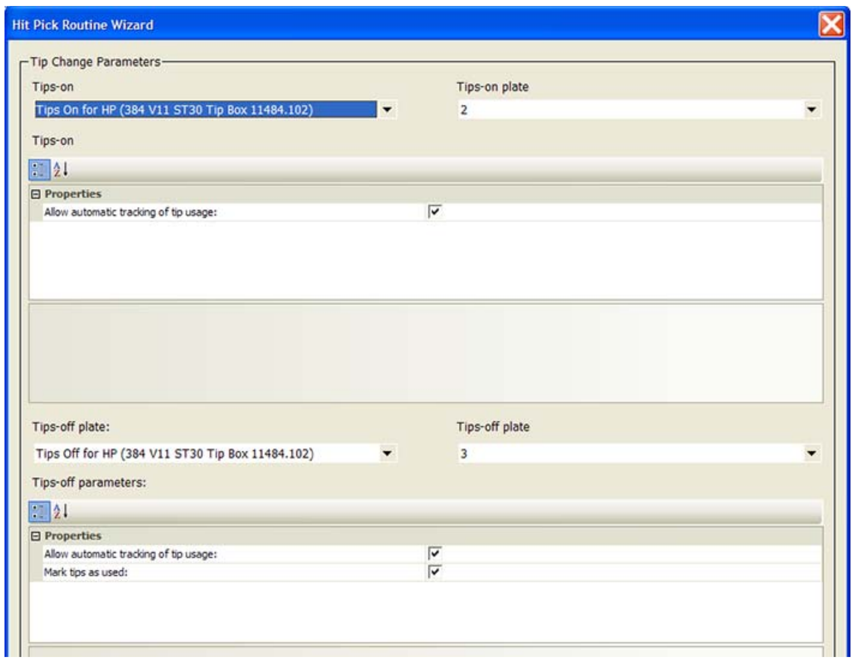
When you are finished, click **Next.**

12 *If you selected a tip-change option in step 8.* Select the tipbox options.

a Select the tipboxes you want to use.

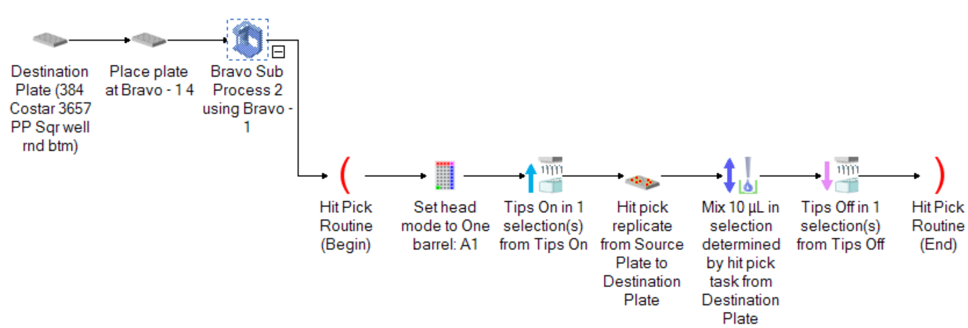
b Select the tip-tracking options you want to use:

- *Allow automatic tracking of tip usage.* Allows the software to track pipette tip usage during the protocol run or across different protocol runs. If you select the option in this task, you must also select the option in the Tips Off task. In general, use the default selections displayed.
- *Mark tips as used.* The option to use only new pipette tips during the protocol run. Select the option so that the software counts the number of tips used during the protocol run. The tips that have been used once are marked as used so that they cannot be reused. Clear the check box so that during the next Tips On task, the same tips can be reused. In general, use the default selections displayed.



13 Click **Finish.**

Based on the information you provide in the wizard, the software adds the Aspirate, Dispense, Mix, Wash Tips, and other tasks that are necessary to produce the desired hit-picking or replication results.



Notice the following:

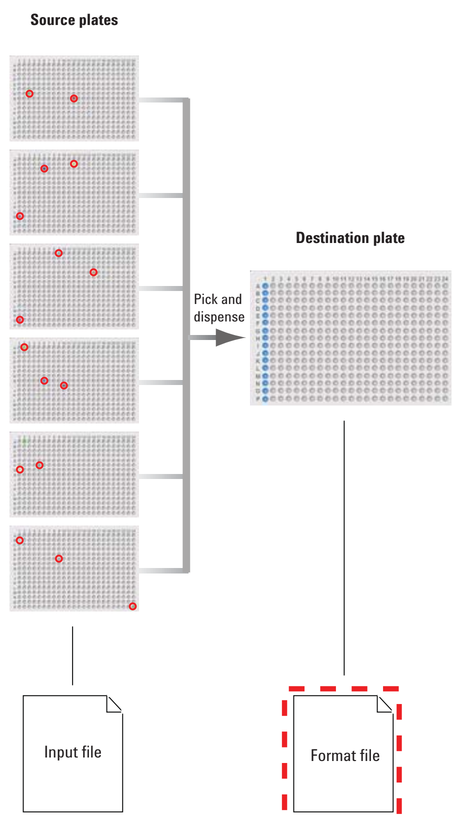
- The Hit Pick Routine (Begin) and Hit Pick Routine (End) tasks mark the beginning and end of the microplate-replication or hit-picking tasks.
- You can modify any of the task parameters directly in the Task Parameters area.
- You can add more tasks to or delete tasks from the routine.
- You can return to the Hit Pick Routine Wizard. To do this, select either the Hit Pick Routine (Begin) or Hit Pick Routine (End) task. In the Task Parameters area, click **Launch hit pick routine wizard.**

CAUTION Whenever you return to the Hit Pick Routine Wizard, make modifications, and finish the wizard, a new set of tasks will replace the existing tasks in the routine.

Creating format files

About format files

For both microplate replication and hit picking, you are required to use a format file. The format file shows the dispense pattern in destination microplates.



Workflow

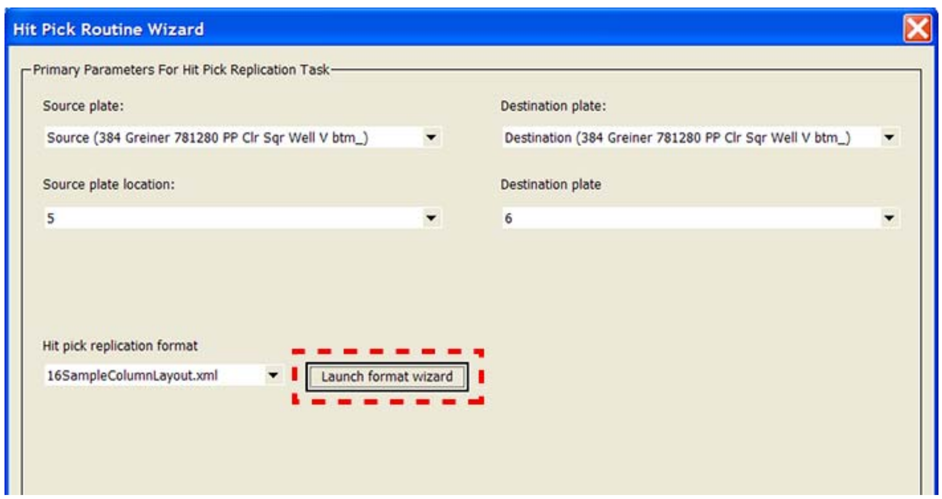
You use the Hit Pick Replication Format Wizard to create the format file. You can access the wizard when you add the Hit Pick Replication task. The overall workflow for creating the format file is:

Step	For this task...	See...
1	Create the format file.	“Creating format files” on page 480
2	Add contents to the format file for one of the following: • Hit picking, and dilution series (if desired) • Replication	One of the following: • “Adding hit-picking information in the format file” on page 482 • “Adding replication information in the format file” on page 488

Creating new format files

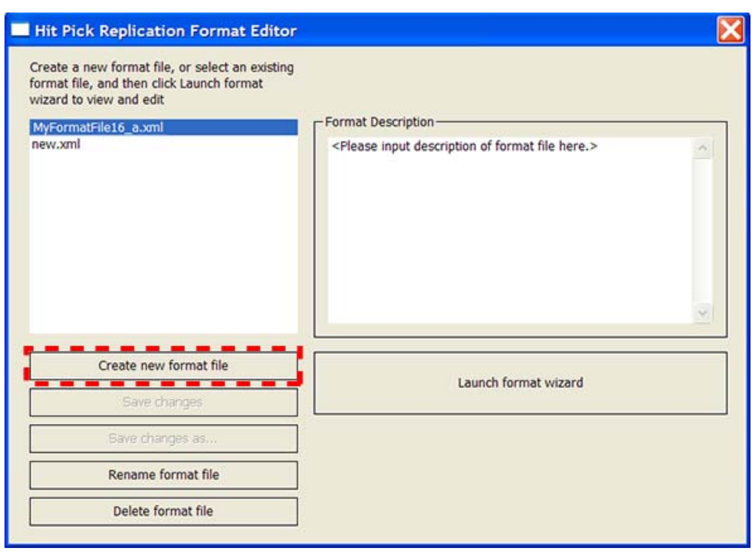
To create a new format file:

1 On the first page of the Hit Pick Routine Wizard, click **Launch format wizard.**



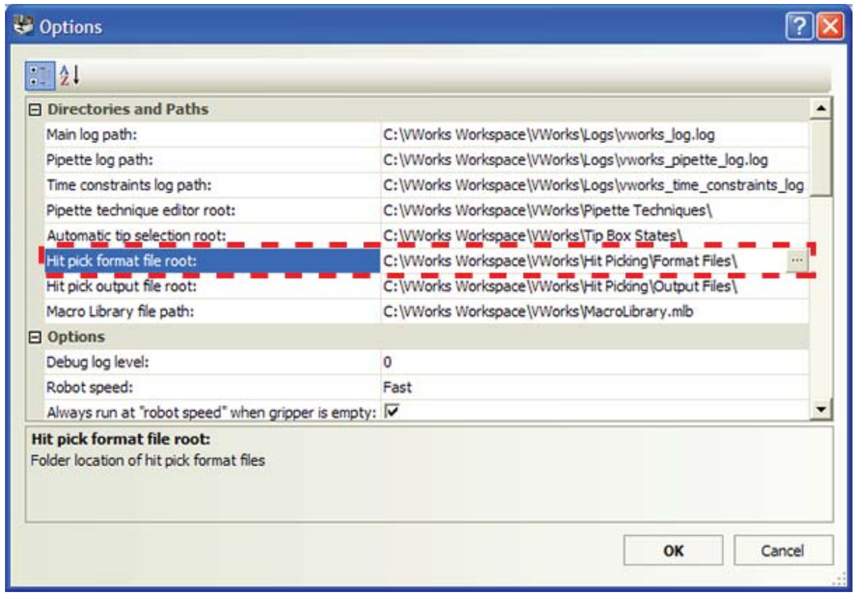
The Hit Pick Replication Format Editor opens.

2 Click **Create new format file,** type a name for the file in the prompt dialog box, and then click **OK.** The new format file name appears in the list.



The software saves the format file in the XML format in the following folder: C:\VWorks Workspace\VWorks\Hit Picking\Format Files\

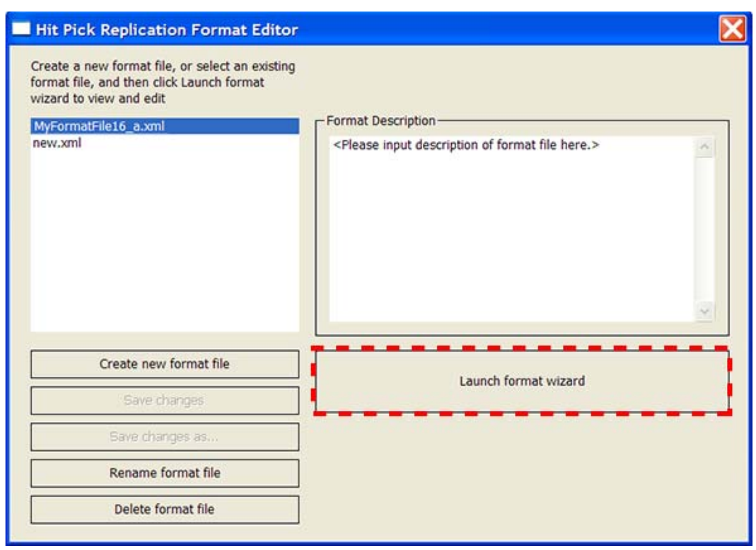
You can change the default storage location. To do this, in the VWorks window, select **Tools > Options.** Change the file path in the **Directories and Paths** area of the **Options** dialog box.



Adding hit-picking information in the format file

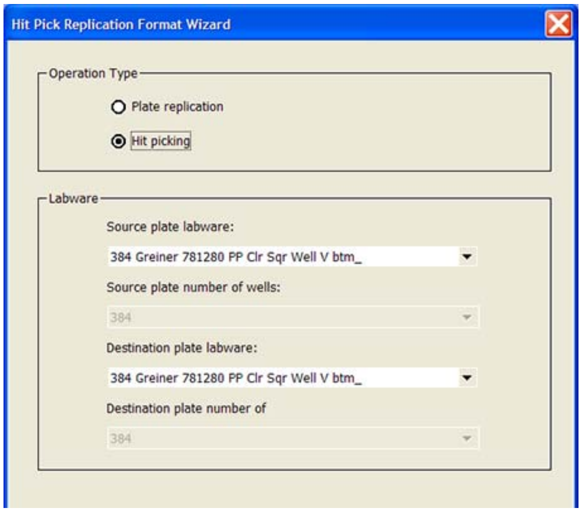
To add hit picking information in the format file:

1 In the **Hit Pick Replication Format Editor,** make sure the format file name is selected, and then click **Launch format wizard.**



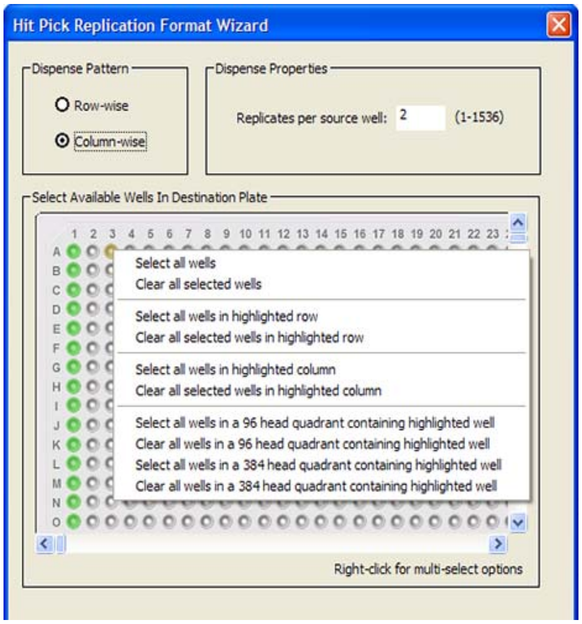
The Hit Pick Replication Format Wizard opens.

2 On the first page of the wizard:



- **a** In the **Operation Type** area, select **Hit picking.**
- **b** In the **Labware** area, select the **Source plate labware** and the **Destination plate labware.**
- **c** Click **Next.**

3 On the dispense information page:



- **a** In the **Dispense Pattern** area, select one of the following dispense methods:
  - Row- wise
  - Column- wise
- **b** In the **Dispense Properties** area, type the number of **Replicates per source well,** if applicable.

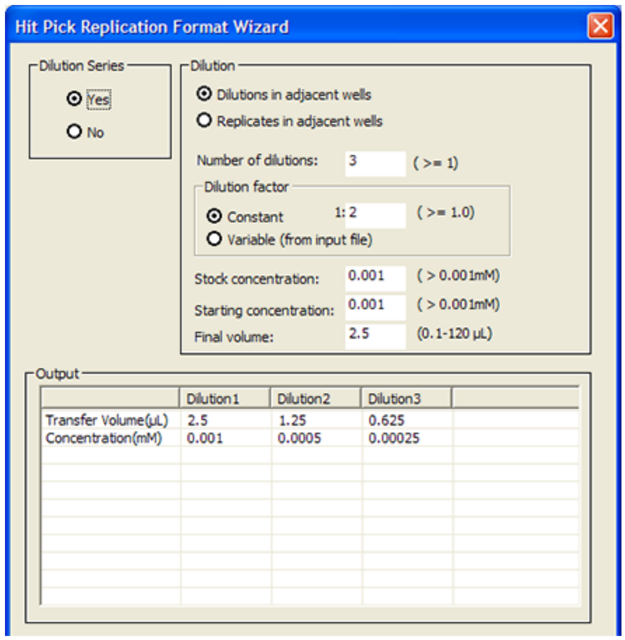
c In the destination plate graphic, select the wells into which the source contents will be dispensed. Click a well to individually select or clear the well. Alternatively, right-click a desired well and select a command from the menu that appears.

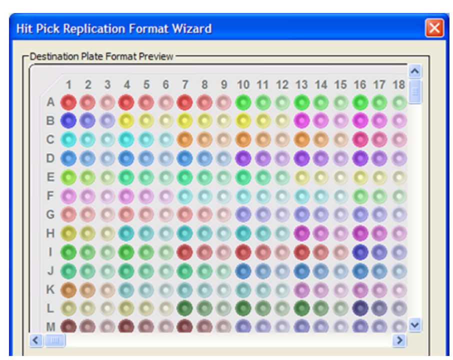
d Click **Next.**

4 On the dilution series page:

a In the **Dilution Series** area, select one of the following:

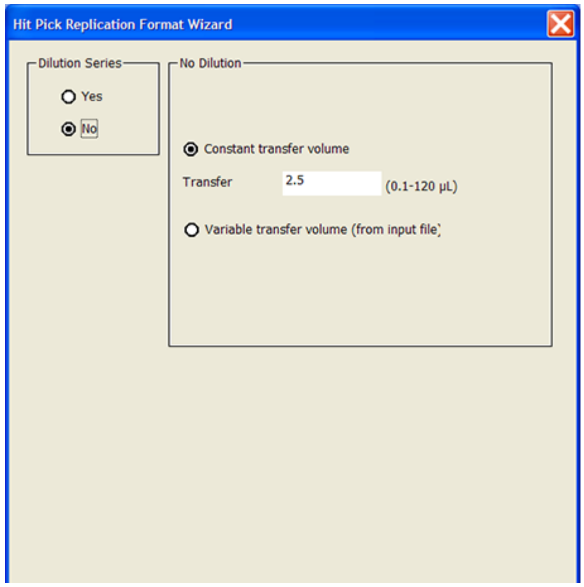
- *Yes*. If you select **Yes,** you must supply additional information about the dilution series.



Dilution selection or parameter	Description
Dilution in adjacent wells	Dispenses the dilution in adjacent wells before replicating the same series, as the following example shows. Notice that the dispense pattern is row-wise. 

Dilution selection or parameter	Description
Replications in adjacent wells	Dispenses the replicates in adjacent wells before continuing the dilution series, as the following example shows. Notice that the dispense pattern is row-wise. 
Number of dilutions	Specifies the number of dilutions to perform. A value of 1 means no dilution is performed.
Dilution factor	Uses the specified dilution factor: • *Constant*. The same dilution factor is used throughout the process. Type the factor to use in the box. For example, a factor of 1:2 dilutes the starting concentration by one-half. If the number of dilutions is 3, then the concentrations in the series are 1, 1:2, and 1:4. • *Variable (from input file)*. Different factors are used during the process. The factors are specified in the input file.
Stock concentration	Specifies the concentration of the stock solution.
Starting concentration	Specifies the concentration of the starting concentration.
Final volume	Specifies the concentration of the final volume in each well. The software will use this value to calculate the volume for prefilling or backfilling. See “Dilute to Final Volume (Bravo)” on page 442.

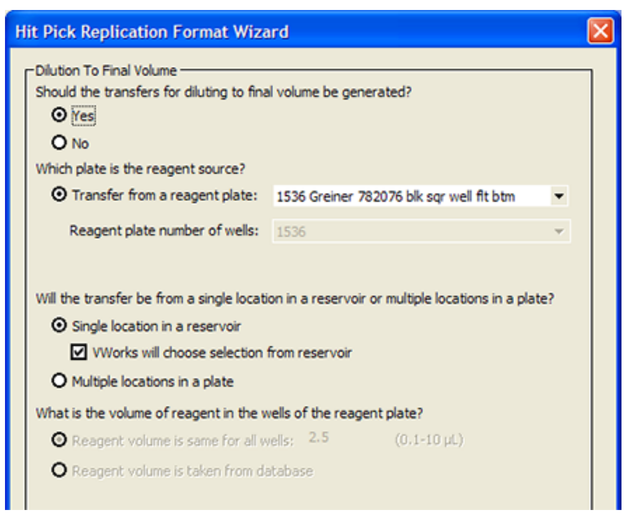
– *No*. If you select **No,** you must select the type of volume transfer.



Dilution selection or parameter	Description
Constant transfer volume	Uses the same transfer volume throughout the process.
Variable transfer volume	Uses different transfer volumes during the process. The volumes are specified in the input file.

b Click **Next.**

5 *If you selected dilution series in step 4*. Select the desired options for dilute to final volume.



a Select whether you want to backfill or prefill the destination microplate. If you select **Yes**, you must supply additional information. If you select **No**, you can proceed to step d.

b Select the reagent microplate to use for the backfilling or prefilling process.

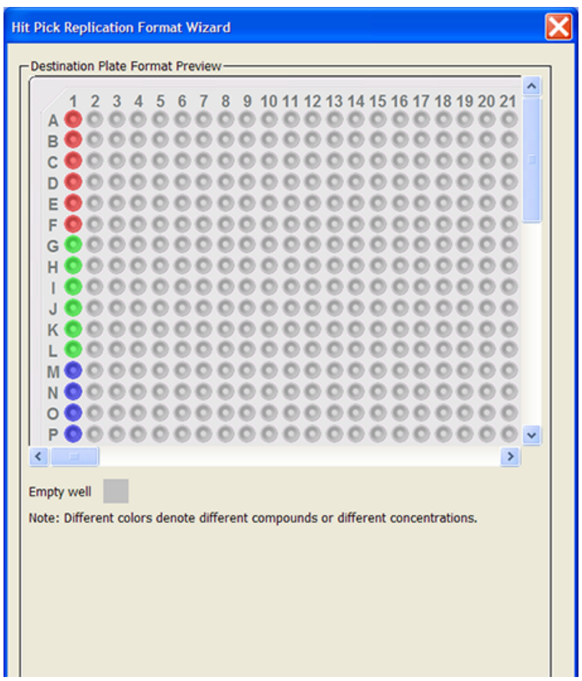
c Select whether the transfer will be from one of the following:

- **Single location in a reservoir.** Some reservoir designs might limit the pipette's access. You can allow the software to determine how to access the reservoir, based on the labware definition. To do this, select **VWorks will choose selection from reservoir.** Alternatively, you can clear the check box and manually select the access point in the reservoir, using the microplate graphic displayed. The wells in the microplate represent different access points in the reservoir.
- **Multiple locations in a plate.** If the transfer will be from a microplate, you can select the wells from which to aspirate. In addition, you can specify a constant reagent volume, or indicate that the reagent volume information will be obtained from a database.

d Click **Next.**

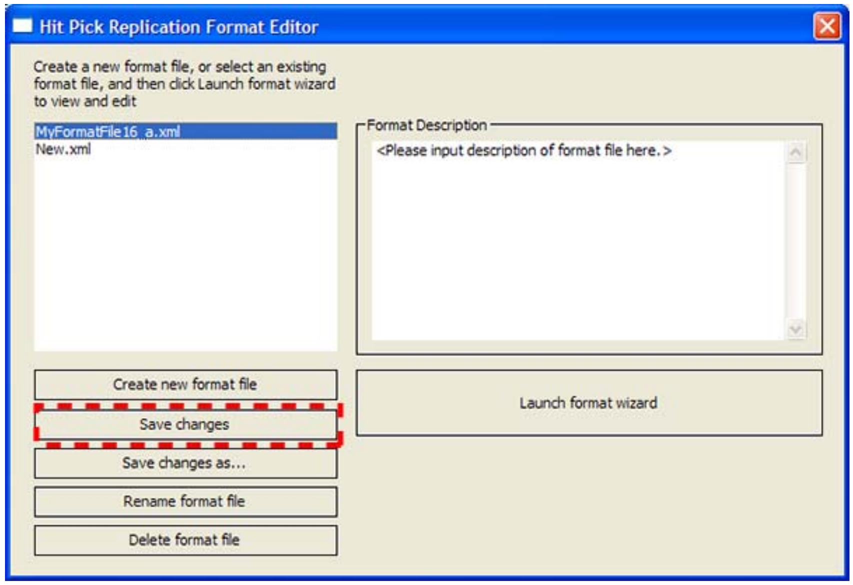
6 On the destination plate preview page, confirm that the dispense pattern is correct.

Note: The dispense pattern shown on the screen is a preview of the *first* destination microplate only. To check the mapping in all of the destination microplates, compile the protocol, and then view the output files. Alternatively, you can check the Pipette Log after running the protocol in simulation mode. For more information about output files and the Pipette Log, see "Verifying source-destination transfers" on page 496.



When you are done, click **Finish**.

7 Back in the **Hit Pick Replication Format Editor,** make sure the correct format file name is selected, and then click **Save changes.** If you altered an existing format file, you can click **Save changes as** to save it as a different file.

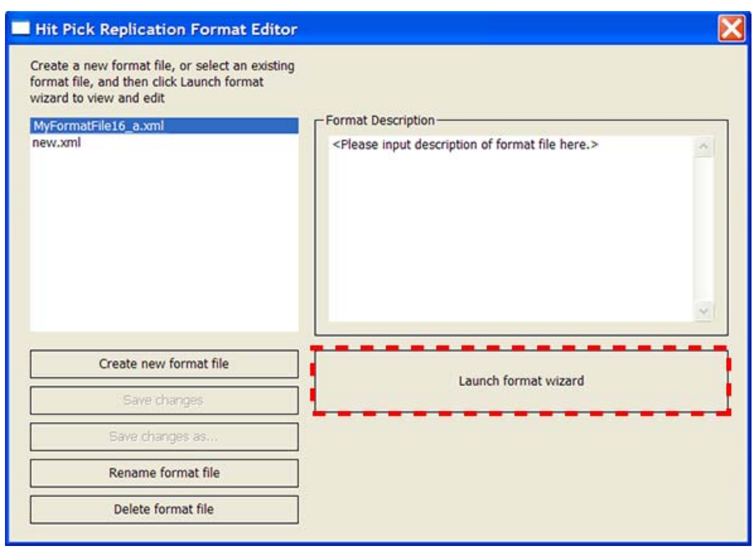


8 Click the close button ( ) at the top right corner of the dialog box to return to the Hit Pick Routine Wizard. To continue in the Hit Pick Routine Wizard, return to step 3 in "Adding the Hit Pick Replication task" on page 470.

Adding replication information in the format file

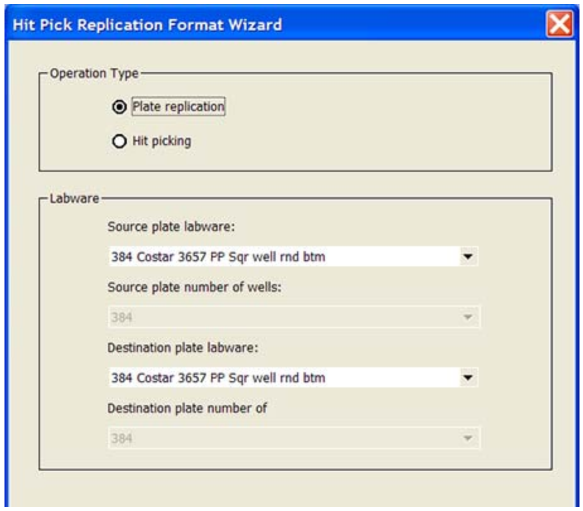
To add replication information in the format file:

1 In the **Hit Pick Replication Format Editor,** make sure the format file name is selected, and then click **Launch format wizard.**



The Hit Pick Replication Format Wizard opens.

2 On the first page of the wizard:

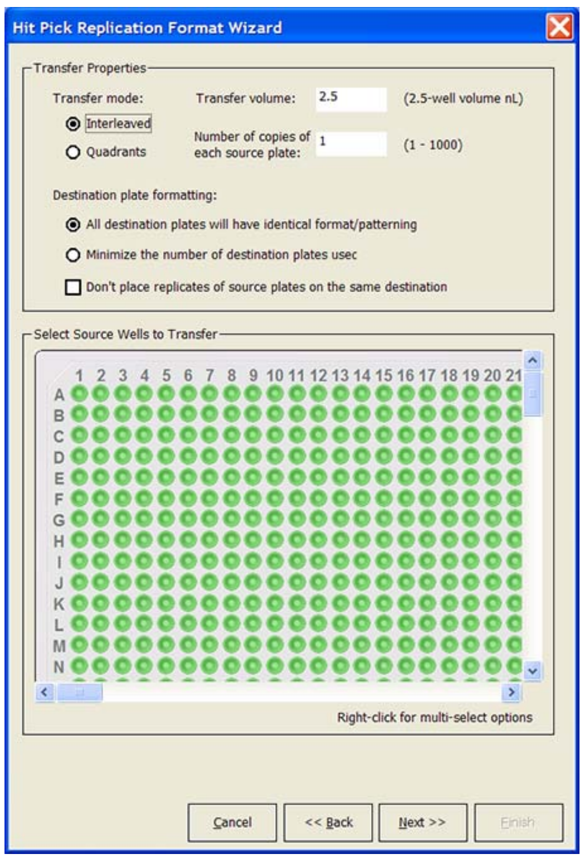


a In the **Operation Type** area, select **Plate replication.**

b In the **Labware** area, select the **Source plate labware** and the **Destination plate labware.**

c Click **Next.**

3 On the transfer page, select the **Transfer mode.**

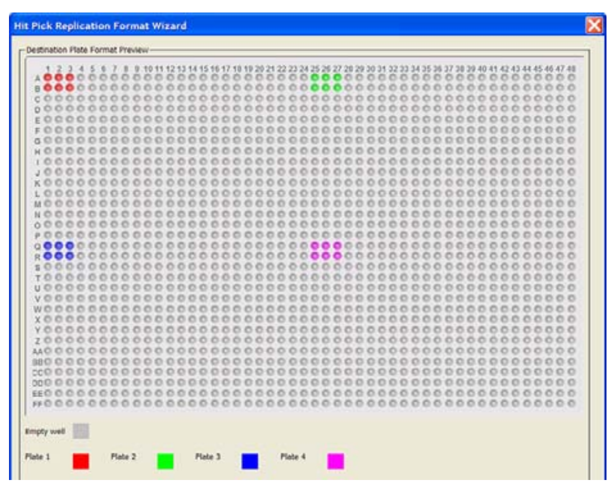


<table>
<tr><th>Transfer mode</th><th>Description</th></tr>
<tr><td>Interleaved</td><td>Contents of corresponding wells from different source microplates are placed consecutively in the destination microplate.

For example, if you are transferring contents from four 384-well source microplates into a 1536-well destination microplate, the Interleaved transfer would result in the following destination microplate layout:


</td></tr>
<tr><td>Quadrants</td><td>Contents from different source microplates are placed in quadrants of the destination microplate.

For example, if you are transferring contents from four 384-well source microplates into a 1536-well microplate, the Quadrant transfer will result in the following destination microplate layout:

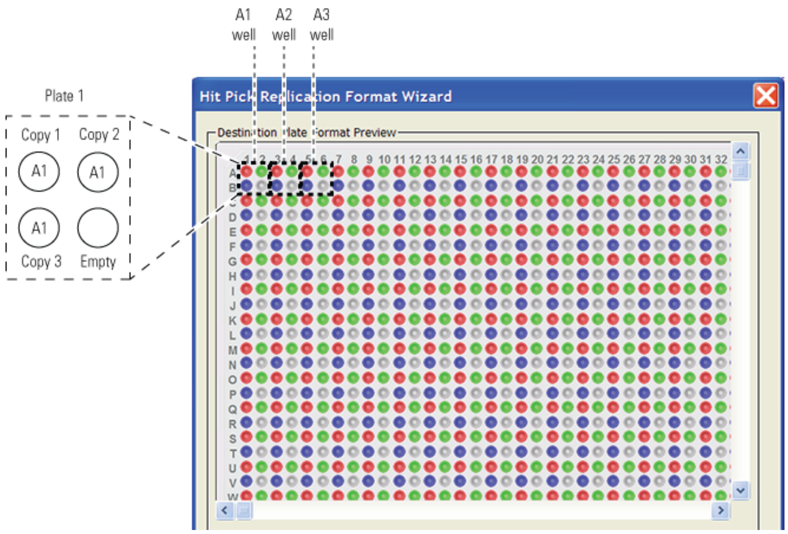

</td></tr>
</table>

4 Type the volume to transfer from each source microplate well.

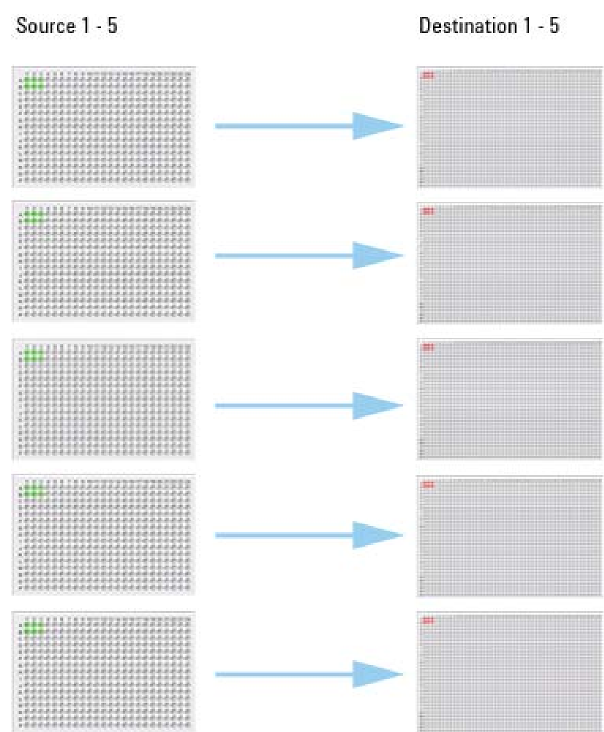
5 Type the number of copies you want to transfer from the source microplate.

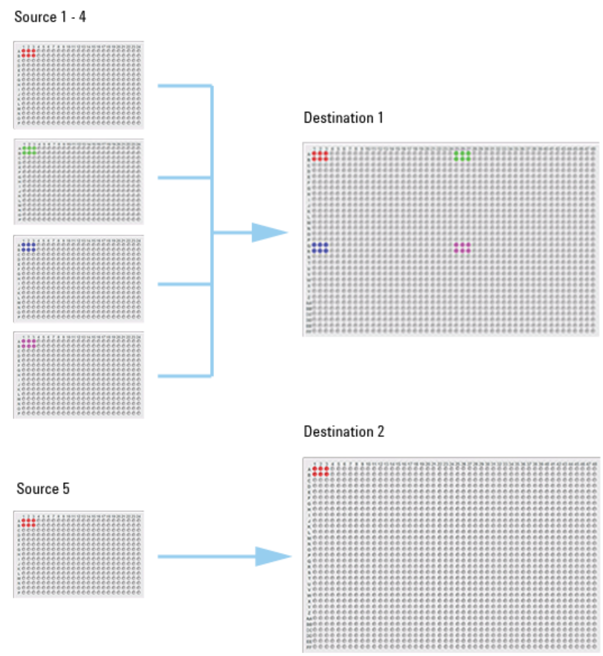
IMPORTANT This value will affect the layout of the contents in the destination microplate.

For example, if you are transferring three copies of the contents from a 384-well source microplate into a 1536-well destination microplate, the Interleaved transfer will result in the following destination microplate layout. Notice that the wells in the fourth quadrant remain empty.



6 Select the destination microplate format:

Transfer mode	Description
All destination plates will have identical format/patterning	All destination microplates will look identical. The layout of the destination microplate depends on the format of the source microplate, the number of copies you want to transfer, the total number of source microplates, and other selections in the wizard. For example, suppose you want to transfer contents from five 384-well source microplates into one or more 1536-well destination microplates. One 1536-well microplate can only accommodate contents from four 384-well microplates. To maintain the same layout across all of the destination microplates, five destination microplates will be used. The contents from each source microplate are transferred to one 1536-well destination microplate. *Note:* In the example, five destination microplates are used. 

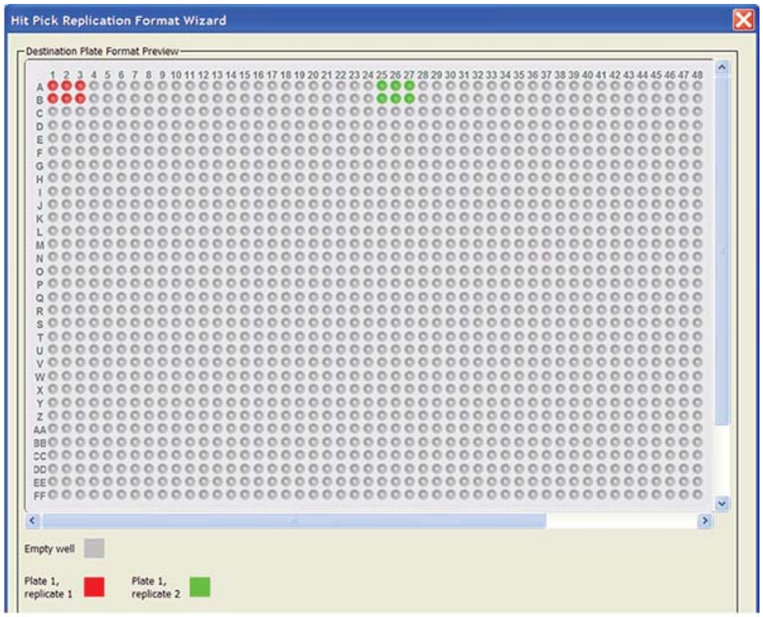
Transfer mode	Description
Minimize the number of destination plates used	The destination microplate will contain as many source contents as allowable to minimize the number of destination microplates used. For example, suppose you want to transfer contents from five 384-well source microplates into one or more 1536-well destination microplates. One 1536-well microplate can accommodate contents from four 384-well microplates. The contents from the fifth source microplate will be transferred into a second 1536-well destination microplate. *Note:* In this example, only two destination microplates are used. 

7 Select **Don't place replicates of the source plates on the same destination** if you want to place replicates in different destination microplates. Do not select the option if you want to place replicates in the same destination microplate.

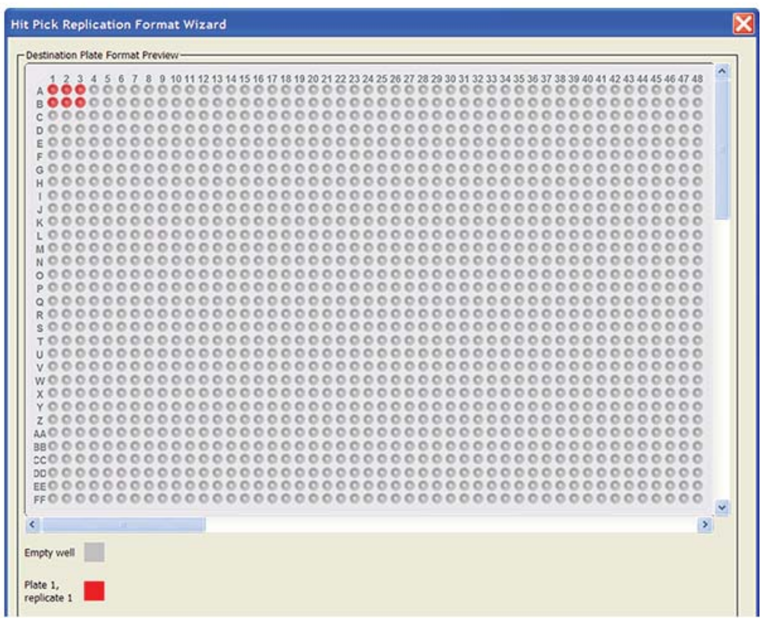
In the following example, the Quadrant transfer mode is used and the **All destination plates will have identical format/patterning option** is selected. Two replicates are requested from each source microplate.

In example a, the two replicates are placed in the same destination microplate. In example b, only one copy is placed in each destination microplate.

a)



b)



8 In the **Select Source Wells to Transfer** area, select the source wells from which to transfer the contents.

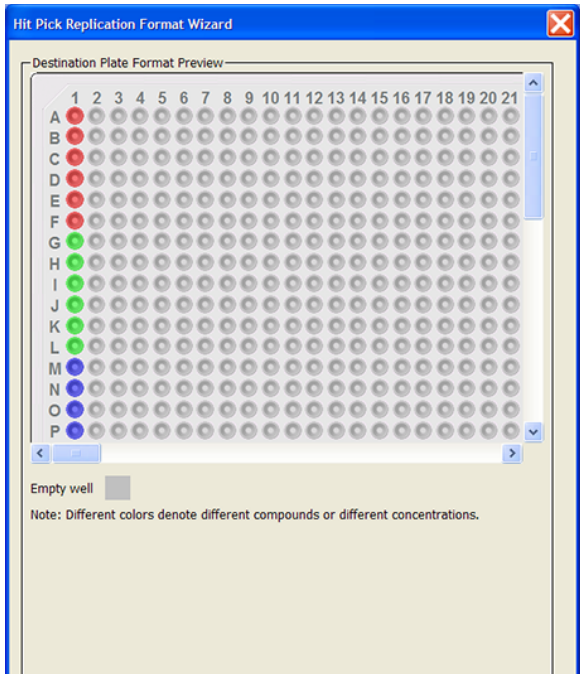
9 Click **Next** to preview the layout of the first destination microplate. The wells are color coded so you can see the replication pattern in the microplate.

To see the layout of all of the destination microplates, compile the protocol and check the output file. Alternatively, you can run the protocol in simulation mode and check the Pipette Log. For more information, see "Verifying source-destination transfers" on page 496.

Note that the layout shown in the preview might differ from the actual layout at run time if:

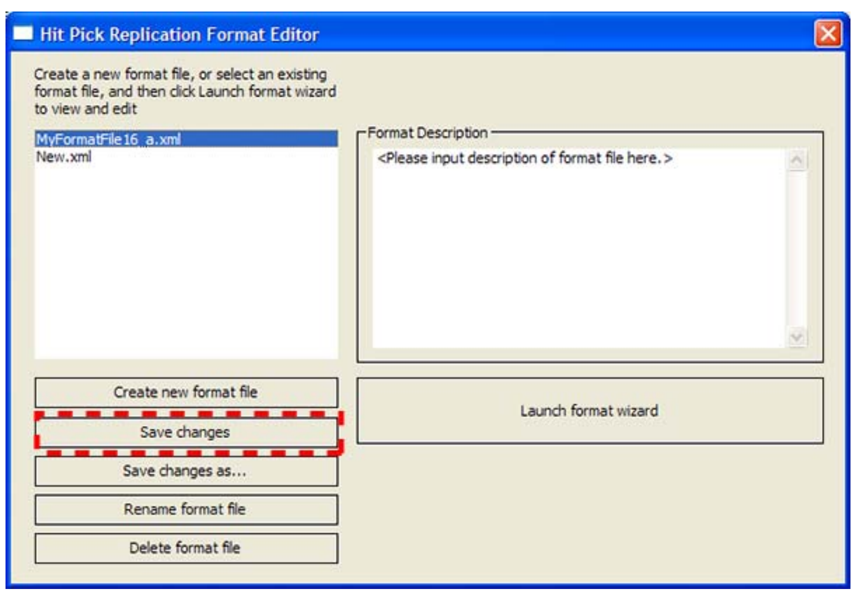
- The destination microplate has more wells than the source microplate.

- More than four source microplates will be processed during the run.
- You have selected **All destination plates will have identical format/patterning** in step 6.
- You have cleared the **Don't place replicates of source plates on the same destination** check box in step step 7.



When you are done, click **Finish.**

10 Back in the **Hit Pick Replication Format Editor,** make sure the correct format file name is selected, and then click **Save changes.** If you altered an existing format file, you can click **Save changes as** to save it as a different file.



11 Click the close button ( ) at the top right corner of the dialog box to return to the Hit Pick Routine Wizard. To continue in the Hit Pick Routine Wizard, return to step 3 in "Adding the Hit Pick Replication task" on page 470.

Verifying source-destination transfers

Before you start an actual run, you can verify that the source-destination transfers are correct using either output files or the Pipette Log.

Using output files

Compile your protocol to generate output files. The files list source-destination transfers by well and show the volume transferred. One output file is generated per source-destination microplate combination. For example, if contents from Source 1 are transferred to Destinations 1 and 2, then two output files are created, one for Source 1/Destination 1, and another for Source 1/Destination 2.

Output files are in the CSV format and are stored in the following folder:

...\VWorks Workspace\hit picking\output files

The following example shows an output file displayed in Excel. The first row shows the source and microplate IDs. Subsequent rows show the source wells, the corresponding destination wells, and the volume transferred.

	A	B	C
1	Plate2	1	Volume
2	Source Well	Destination Well	Volume in µL
3	C6	E1	2.5
4	C6	F1	2.5
5	B12	G1	2.5
6	B12	H1	2.5
7	M1	I1	2.5
8	M1	J1	2.5

Using the Pipette Log

Run your protocol in simulation mode and check the Pipette Log in the log area. The Pipette Log records every pipetting event that occurs, including aspirate location, dispense location, and volume information. For details, see "Pipette Log" on page 633.

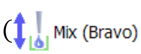
Related information

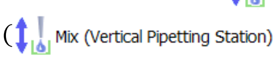
For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Creating a protocol	• "Creating a protocol: basic procedure" on page 13
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Dilute to Final Volume task	"Dilute to Final Volume (Bravo)" on page 442

For information about...	See...
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
Mix task	"Mix (Bravo, Vertical Pipetting Station)" on page 497
Pipette Log	"Pipette Log" on page 633
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Mix (Bravo, Vertical Pipetting Station)

Description

The Mix (Bravo) (

) and Mix (Vertical Pipetting Station) (

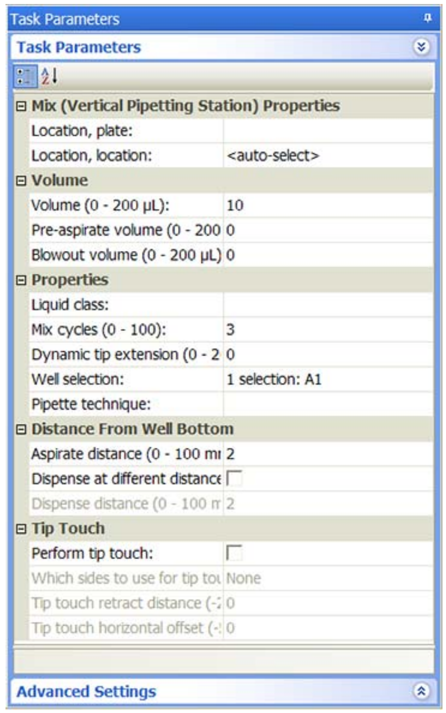
) tasks aspirate and dispense liquid multiple times to mix it. The task allows you to specify different well-bottom distances for the aspirate and dispense actions.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Task parameters

Note: The task parameters for Mix (Bravo) and Mix (Vertical Pipetting Station) are identical.

After adding the Mix task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Mix task.
Location, location	The location at which the Mix task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Volume (0–200 µL)	The volume of liquid to be mixed in each well.
Pre-aspirate volume (0–200 µL)	The volume of air to be drawn before the pipette tips enter the liquid.
Blowout volume (0–200 µL)	Specifies the volume of air to dispense after the main volume has been dispensed while the tips are still in the wells. Typically, the blowout volume is the same as the pre-aspirate volume.
Liquid class	The pipetting speed and accuracy. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.

Parameter	Description
Mix cycles ((0–100)	Specifies how many times to repeat the aspirate-and-dispense cycle.
Dynamic tip extension (0–20 mm/µL)	The rate at which the pipette head moves during the Aspirate task. The software calculates the distance over which the tips will move without crashing. Use dynamic tip extension to prevent spills as the pipette tips displace the liquid. To move the tips: • *At the same rate as the volume change.* Calculate dynamic tip extension (DTE) as follows: DTE = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTE > 1/A • *Slower than the volume change.* DTE < 1/A The starting and ending positions can be calculated as follows: $(V_{aspirated} * DTE) + Distance_{well\ bottom}$ *Note:* Instead of a negative aspirated volume, the software automatically moves downward toward the well bottom with each aspirate action.
Well selection	The wells at which the Dispense task occurs. Click the field, and then click the ... button to select the wells in the Well Selection dialog box. Use this parameter only if the pipette head has fewer tips than the number of wells in the microplate, or if you are in single-row or single-column mode.
Pipette technique	The pipette location offset you want to use for the Dispense task. The list of pipette techniques are defined in the Pipette Technique Editor.
Aspirate distance (0–100 mm)	The distance between the end of the pipette tips and the well botttoms during the aspirate action. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.

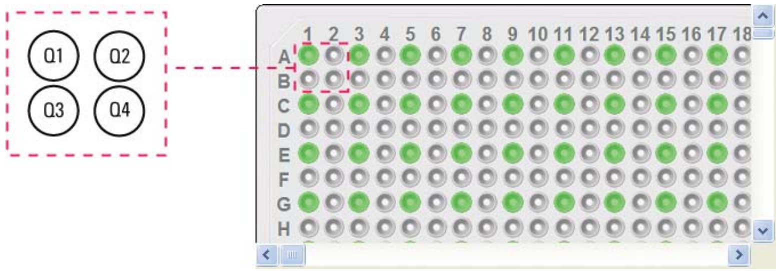
Parameter	Description
Dispense at different distance	The option to dispense at a pipette tip height that is different than the aspirate distance. Select the check box to enter a value for the dispense distance.
Dispense distance (0–100 mm)	The distance between the end of the pipette tips and the well bottoms during the dispense action.
Perform tip touch	The option to touch the pipette tip on one or more sides of the well.
Which sides to use for tip touch	The side or sides of the well to use during tip touch: North, South, East, West, North/South, West/East, West/East/South/North.
Tip touch retract distance (–20 to 50 mm)	The vertical distance for the pipette tips to rise before touching the sides of the wells.
Tip touch horizontal offset (–5 to 5 mm)	The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition. For example, if you set a value of: • 0, the tips move a horizontal distance equal to the well radius • > 0, the tips attempt to move past the well radius, which results in a more forceful tip touch • < 0, the tips move a distance less than the radius of the well, resulting in a lighter tip touch

Quadrant pattern well selection

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head. The following table lists the types of pipette heads and the number of accessible quadrants in various microplates.

Pipette head channels/ pin tool pins	Microplate	Number of quadrants
96	96-well	1
	384-well	4
	1536-well	16
384	384-well	1
	1536-well	4
1536 (pin tool only)	1536-well	1

The following diagram demonstrates the concept of quadrants. The diagram shows a portion of a 384-well microplate and highlights the four quadrants (Q1, Q2, Q3, and Q4) that are accessible by the A1 tip of a 96-channel pipette head. Notice that the green color highlights all of the quadrant 1 (Q1) wells across the microplate.



Instead of a column- or row-wise pattern, you can select a quadrant pattern during well selection.

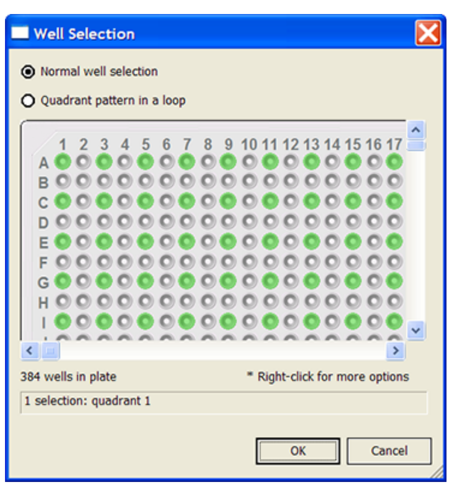
The quadrant pattern option is available only if:

- The number of channels in the pipette head (or pins in a pin tool) is fewer than the number of wells in the microplate. For example, you can use a 96-channel pipette head to dispense liquid into a 384-well microplate or 1536-well microplate.
- All the channels are selected in the Set Head Mode task when using a pipette head. (The Set Head Mode task is not an option when using a pin tool).
- The liquid-handling task is inside a loop.

IMPORTANT If you select a quadrant pattern, specifications in the Well Selection dialog box will override `task.Wellselection` values assigned in the Advanced Settings area.

To select a quadrant pattern:

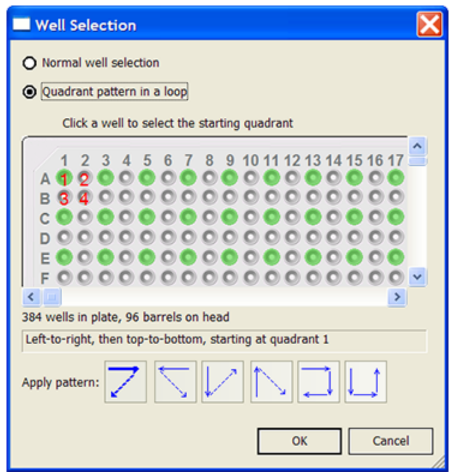
1 In the **Task Parameters** area, click the **Well selection** parameter box, and then click the Browse button. The Well Selection dialog box opens. By default, the **Normal well selection** option is selected. This option is used for column- and row-wise liquid-handling patterns.



2 Select **Quadrant pattern in a loop.** The contents of the dialog box change. Notice the following:

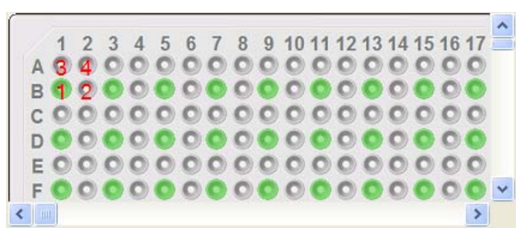
- Red numbers (1 through 4) appear on wells A1, A2, B1, and B2. The numbers indicate the pipetting sequence: 1 is the starting well, and 4 is the last well. In the following example, the sequence is A1, A2, B1, B2.
- Green wells indicate the starting well in the pipetting sequence.
- Pattern buttons at the bottom of the dialog box indicate the movement of the pipette channels. The movement description is provided in the text box above the buttons.

  Note: The last two patterns are unavailable if a group contains 16 wells. For example, the last two patterns are not available if you have a 96-well pipette head and a 1536-well microplate.



3 Select the starting well. The well becomes green and is labeled 1.

In the following example, the third quadrant (B1 well) is selected.



4 Click a pattern button to specify the pipette channel movement. After you click a pattern, the red numbers in the graphic are updated to show the sequence.

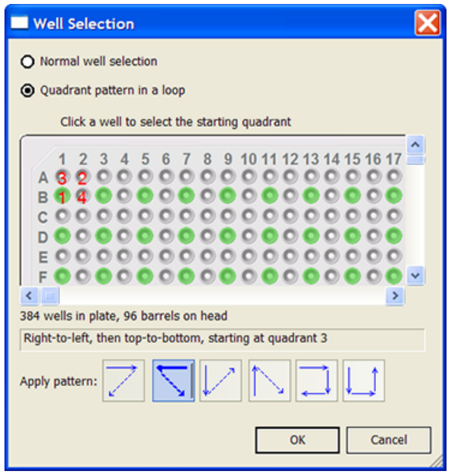
In the following example, the second pattern is selected (right-to-left, then top-to-bottom). The third quadrant (B1) is the starting well. The resulting movement is:

Quadrant 3 (B1)

Quadrant 2 (A2)

Quadrant 1 (A1)

Quadrant 4 (B2)



5 When you are finished, click **OK** to save the changes and return to the VWorks window.

Example: Mix the contents in the destination microplate on the Bravo Platform

Goal

Aspirate contents from a source microplate (Source 1), dispense into a destination microplate, and then mix the contents in the destination microplate. Use the default Mix parameters.

Implementation

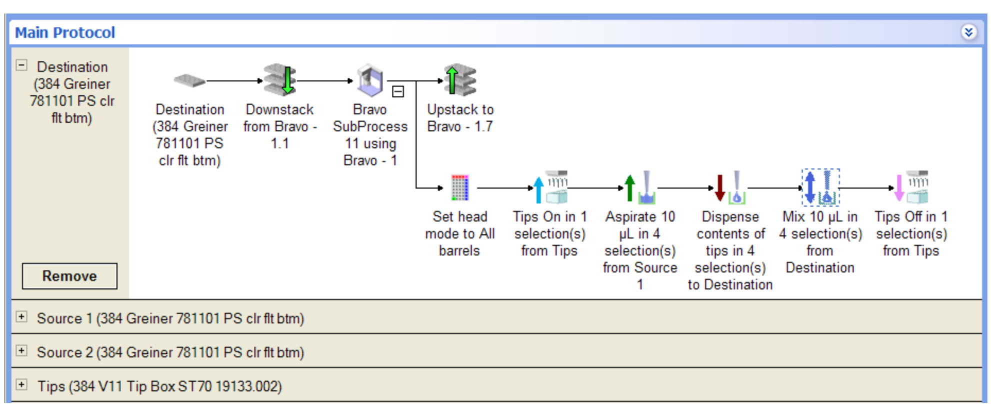
The Bravo deck is physically set up as follows:

- The destination microplates are at Bravo deck location 1.
- The source microplate is at deck location 2.
- The tipbox is at deck location 9.

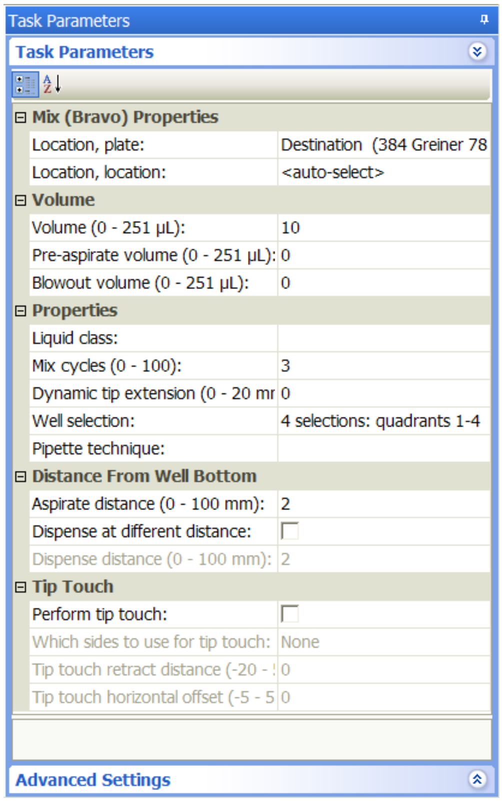
In the protocol, the following are added:

- Process for the destination microplate
- Configured labware for the source microplate
- Configured labware for the tipbox

In the Destination plate process, a Bravo subprocess is added. Within the subprocess, a Mix task is added after the Aspirate and Dispense tasks as shown in the following example.



In the Mix Task Parameters area, Destination is selected, because the goal is to mix the contents in the destination microplate.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
Set Head mode task	"Set Head Mode (Bravo)" on page 529
Tips On task	"Tips On (Bravo, Vertical Pipetting Station)" on page 542
Tips Off task	"Tips Off (Bravo, Vertical Pipetting Station)" on page 539
Pipette technique	"Specifying pipetting techniques" on page 613

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

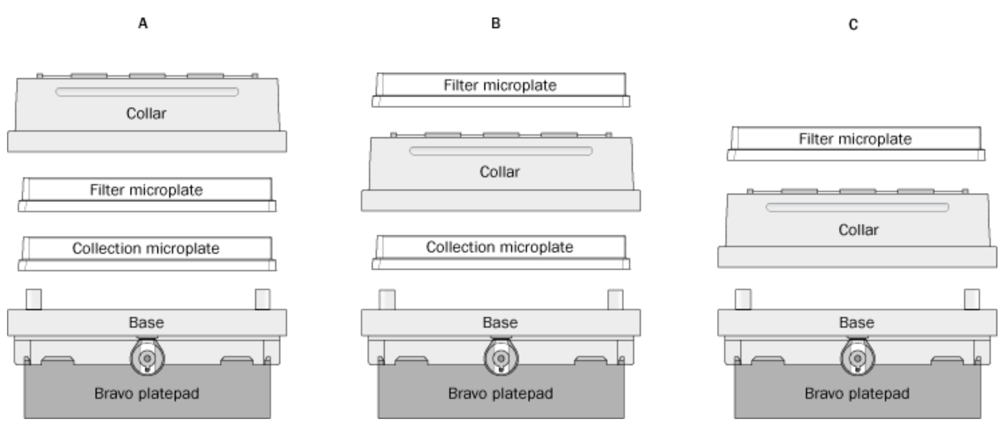
Move and Filter Plate (Bravo)

Description

The Move and Filter Plate (Bravo) task (Move and filter plate (Bravo)):

- Moves a specified filter microplate from its current location to the Vacuum Filtration Station.
- Turns on and turns off the vacuum.

Use the Move and Filter Plate task if the Vacuum Filtration Station has configuration B or C shown in the following diagram. In this configuration, the filter microplate is not part of the station assembly.



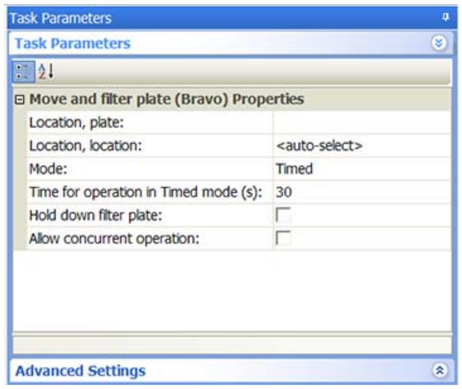
If the Vacuum Filtration Station has configuration A, where the filter microplate is part of the assembly (it sits under the collar), use the Toggle Vacuum task to turn on and turn off the vacuum. See "Toggle Vacuum (Bravo, Vertical Pipetting Station)" on page 545.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

Task parameters

Note: The Move and Filter task parameters displayed can vary, depending on the type of vacuum pump used with the Vacuum Filtration Station.

After adding the Move and Filter task at the desired point in the Bravo Subprocess, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The name of the filter microplate. The robot will move this filter microplate to the Vacuum Filtration Station (specified by the Location, location parameter).
Location, location	The deck location of the Vacuum Filtration Station. Select <auto-select> if you want the robot to place the filter microplate at the first-available Vacuum Filtration Station. If more than one Vacuum Filtration Station is on the deck, select the specific deck location.
Mode	The action of the task: • *On*. Turns on the vacuum. • *Off*. Turns off the vacuum. • *Timed*. Turns on the vacuum timer. You must specify the vacuum length of time. If you plan to time the filtering process, add only one Move and Filter Plate task (the task turns on the vacuum, and at the end of the time period, the vacuum turns off automatically). If you are not timing the filtering process, add two Move and Filter Plate tasks in the protocol (one to turn on the vacuum and the other to turn off the vacuum). *Note:* If the filtering process is not timed, the protocol can perform other tasks in parallel.

Parameter	Description
When filtration timing begins (ME4C Vario Vacuum Pump Unit only)	The different options for when to start timing the filtration process: • When pressure is achieved • When the pump starts This parameter is only used if the ME4C Vario Vacuum Pump Unit is configured as part of the Vacuum Filtration Station.
Time for operation in Timed mode	The length of time, in seconds, you want to leave the vacuum on. At the end of the period, the vacuum will turn off.
Hold down filter plate	The option to have the robot hold down the filter microplate when the vacuum is turned on to ensure a secure vacuum seal.
Allow concurrent operation	The option to permit the accessory to operate simultaneously with other tasks.
Time allowed to reach pressure(s) (ME4C Vario Vacuum Pump Unit only)	The length of time, in seconds, to allow the vacuum to reach the specified target pressure. An error message displays if the target pressure is not reached within the time specified. This parameter is only used if the ME4C Vario Vacuum Pump Unit is configured as part of the Vacuum Filtration Station.
Target pressure (ME4C Vario Vacuum Pump Unit only)	The desired vacuum pressure, in mbar, Torr, or hPa. The pressure unit is set on the Vario vacuum pump. This parameter is only used if the ME4C Vario Vacuum Pump Unit is configured as part of the Vacuum Filtration Station.
Vent delay (ME4C Vario Vacuum Pump Unit only)	The length of time, in seconds, to wait for the air pressure under the filter to equalize with the ambient air pressure. This parameter is only used if the ME4C Vario Vacuum Pump Unit is configured as part of the Vacuum Filtration Station.

Example

Goal

Assemble a Vacuum Filtration Station whose configuration is base-collection plate-collar. Move a filter microplate to the station. Filter for 30 seconds. Disassemble the station.

Implementation

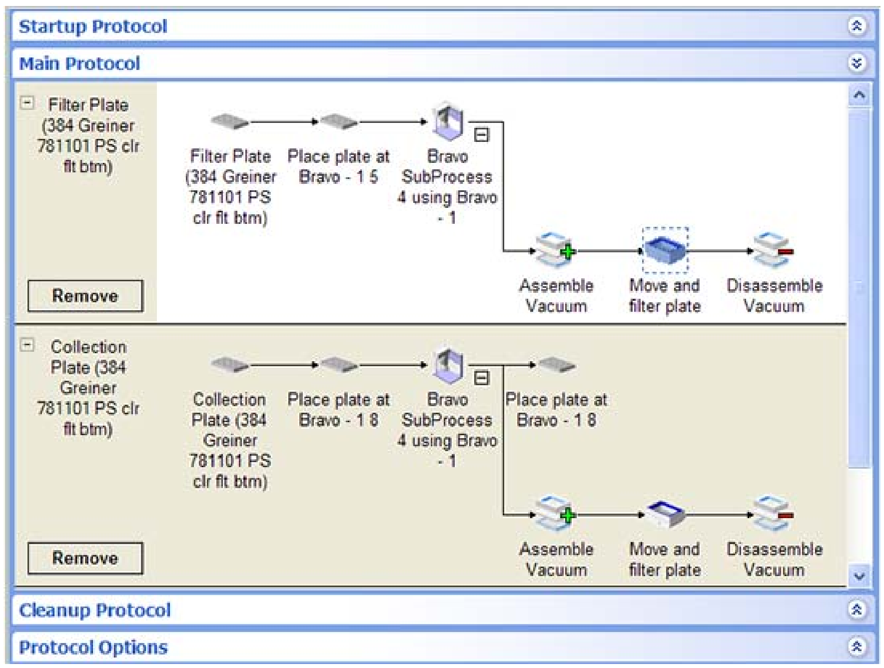
In Bravo Diagnostics, the Bravo deck is configured such that:

- The Vacuum Filtration Station will be assembled at deck location 3.

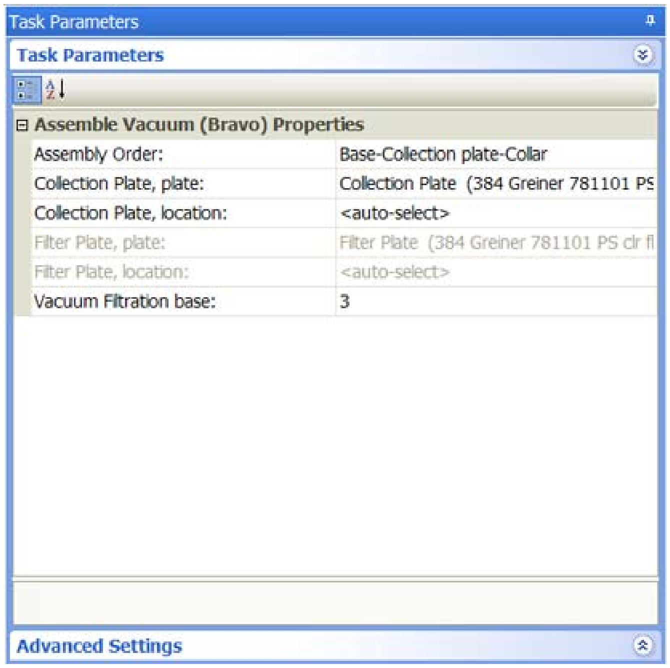
- The Vacuum Filtration Station base is at deck location 3.
- The collar is at deck location 9.
- During the disassembly process, the filter microplate will be placed at the first-available deck location.

In the protocol, the following are added as shown:

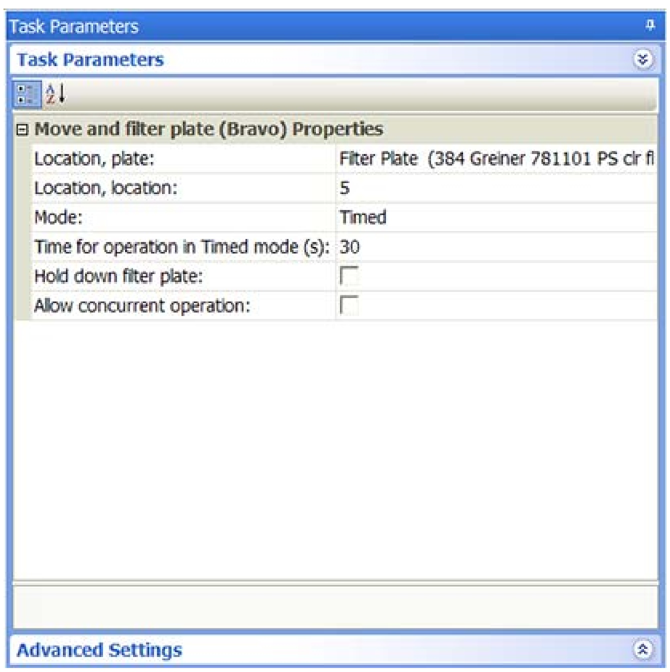
- Filter Plate process
- Collection Plate configured labware (at deck location 8)
- Place Plate (to place the Filter Plate at location 5), Assemble Vacuum, Move and Filter Plate, and Disassemble Vacuum tasks



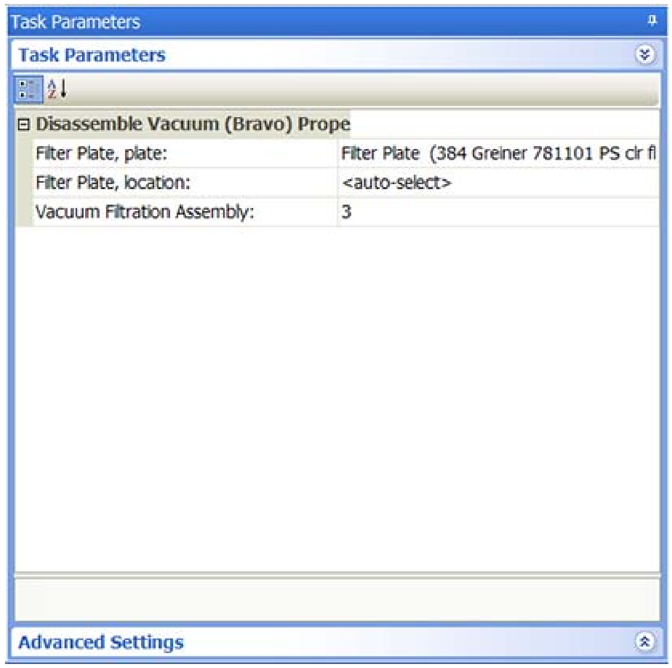
The Assemble Vacuum task parameters are set as follows:



The Move and Filter Plate task parameters are set as follows:



Set the Disassemble Vacuum task parameters as follows:



The resulting protocol will run as follows:

1 Acknowledge that the filter microplate (Filter Plate) is starting at deck location 5.
2 Acknowledge that the collection microplate (Collection Plate) is starting at deck location 8.
3 Assemble the Vacuum Filtration Station (base-collection plate-collar):
   a Move the Collection Plate from deck location 8 and place it on top of the base at deck location 3.
   b Move the collar from deck location 9 and place it on top of the Collection Plate at deck location 3.
4 Move the Filter Plate from deck location 5 and place it on top of the assembled Vacuum Filtration Station at deck location 3.
5 Turn on the vacuum for 30 seconds, and then turned off the vacuum.
6 Disassemble the Vacuum Filtration Station:
   a Move the Filter Plate to an available deck location.
   b Move the collar back to deck location 9.
   c Move the Collection Plate back to deck location 8.

Related information

For information about...	See...
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53

For information about...	See...
Assemble Vacuum task	"Assemble Vacuum (Bravo)" on page 439
Disassemble Vacuum task	"Disassemble Vacuum (Bravo)" on page 449
Toggle Vacuum task	"Toggle Vacuum (Bravo, Vertical Pipetting Station)" on page 545
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Pin Tool (Bravo, Vertical Pipetting Station)

Description

The Pin Tool (Bravo) (Pin Tool (Bravo)) and Pin Tool (Vertical Pipetting Station) (Pin Tool (Vertical Pipetting Station)) tasks can be used to perform low-volume transfers of a fixed volume using a pin tool. You can use the Pin Tool task repeatedly in a protocol subprocess to perform all the pin-tool-related steps, such as:

- Pin Tool—Adsorb
- Pin Tool—Dispense
- Pin Tool—Wash
- Pin Tool—Blot
- Pin Tool—Mix

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Before you begin

Ensure the following:

- The Bravo or Vertical Pipetting Station device profile specifies an appropriate pin tool. To create or edit a profile, see the user guide for the applicable device.

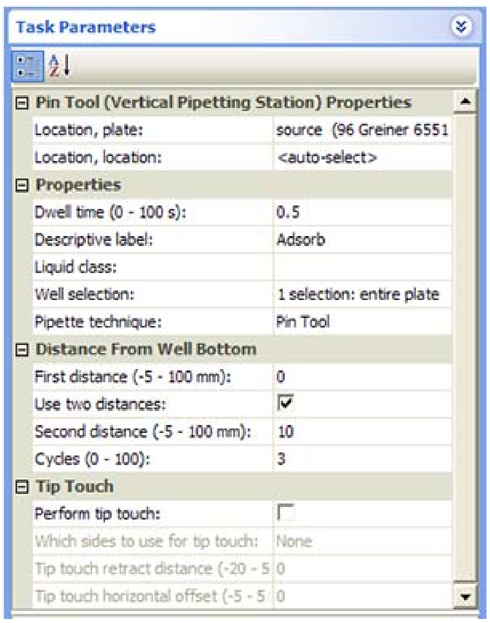
  IMPORTANT Ensure the pin tool teachpoints are set up in the same manner as a fixed-tip pipette head.

- The protocol includes the SubProcess (Bravo) or SubProcess (Vertical Pipetting Station).

Task parameters

Note: The Pin Tool task parameters for the Bravo Platform and Vertical Pipetting Station are identical.

After adding the Pin Tool task at the desired point in the subprocess, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Pin Tool task.
Location, location	The location at which the Pin Tool task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.

Parameter	Description
Dwell time (s)	The time duration that the pins remain at the specified height (First distance or Second distance) within the well. For example, you might start with the following values: • Adsorb, Dispense into fluid, or Mix—0.5 s or longer for more viscous fluids • Blot—2 s, or longer for more viscous fluids
Descriptive label	A text label that you can add to the task icon in the protocol. Click the arrow in the Descriptive label box to choose an option. The options include: • Enter a JavaScript variable or script. • Use a predefined label: Adsorb, Dispense, Wash, Blot, or Mix • Type your own label in the box.
Liquid class	A parameter that you can use to control the accuracy and the speed of the pin tool as it moves into and out of the wells. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.
Well selection	The wells at which the Pin Tool task occurs. Use this parameter only if the pin tool has fewer pins than the number of wells in the microplate, for example, a 96-pin pin tool and a 384-well microplate. Click in the parameter box, and then click the Browse button to select the wells in the Well Selection dialog box.
Pipette technique	The pipette location offset you want to use for the Pin Tool task. The list of pipette techniques are defined in the Pipette Technique Editor.

Parameter	Description
First distance (mm)	The first height for the pin tool during the Pin Tool task. The value is the distance between the pin tips and the well bottoms. For example, during an adsorb step, you might set this value to 0 mm so that the pin tips touch the bottom of the wells. This parameter can affect the quantity adsorbed. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the pins at the correct distance from the well bottom.
Use two distances	The option to specify a second height for the pins during the Pin Tool task. For example, you could cycle the pin positions between two heights within the wells repeatedly to perform mixing or to wash the pins. Default: Not selected
Second distance (mm)	The distance between the pin tips and the well bottoms at the second height for the pins. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the pins at the correct distance from the well bottom.
Cycles	Available if you select the Use two distances option. The Cycles parameter sets the number of times to move the pins repeatedly to the two heights, for example to perform mixing or to wash the pins.
Perform tip touch	The option to touch the pins on one or more sides of the well, or to enable the pins to make lateral stirring moves inside the fluid, for example during a wash task.
Which sides to use for tip touch	The side or sides of the well to use during tip touch: North, South, East, West, North/South, West/East, West/East/South/North.
Tip touch retract distance (mm)	The vertical distance for the pins to move before moving laterally within the well, where • 0 is the vertical distance equal to the well bottom • > 0 is the vertical distance the pins rise above the bottom • < 0 is the vertical distance the pins attempt to move past the well bottom

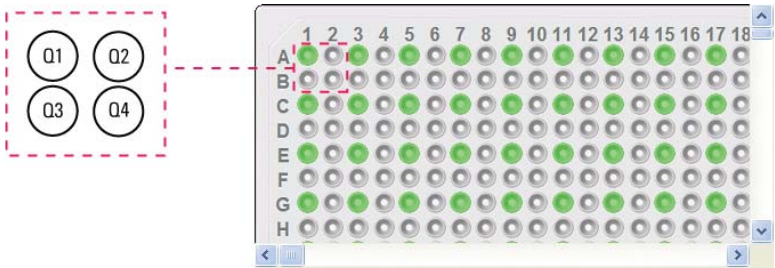
Parameter	Description
Tip touch horizontal offset (mm)	The horizontal distance that the pins move. The value is based on the well diameter specified by the labware definition, where • 0 is a distance equal to the well radius • > 0 is the distance the pins attempt to move past the well radius, which results in a more forceful tip touch • < 0 is a distance less than the radius of the well, resulting in a lighter tip touch or no tip touch

Quadrant pattern well selection

A quadrant is an evenly spaced array of locations that are accessible by the tips on a pipette head. The following table lists the types of pipette heads and the number of accessible quadrants in various microplates.

Pipette head channels/ pin tool pins	Microplate	Number of quadrants
96	96-well	1
	384-well	4
	1536-well	16
384	384-well	1
	1536-well	4
1536 (pin tool only)	1536-well	1

The following diagram demonstrates the concept of quadrants. The diagram shows a portion of a 384-well microplate and highlights the four quadrants (Q1, Q2, Q3, and Q4) that are accessible by the A1 tip of a 96-channel pipette head. Notice that the green color highlights all of the quadrant 1 (Q1) wells across the microplate.



Instead of a column- or row-wise pattern, you can select a quadrant pattern during well selection.

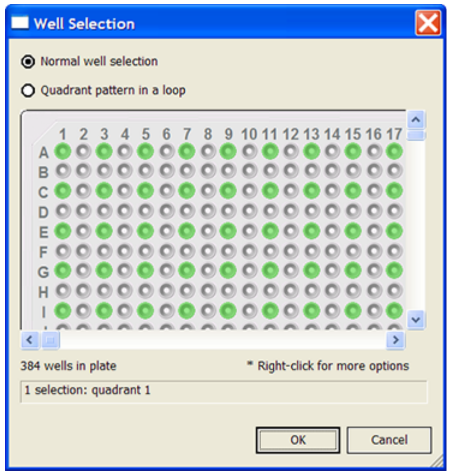
The quadrant pattern option is available only if:

- The number of channels in the pipette head (or pins in a pin tool) is fewer than the number of wells in the microplate. For example, you can use a 96-channel pipette head to dispense liquid into a 384-well microplate or 1536-well microplate.
- All the channels are selected in the Set Head Mode task when using a pipette head. (The Set Head Mode task is not an option when using a pin tool).
- The liquid-handling task is inside a loop.

IMPORTANT If you select a quadrant pattern, specifications in the Well Selection dialog box will override `task.Wellselection` values assigned in the Advanced Settings area.

To select a quadrant pattern:

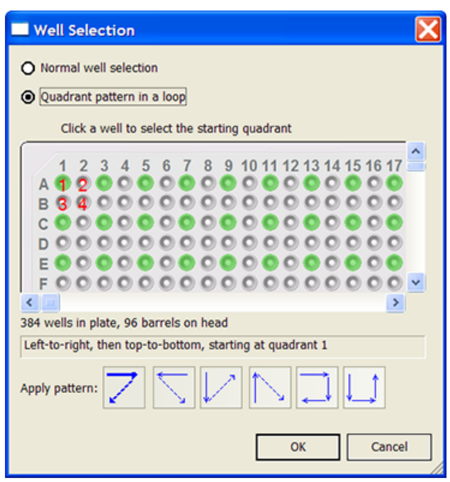
1 In the **Task Parameters** area, click the **Well selection** parameter box, and then click the Browse button. The Well Selection dialog box opens. By default, the **Normal well selection** option is selected. This option is used for column- and row-wise liquid-handling patterns.



2 Select **Quadrant pattern in a loop.** The contents of the dialog box change. Notice the following:

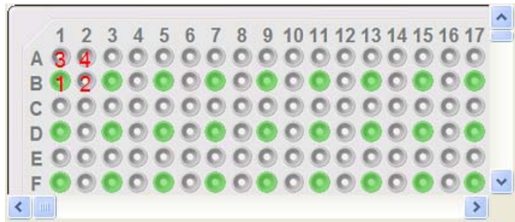
- Red numbers (1 through 4) appear on wells A1, A2, B1, and B2. The numbers indicate the pipetting sequence: 1 is the starting well, and 4 is the last well. In the following example, the sequence is A1, A2, B1, B2.
- Green wells indicate the starting well in the pipetting sequence.
- Pattern buttons at the bottom of the dialog box indicate the movement of the pipette channels. The movement description is provided in the text box above the buttons.

  Note: The last two patterns are unavailable if a group contains 16 wells. For example, the last two patterns are not available if you have a 96-well pipette head and a 1536-well microplate.



3 Select the starting well. The well becomes green and is labeled 1.

In the following example, the third quadrant (B1 well) is selected.



4 Click a pattern button to specify the pipette channel movement. After you click a pattern, the red numbers in the graphic are updated to show the sequence.

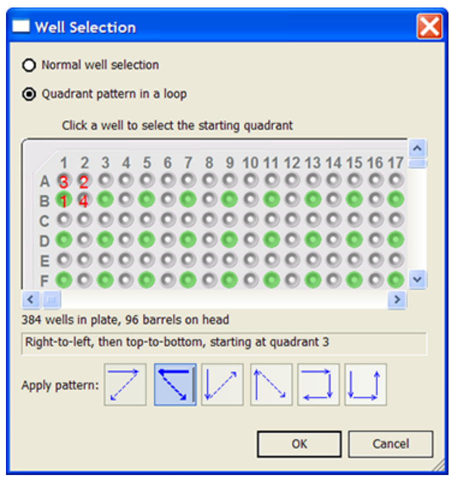
In the following example, the second pattern is selected (right-to-left, then top-to-bottom). The third quadrant (B1) is the starting well. The resulting movement is:

Quadrant 3 (B1)

Quadrant 2 (A2)

Quadrant 1 (A1)

Quadrant 4 (B2)



5 When you are finished, click **OK** to save the changes and return to the VWorks window.

Example: Pin Tool tasks on a Vertical Pipetting Station

Goal

Using a pin tool, transfer a small volume from a source microplate (Source 1) into a destination microplate, and wash and blot the pins.

Implementation

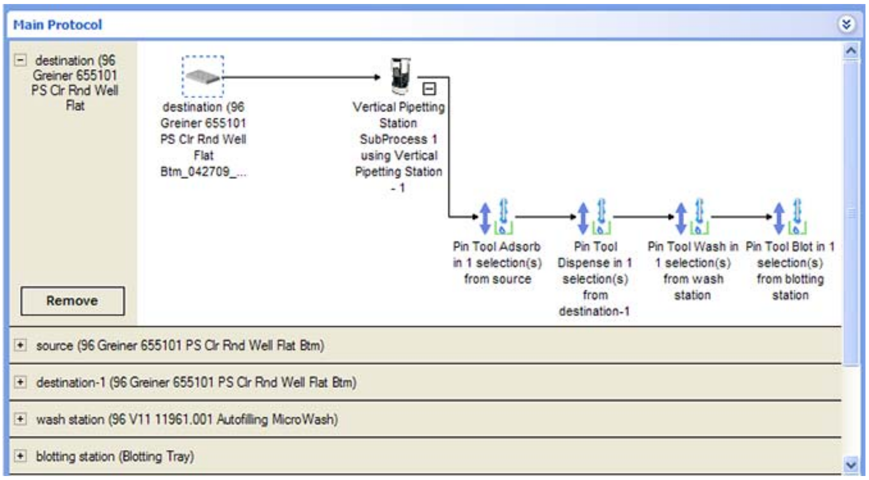
The Vertical Pipetting Station is physically set up as follows:

- The source microplate is on shelf 3.
- The destination microplate is on shelf 4.
- The wash station is on shelf 5.
- The blotting material is on shelf 6.

In the protocol, the following are added:

- Process for the destination microplate
- Configured labware for the source microplate
- Configured labware for the wash station
- Configured labware for the blotting station

In the process, a Vertical Pipetting Station subprocess is added. Within the subprocess, four Pin Tool tasks are added for Adsorb, Dispense, Wash, and Blot.



Parameter settings for each Pin Tool task in the example

- Adsorb
  - Dwell time 0.5 s
  - First distance 0 mm
  - Second distance 10 mm
  - Cycles 3
  - No tip touch
- Dispense (into dry microplate)
  - Dwell time 0.5 s
  - First distance –2 mm
  - No second distance
  - No tip touch
- Wash
  - Dwell time 0.5 s
  - First distance 0 mm
  - Second distance 10 mm
  - Cycles 3
  - Perform tip-touch on all sides with horizontal offset of 0 mm and at a retract distance of 2 mm
- Blot
  - Dwell time 2 s
  - First distance –2 mm
  - No second distance
  - No tip touch

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Liquid classes	*VWorks Automation Control Setup Guide*
Set Head Mode task	"Set Head Mode (Bravo)" on page 529
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Pump Reagent (Bravo, Vertical Pipetting Station)

Description

The Pump Reagent (Bravo) (Pump Reagent (Bravo)) and Pump Reagent (Vertical Pipetting Station) (Pump Reagent (Vertical Pipetting Station)) tasks fill or empty the Auto Filling Reservoir and Tip Wash Station (also known as MicroWash Reservoir) by pumping for a specified number of seconds or until the percent of maximum tared weight is reached. If the reservoir is on a Weigh Station or Weigh Shelf, the pump stops fluid flow when the target weight is reached. Otherwise, gravity drain is used to empty the reservoir.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Requirements

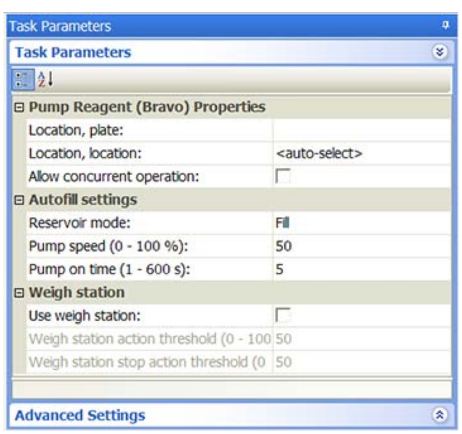
The following must be configured on the Bravo Platform or the Vertical Pipetting Station:

- Autofilling Reservoir or Tip Wash Station
- Pump Module
- Weigh Station or Weigh Shelf (optional)

Task parameters

Note: The task parameters for Pump Reagent (Bravo) and Pump Reagent (Vertical Pipetting Station) are identical.

After adding the Pump Reagent task at the desired point in the subprocess, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Pump Reagent task.
Location, location	The location at which the Pump Reagent task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Reservoir mode	The action of the task: • Fill • Empty
Pump speed	The speed, in percent of maximum, at which to pump the reagent.
Pump on time	The duration of the pumping time, in seconds.
Use weigh station/shelf	The option to use the Weigh Station or Weigh Shelf.
Weigh station/shelf action threshold	The minimum fluid weight, in percent of the full weight that was calibrated on the Weigh Station or Weigh Shelf. For example, you can set the minimum threshold at 45% so that when the fluid reaches 45% of the full weight, fluid starts to pump into the reservoir.
Weigh station stop action threshold	The maximum fluid weight, in percent of the full weight that was calibrated on the Weigh Station or Weigh Shelf. For example, you can set the stop threshold at 60% so that when the fluid reaches 60% of the full weight, fluid starts to drain or pump out of the reservoir.
Allow concurrent operation	The option to permit the accessory to operate simultaneously with other tasks.

Example: Fill the Tip Wash Station after the Tip Wash task

Goal

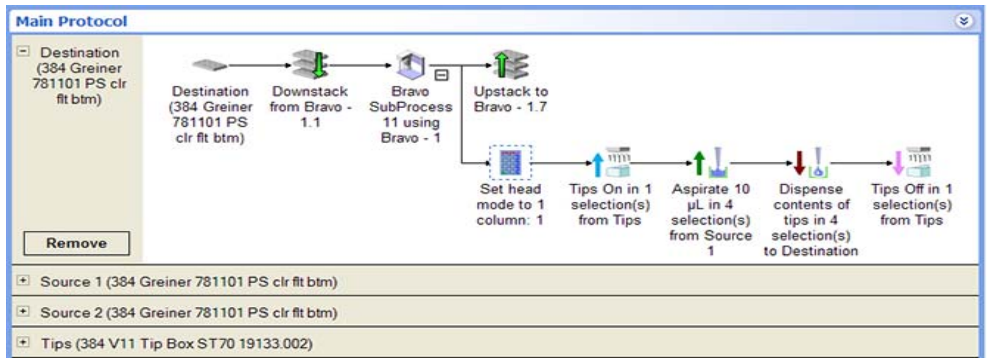
After some liquid-handling tasks, wash the tips, and then fill the Tip Wash Station.

Implementation

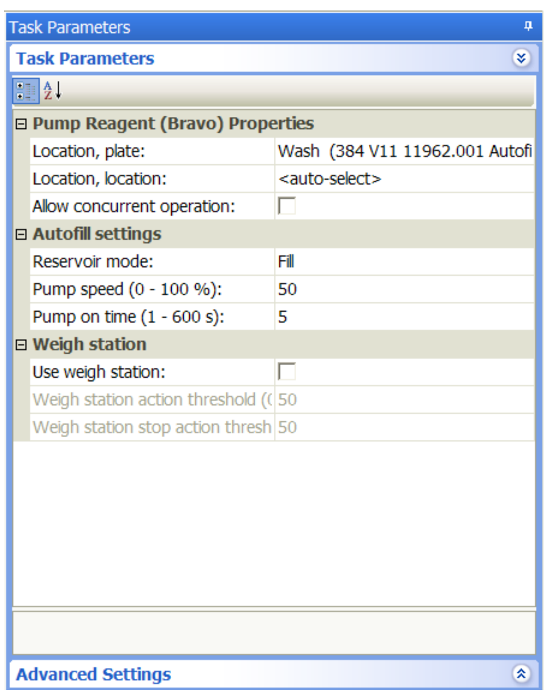
In Bravo Diagnostics, set up the Auto Filling Reservoir on a deck location. Associate the reservoir with one or more pump modules.

In the protocol, add a configured labware for the Auto Filling Reservoir in addition to the other processes and configured labware required by the protocol. In the example shown, the configured reservoir is called Wash.

In the Bravo subprocess where the liquid-handling tasks are specified, a Pump Reagent task is added after the Wash Tips task. After adding the Pump Reagent task, the task name changes to Fill or empty a reservoir in the protocol, as shown in the following example.



In the Pump Reagents Task Parameters area, Wash (the name for the Auto Filling Reservoir) is selected, because the goal is to pump fluid into the Auto Filling Reservoir.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
Set Head Mode task	"Set Head Mode (Bravo)" on page 529
Tips On task	"Tips On (Bravo, Vertical Pipetting Station)" on page 542
Tips Off task	"Tips Off (Bravo, Vertical Pipetting Station)" on page 539
Wash Tips task	"Wash Tips (Bravo, Vertical Pipetting Station)" on page 550
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Serial Dilution (Bravo, Vertical Pipetting Station)

Description

The Serial Dilution (Bravo) (Serial Dilution (Bravo)) and Serial Dilution (Vertical Pipetting Station) (Serial Dilution (Vertical Pipetting Station)) tasks allows you to set up serial dilution in a microplate using a single task. You use the Serial Dilution wizard to set up the task parameters. The end result is a sequence of Aspirate, Dispense, and optional Mix tasks that produce a linear or non-linear concentration gradient in selected wells.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

Task is available for...	Task is available in...
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

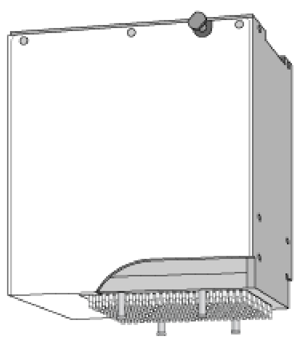
Requirements

Setup

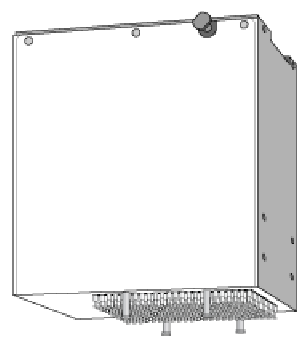
Make sure you:

- Check that the Vertical Pipetting Station device has the 8- or 16-channel serial dilution head installed. In general, the Vertical Pipetting Station device can perform single-column or single-row serial dilution only.
- Configure the labware on the Bravo deck or the Vertical Pipetting Station shelves in the software. For instructions, see "Configuring labware" on page 40.
- If you are using a Series III pipette head on the Bravo Platform, add a Set Head mode task before the Serial Dilution task to select the pipette channels. See "Set Head Mode (Bravo)" on page 529 for details.
- If you are using a Series II pipette head and you want to dilute by quadrant, check that the number of channels is fewer than the number of wells in the microplate.

Series III pipette head



Series II pipette head



The Serial Dilution task can be used with the following pipette head and microplate formats:

Series II pipette head	Microplate format
8-channel head	96, 384, 1536
16-channel head	384, 1536
96-channel head	384, 1536
384-channel head	1536
Series III pipette head	Microplate format
96-channel head with disposable tips	96, 384, 1536

Series II pipette head	Microplate format
384-channel head with disposable tips	384, 1536

Labware

IMPORTANT The Serial Dilution task can only be used with a microplate. The task cannot be used with a reservoir.

Make sure the serial dilution microplate meets the following requirements:

- A column, row, or quadrant contains the starting concentration of a compound to be diluted.
- One or more columns, rows, or quadrants contain the same amount of diluent.

Pipette-tip tracking

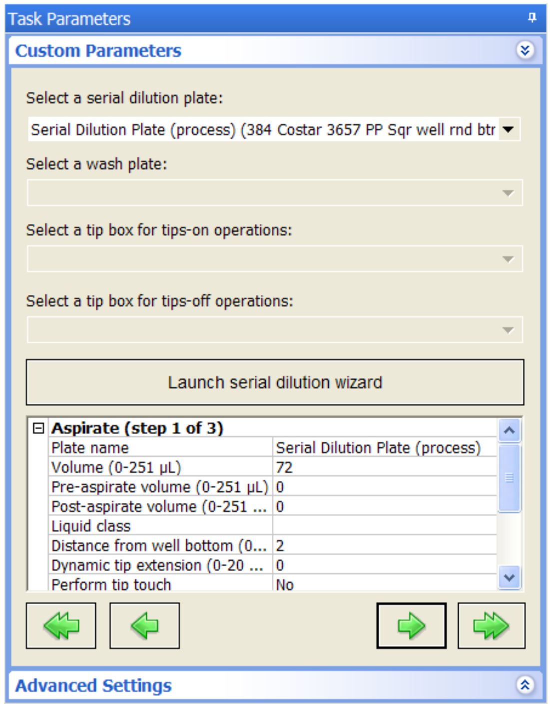
You can track pipette tip usage on the Bravo Platform and the Vertical Pipetting Station. To track pipette tip usage, turn on the tracking options in the following:

- Tips On task
- Tips Off tasks.
- Serial Dilution wizard, step 2

Custom parameters

Note: The custom parameters for Serial Dilution (Bravo) and Serial Dilution (Vertical Pipetting Station) are identical.

After adding the Serial Dilution task at the desired point in the protocol, set the following parameters in the **Custom Parameters** area:



Parameter	Description
Select a serial dilution plate	The labware involved in the Serial Dilution task.
Launch serial dilution wizard	The command that opens the Serial Dilution Wizard dialog box. See "Serial dilution wizard" on page 527 for instructions.

Note: The remaining task parameters will be filled in after you go through the Serial Dilution wizard.

Serial dilution wizard

The serial dilution wizard guides you through the serial dilution setup.

To use the wizard:

1. In the **Task Parameters** area, click **Launch serial dilution wizard**. The Serial Dilution Wizard dialog box opens.
2. Follow the instructions to set up the serial dilution.

   IMPORTANT In step 1 of the wizard, if you want the transfer volume to be determined by a concentration gradient, be sure to type a gradient factor.

   For example, if the concentration gradient is 2, then the concentration of the first dilution will be the concentration in the starting column C_1 divided by 2, or $C_1/2$. The concentration of the second dilution will be the concentration of the second column C_2 divided by 2, or $C_2/2$, and so on.

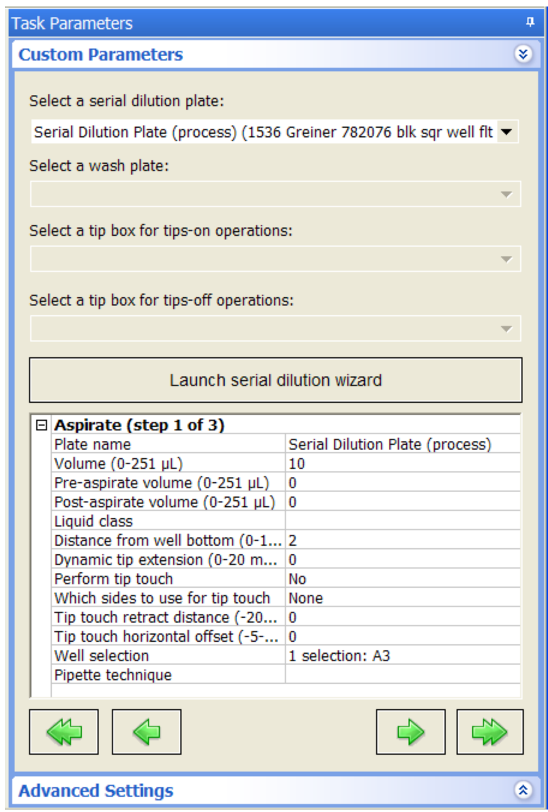
The upper range of the transfer volume is determined by the capacity of the pipette head and the well volume of the microplate.

IMPORTANT Select whether you want the software to track tip usage in Step 2 of the wizard.

IMPORTANT If you want to track tips, make sure you also select the tracking options in the Tips On and Tips Off tasks.

For the description of the Aspirate parameters, see “Aspirate (Bravo, Vertical Pipetting Station)” on page 392. For the description of the Dispense parameters, see “Dispense (Bravo, Vertical Pipetting Station)” on page 451. For the description of the Mix parameters, see “Mix (Bravo, Vertical Pipetting Station)” on page 497.

When you are finished setting up the serial dilution task, the parameters you specified in the wizard appear in the **Task Parameters** area. Review the parameters. To see additional parameters, click the arrow buttons under the parameter table. If you need to edit a parameter value, double-click in the box and type the new value.



Related information

For information about...	See...
Configured labware	“Planning labware use” on page 20
Static labware	“Planning labware use” on page 20

For information about...	See...
Startup Protocol	"Setting up Startup and Cleanup Protocol processes" on page 60
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Set Head Mode (Bravo)

Description

The Set Head Mode (Bravo) (Set Head Mode (Bravo)) task specifies the channels (or barrels) in the pipette head to be used for pipetting. You can select an m × n array of channels (barrels) for one of the following configurations:

- All of the pipette channels
- The first or last full column or row of pipette channels
- Multiple full columns or rows of pipette channels
- The first or last partial column or row of pipette channels
- Multiple partial columns or rows of pipette channels
- A single pipette barrel at the corner of the pipette head

This task should only be used if the Series III pipette head is installed.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess

Pipette head requirements

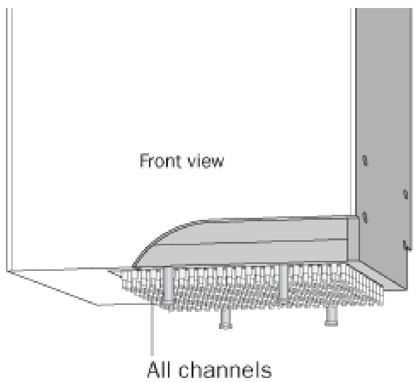
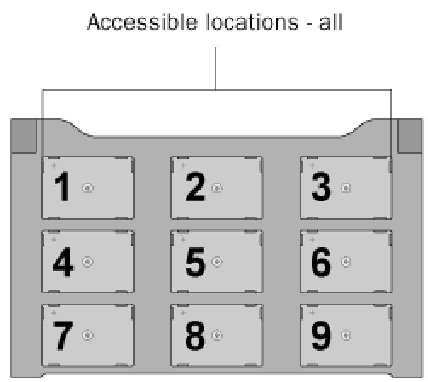
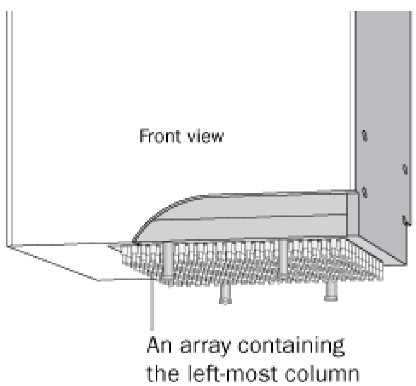
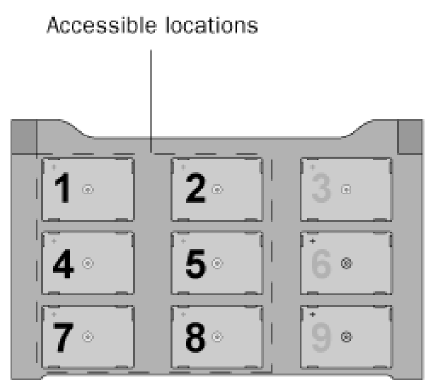
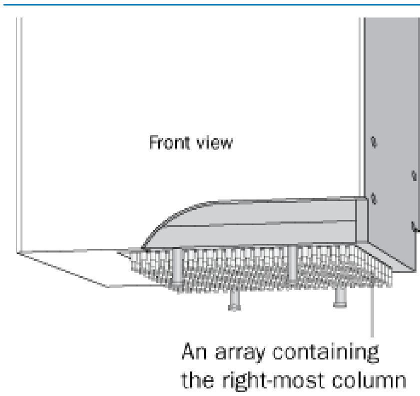
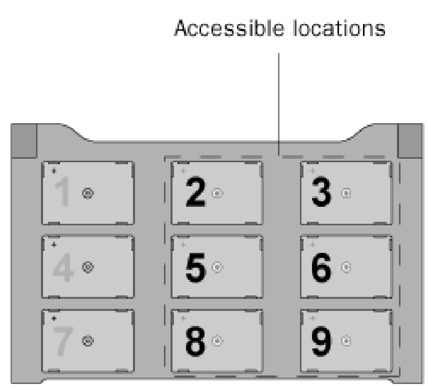
You can use the following pipette heads with a flexible array of pipette tips:

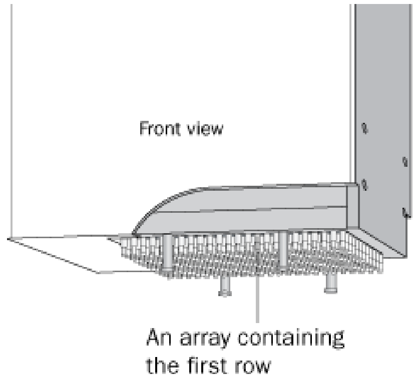
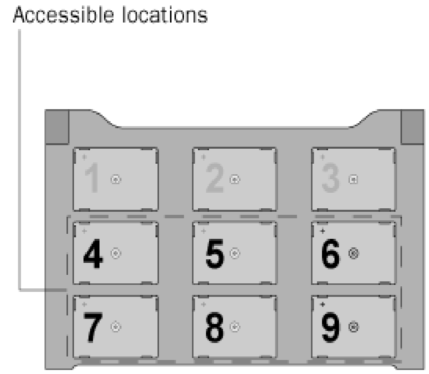
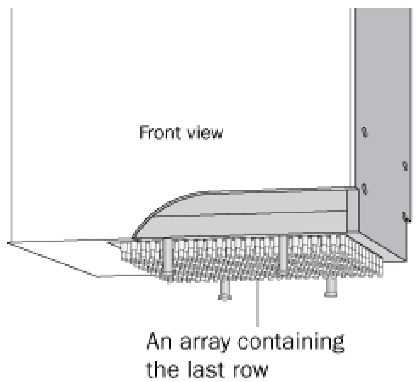
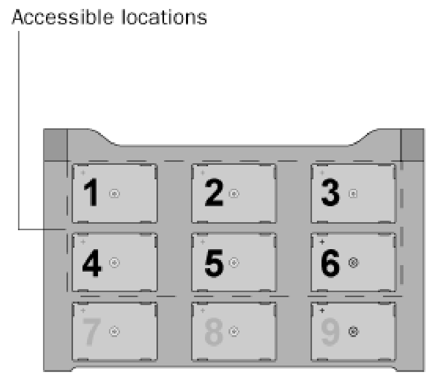
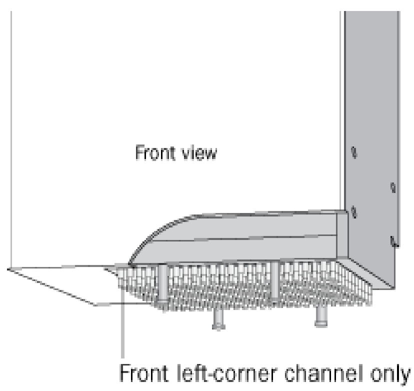
Pipette head	Microplate format
Series III 96-channel head with disposable tips	96, 384, or 1536

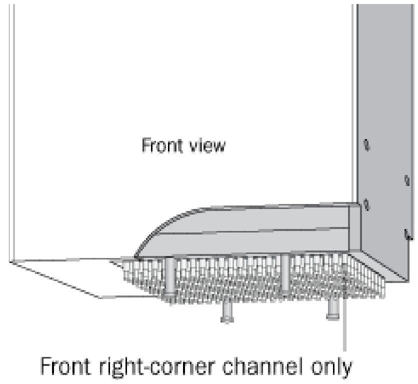
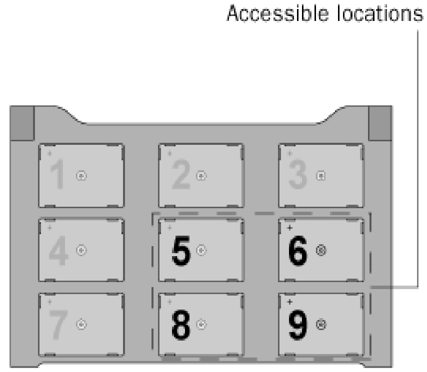
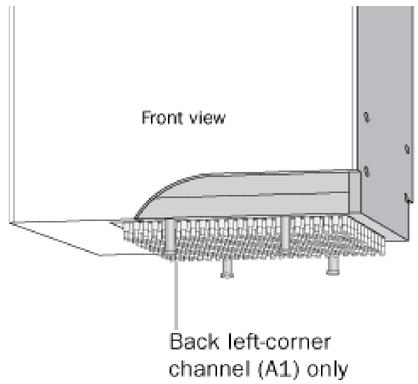
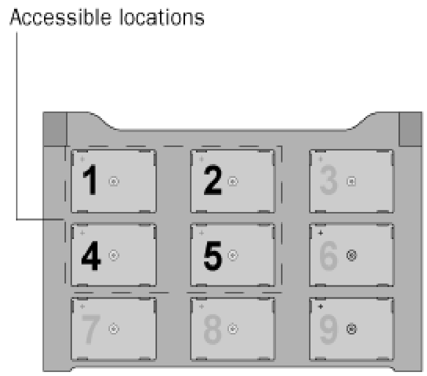
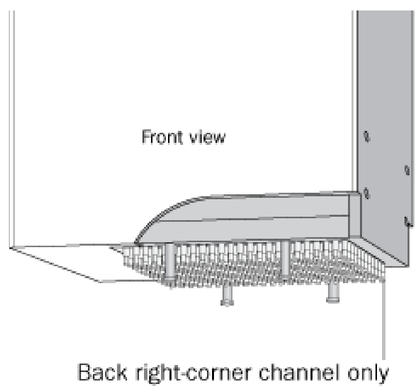
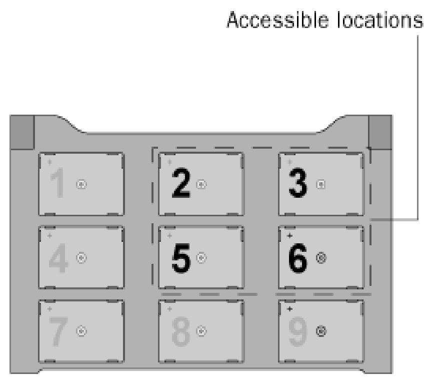
Pipette head	Microplate format
Series III 384-channel head with disposable tips	96, 384 or 1536

Accessible deck locations

The deck locations you can access depends on the pipette head channels you select. The following table shows the channel selections and corresponding deck location limits. Use this information when you set up labware on the Bravo deck.

Pipette channel selection	Accessible Bravo deck locations
	
	
	

<table>
<tr><th>Pipette channel selection</th><th>Accessible Bravo deck locations</th></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

Pipette channel selection	Accessible Bravo deck locations
	
	
	

Before you add the task

Make sure:

- You have installed the correct pipette head. See "Pipette head requirements" on page 529 and the *Bravo Automated Liquid Handling Platform User Guide*.
- If you are going to use partial rows or columns of channels (barrels) on the pipette head, retract the tip box stripper pins on the pipette head. See the *Bravo Automated Liquid Handling Platform User Guide* for this procedure.

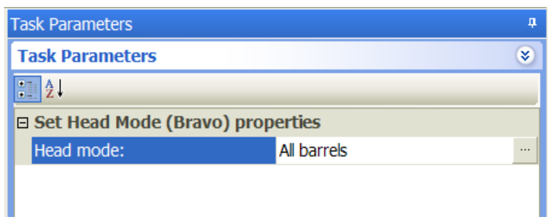
- The correct Bravo device file is open in the VWorks window.
- The profile you selected shows the correct head type, tip type, and miscellaneous settings.
- All the teachpoints have been added and verified.

Task parameters

IMPORTANT The Set Head Mode task should always precede the pipetting tasks that require the specified subset of pipettes.

IMPORTANT The Set Head Mode task should precede the Serial Dilution task. If you plan to change tips during the serial dilution process, add the Set Head Mode task before the Tips On task.

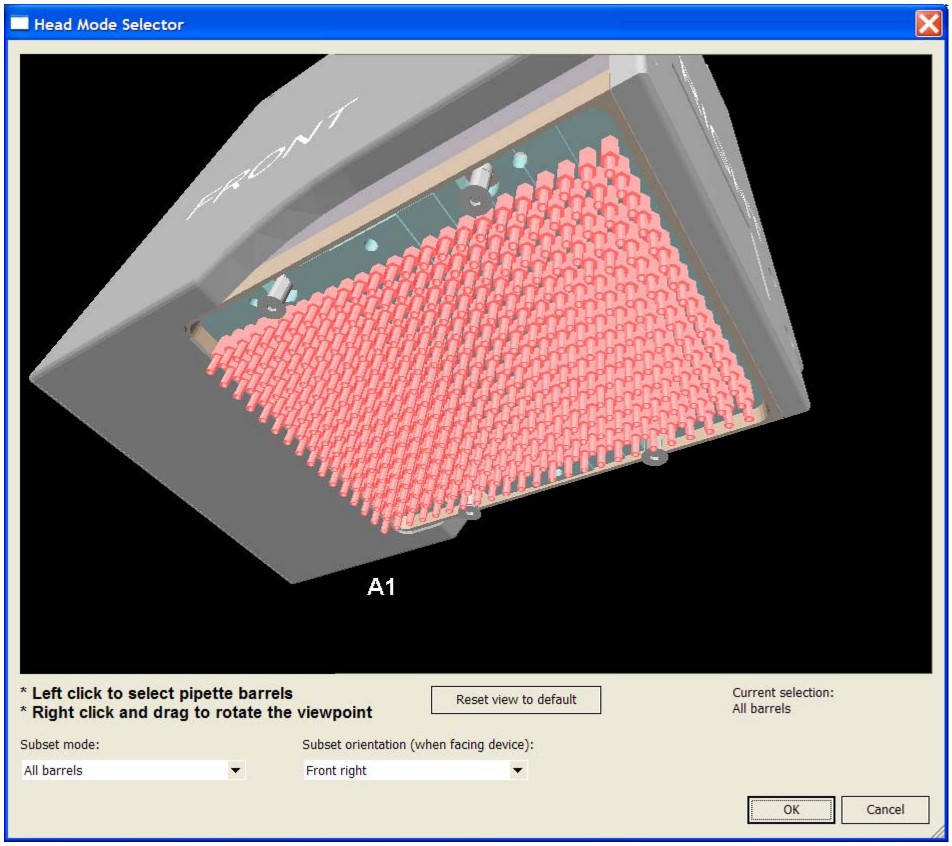
After adding the Set Head Mode task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Head mode	The channel selection. Click the Head mode field, and then click the ... button that appears. In the Head Mode Selector dialog box that opens, select the pipette channels.

Head Mode Selector dialog box

The Head Mode Selector dialog box allows you to select the pipette channels. Except for the **Subset: All barrels** mode, you select the desired channels using a combination of the **Subset mode** selection with the **Subset orientation** selection.



To select the pipette channels:

1 In the **Subset** list, select one of the following:

Subset mode	Description
All barrels	Uses all of the pipette channels.
Full column	Uses one or more full columns of pipette channels, starting from the right-most or left-most column.
Full row	Uses one or more full rows of pipette channels, starting from the first row or the last row.
Partial row/column	Uses part of the selected columns or rows.

2 In the **Subset orientation** list, select one of the following:

Subset orientation	Description
Front right	Uses pipette channels that contain the single channel in the front right corner.
Back right	Uses pipette channels that contain the single channel in the back right corner.
Front left	Uses one or more full rows of pipette channels, starting from the first row or the last row.
Back right	Uses part of the selected columns or rows.

3 When you are finished, click **OK** to save the selection.

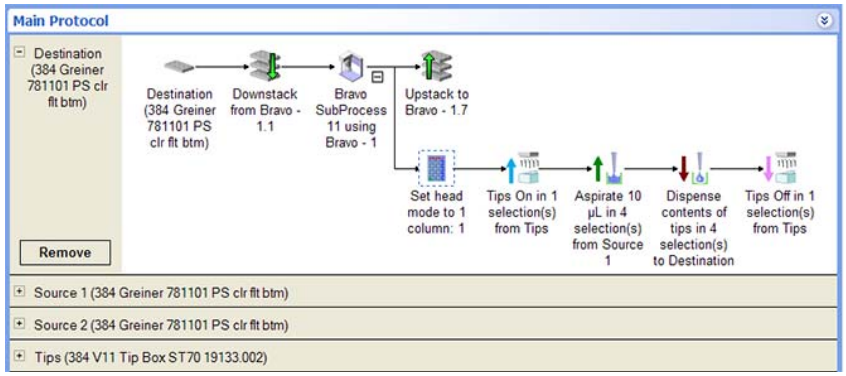
Example: Specify the pipette channels to use on the Bravo Platform

Goal

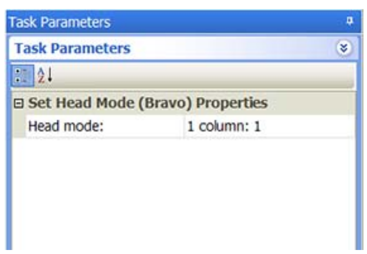
Use all of the channels of a Series III 96-channel pipette head for the liquid-handling tasks.

Implementation

At the beginning of the Bravo subprocess, add a Set Head Mode task as shown.



In the Set Head Mode Task Parameters area, All barrels is selected.



Related information

For information about...	See...
Configured labware	“Planning labware use” on page 20
Static labware	“Planning labware use” on page 20
Startup Protocol	“Setting up Startup and Cleanup Protocol processes” on page 60
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Tips On task	“Tips On (Bravo, Vertical Pipetting Station)” on page 542

For information about...	See...
Tips Off task	"Tips Off (Bravo, Vertical Pipetting Station)" on page 539
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Shake (Bravo, Vertical Pipetting Station)

Description

The Shake (Bravo) (Shake (Bravo)) and Shake (Vertical Pipetting Station) (Shake (Vertical Pipetting Station)) tasks instructs the Orbital Shaking Station to shake.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

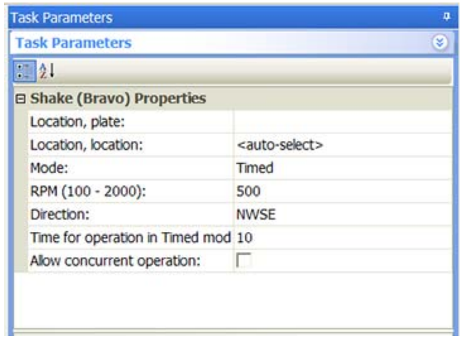
Requirements

To use the Shake task, you must first configure the Orbital Shaking Station in Bravo Diagnostics or Vertical Pipetting Station Diagnostics.

Task parameters

Note: The task parameters for Shake (Bravo) and Shake (Vertical Pipetting Station) are identical.

After adding the Shake task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Shake task.
Location, location	The location at which the Shake task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Mode	The action of the task: • *On*. Turns on the Orbital Shaking Station. • *Off*. Turns off the Orbital Shaking Station. • *Timed*. Turns on the shaking timer. You must specify the length of time to shake. If you plan to time the shaking, add only one Shake task (the task turns on the shaking, and at the end of the time period, the shaking turns off automatically). If you are not timing the shaking, add two Shake tasks in the protocol (one to turn on the shaking and the other to turn off the shaking).
RPM	The shake speed, in revolutions per minute.
Direction	The direction to shake. Select one of the direction combinations: NWSE, NESW, NS, EW, NW/SE, NE/SW.
Time for operation in Timed mode	The length of time, in seconds, you want to leave the shaking on. At the end of the period, the shaking will turn off.

Parameter	Description
Allow concurrent operation	The option to permit the accessory to operate simultaneously with other tasks. For example, the Shake and Mix tasks can operate simultaneously. **CAUTION** To shake and mix concurrently, use only 96-well disposable-tip pipette heads in 96-well microplates.

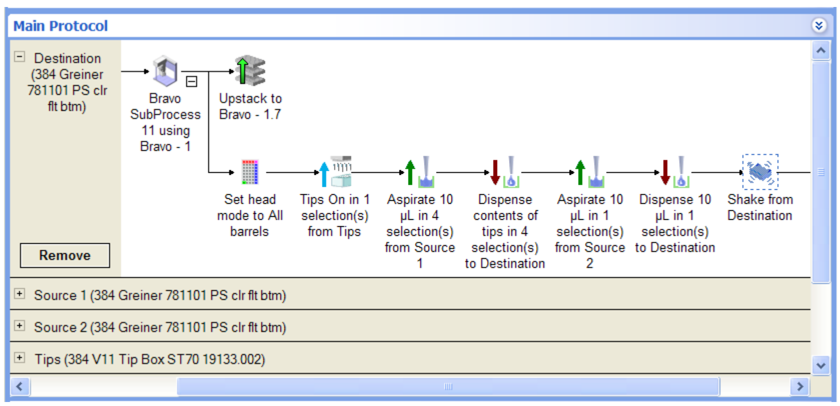
Example: Shake the destination microplate after adding reagents

Goal

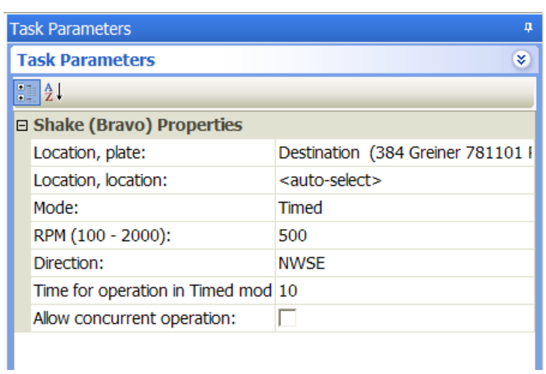
Shake the destination microplate after adding reagents from two source microplates.

Implementation

In the liquid-handling subprocess, a Shake task is added after the second reagent is added. In the following example, a Shake task is added in a Bravo subprocess.



In the Shake Task Parameters area, Destination is selected, because the goal is to shake the destination microplate.



Related information

For information about...	See...
Configured labware	"Planning labware use" on page 20
Static labware	"Planning labware use" on page 20
Startup Protocol	"Setting up Startup and Cleanup Protocol processes" on page 60
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Tips Off (Bravo, Vertical Pipetting Station)

Description

The Tips Off (Bravo) (Tips Off (Bravo)) and Tips Off (Vertical Pipetting Station) tasks (Tips Off (Vertical Pipetting Station)) remove disposable pipette tips from the pipette heads that are compatible with disposable tips.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Note: Options in the Tips On and Tips Off tasks permit tracking of pipette tip usage during and across protocol runs. In addition, an option in the Tips Off task allows you to reuse pipette tips for a portion of the protocol run.

Requirements

Tracking tips in serial dilution tasks

In addition to selecting the tracking option in the Tips On and Tips Off tasks, you must also turn on the tracking in the Serial Dilution Wizard. For more detailed information, see "Serial Dilution (Bravo, Vertical Pipetting Station)" on page 524.

Tracking tips across different protocols

If you want to track pipette tip usage across different protocols that contain Bravo and Vertical Pipetting Station Subprocesses, make sure:

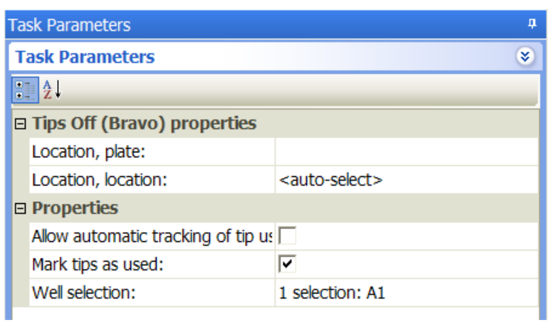
- The protocols reference the same device file.
- The tip boxes are at the same physical locations.
- In the software, the tip boxes are configured at the same locations across the protocols.

Note: When a set of tipboxes are designated as a process plate, tip usage is tracked during the protocol run. At the end of a run, the software resets the tipbox to the original state. The software assumes that in each subsequent run, you will load tipboxes in the original state before the run.

Task parameters

Note: The task parameters for Tips Off (Bravo) and Tips Off (Vertical Pipetting Station) are similar.

After adding the Tips Off task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Tips Off task.
Location, location	The location at which the Tips Off task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.

Parameter	Description
Allow automatic tracking of tip usage	The option to allow the software to track pipette tip usage during the protocol run or across different protocol runs. When you start the run, the software will determine the positions to use in the tipbox. If you select the option in this task, you must also select the option in the Tips On task. If you do not select this option, you must specify the positions to use in the tip box using the Well selection parameter.
Mark tips as used	The option to use only new pipette tips during the protocol run. Select the option so that the software counts the number of tips used during the protocol run. The tips that have been used once are marked as used so that they cannot be picked up and reused. Clear the check box so that during the next Tips On task, the same tips can be reused.
Well selection	The well positions to use for the Tips Off task. This parameter is available only for manual tracking of pipette tips.

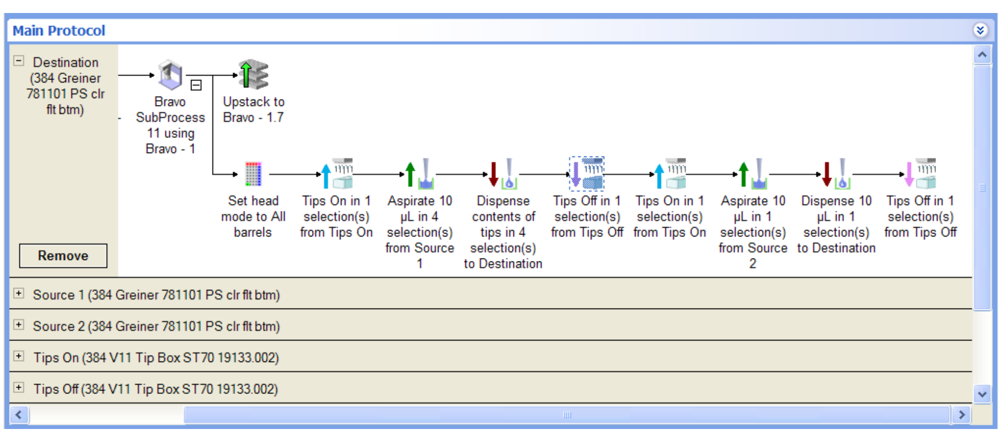
Example: Change tips during liquid-handling tasks on the Bravo Platform

Goal

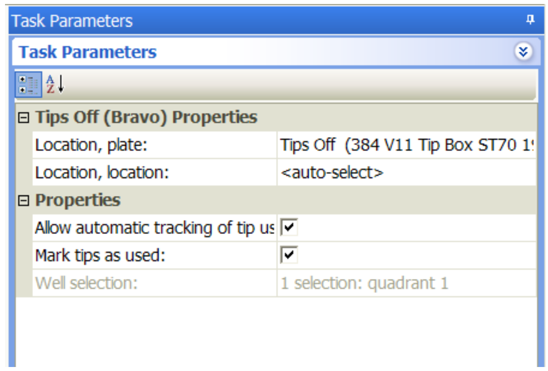
Add reagents from two source microplates into a destination microplate. Change tips between source microplate 1 and source microplate 2 to prevent contaminating the source microplates.

Implementation

In the liquid-handling subprocess, add a Tips Off task after each Dispense task as shown.



Because the tips are removed at a Tips Off tipbox, the Tips Off tipbox is selected in the Tips Off Task Parameters area.



Related information

For information about...	See...
Tips On task	"Tips On (Bravo, Vertical Pipetting Station)" on page 542
Serial Dilution task	"Serial Dilution (Bravo, Vertical Pipetting Station)" on page 524
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Tips On (Bravo, Vertical Pipetting Station)

Description

The Tips On (Bravo) (Tips On (Bravo)) and Tips On (Vertical Pipetting Station) tasks (Tips On (Vertical Pipetting Station)) presses disposable pipette tips on the pipette head. The task is not for use with fixed-tip pipette heads.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Note: Options in Tips On and Tips Off tasks permit the tracking of pipette tip usage during and across protocol runs. In addition, an option in the Tips Off task allows you to reuse pipette tips for a portion of the protocol run.

Requirements

Tracking tips in serial dilution tasks

In addition to selecting the tracking option in the Tips On and Tips Off tasks, you must also turn on the tracking in the Serial Dilution Wizard. For more detailed information, see “Serial Dilution (Bravo, Vertical Pipetting Station)” on page 524.

Tracking tips across different protocols

If you want to track pipette tip usage across different protocols that contain Bravo and Vertical Pipetting Station Subprocesses, make sure:

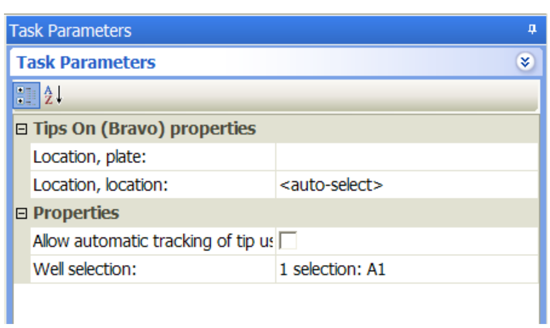
- The protocols reference the same device file.
- The tip boxes are at the same physical deck locations.
- In the software, the tip boxes are configured at the same locations across the protocols.

Note: When a set of tip boxes are designated as a process plate, tip usage is tracked during the protocol run. At the end of a run, the software resets the tip box to the original state. The software assumes that in each subsequent run, you will load tip boxes in the original state before the run.

Task parameters

Note: The task parameters for Tips On (Bravo) and Tips On (Vertical Pipetting Station) are similar.

After adding the Tips On task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Tips On task.
Location, location	The location at which the Tips On task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Allow automatic tracking of tip usage	The option to allow the software to track pipette tip usage during the protocol run or across different protocol runs. When you start the run, the software will determine the positions to use in the tipbox. If you select the option in this task, you must also select the option in the Tips Off task. If you do not select this option, you must specify the positions to use in the tip box using the Well selection parameter.
Well selection	The well positions to use for the Tips On task. This parameter is available only for manual tracking of pipette tips.

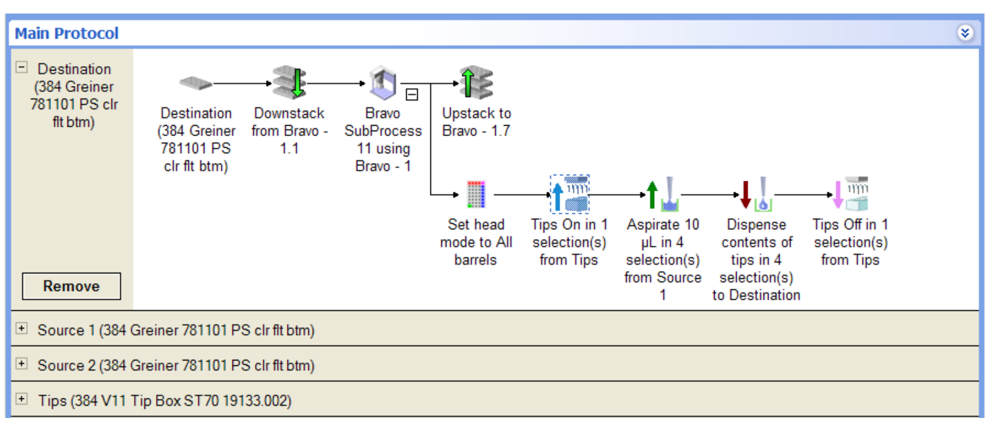
Example: Press tips on before liquid-handling tasks on the Bravo Platform

Goal

Make sure tips are pressed on before liquid-handling tasks start.

Implementation

In the liquid-handling subprocess, a Tips On task is added at the beginning. In the following example, because the protocol is run on a Bravo Platform, a Set Head Mode task must be added before the Tips On task.



Related information

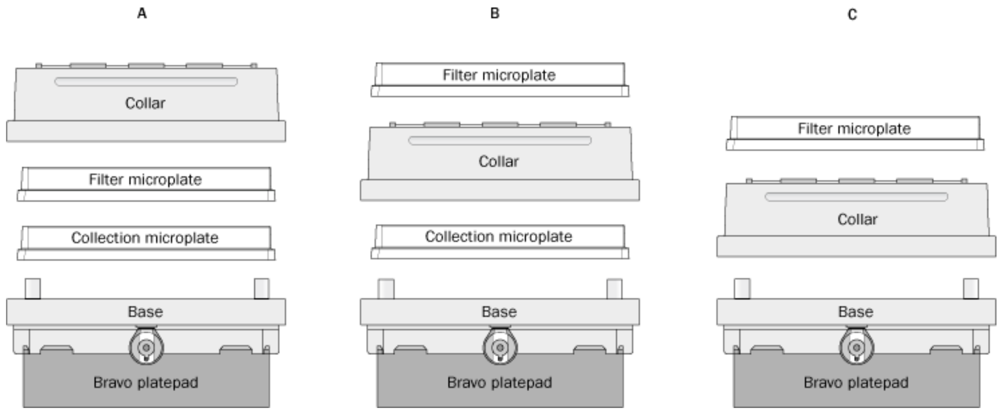
For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Tips On task	"Tips On (Bravo, Vertical Pipetting Station)" on page 542
Serial Dilution task	"Serial Dilution (Bravo, Vertical Pipetting Station)" on page 524
Set Head Mode task	"Set Head Mode (Bravo)" on page 529
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Toggle Vacuum (Bravo, Vertical Pipetting Station)

Description

The Toggle Vacuum (Bravo) (Toggle Vacuum (Bravo)) and Toggle Vacuum (Vertical Pipetting Station) (Toggle Vacuum (Vertical Pipetting Station)) tasks turn on and turn off the vacuum.

In a Bravo Subprocess, use the Toggle Vacuum (Bravo) task if the Vacuum Filtration Station has configuration A as shown in the following diagram. In configuration A, the filter microplate is part of the station assembly.



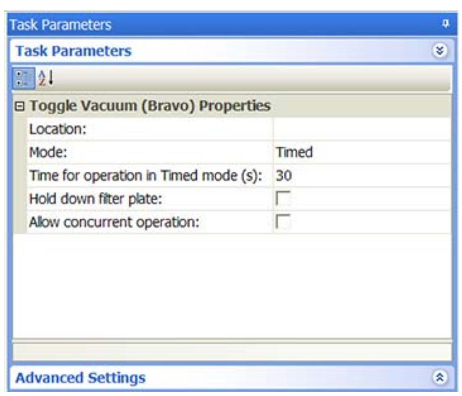
If the Vacuum Filtration Station has configuration B and you want the robot to move the filter microplate from another location to the station, use the Move and Filter Plate task to turn on and turn off the vacuum. See "Move and Filter Plate (Bravo)" on page 505.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Task parameters

Note: The task parameters for Toggle Vacuum (Bravo) and Toggle Vacuum (Vertical Pipetting Station) are identical.

After adding the Toggle Vacuum task at the desired point in the subprocess, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location	The location of the Vacuum Filtration Station.

Parameter	Description
Mode	The action of the task: • *On.* Turns on the vacuum. • *Off.* Turns off the vacuum. • *Timed.* Turns on the vacuum timer. You must specify the vacuum length of time. If you plan to time the filtering process, add only one Toggle Vacuum task (the task turns on the vacuum, and at the end of the time period, the vacuum turns off automatically). If you are not timing the filtering process, add two Toggle Vacuum tasks in the protocol (one to turn on the vacuum and the other to turn off the vacuum). *Note:* If the filtering process is not timed, the protocol can perform other tasks in parallel.
When filtration timing begins	The different options for when to start timing the filtration process: • When pressure is achieved • When pump starts This parameter is for the Vario vacuum pump only.
Time for operation in Timed mode	The length of time, in seconds, you want to leave the vacuum on. At the end of the period, the vacuum will turn off.
Hold down filter plate	The option to have the robot hold down the filter microplate when the vacuum is turned on to ensure a secure vacuum seal.
Time allowed to reach pressure	The length of time, in seconds, during which the pump is allowed to reach the target pressure. The software will display an error message if the target pressure is not reached at the end of the specified time. This parameter is for the Vario vacuum pump only.
Target pressure	The desired vacuum pressure, in mbar, Torr, or hPa. The pressure unit is set on the Vario vacuum pump. This parameter is for the Vario vacuum pump only.
Allow concurrent operation	The option to permit the accessory to operate simultaneously with other tasks.

Example

Goal

Assemble a Vacuum Filtration Station whose configuration is base-collection plate-filter plate-collar. Filter for 30 seconds. Disassemble the station.

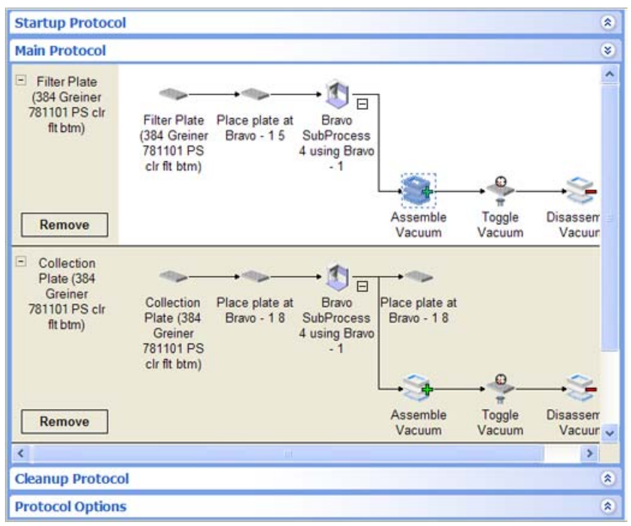
Implementation

In Bravo Diagnostics, configure the Bravo deck such that:

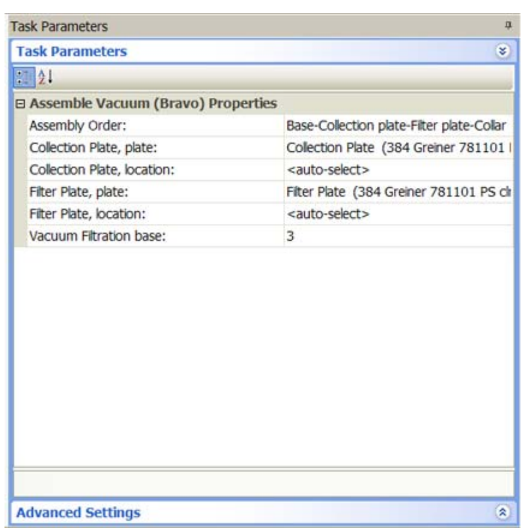
- The Vacuum Filtration Station will be assembled at deck location 3.
- The Vacuum Filtration Station base is at deck location 3.
- The collar is at deck location 9.

In the protocol, add the following as shown:

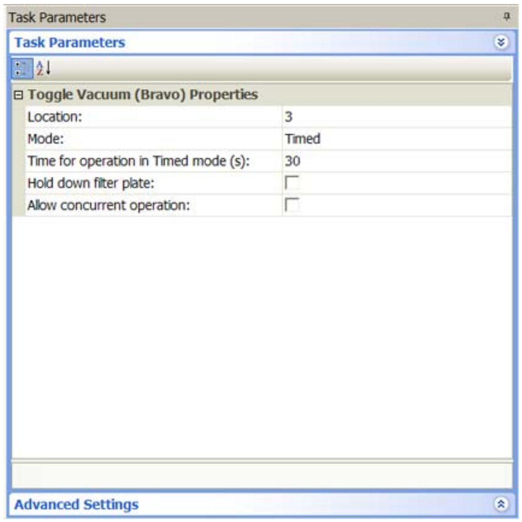
- Filter Plate process
- Collection Plate configured labware (at deck location 8)
- Place Plate (to place the Filter Plate at location 5), Assemble Vacuum, Toggle Vacuum, and Disassemble Vacuum tasks



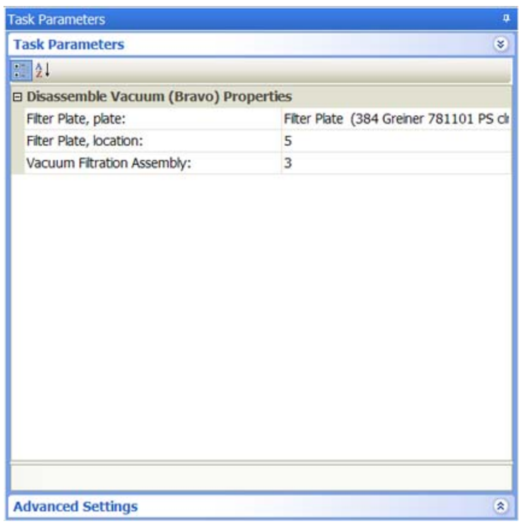
Set the Assemble Vacuum task parameters as shown:



Set the Toggle Vacuum task parameters as shown:



Set the Disassemble Vacuum task parameters as shown:



The resulting protocol will run as follows:

1 Acknowledge that the filter microplate (Filter Plate) is starting at deck location 5.

2 Acknowledge that the collection microplate (Collection Plate) is starting at deck location 8.

3 Assemble the Vacuum Filtration Station (base-collection plate-filter plate-collar):

   a Move the Collection Plate from deck location 8 and place it on top of the base at deck location 3.

   b Move the Filter Plate from deck location 5 and place it on top of the Collection Plate at deck location 3.

   c Move the collar from deck location 9 and place it on top of the Filter Plate at deck location 3.

4 Turn on the vacuum for 30 seconds, and then turn off the vacuum.

5 Disassemble the Vacuum Filtration Station:

- **a** Move the collar back to deck location 9.
- **b** Move the Filter Plate back to deck location 5.
- **c** Move the Collection Plate back to deck location 8.

Related information

For information about…	See…
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Assemble Vacuum task	"Assemble Vacuum (Bravo)" on page 439
Disassemble Vacuum task	"Disassemble Vacuum (Bravo)" on page 449
Move and Filter Plate task	"Move and Filter Plate (Bravo)" on page 505
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Wash Tips (Bravo, Vertical Pipetting Station)

Description

The Wash Tips (Bravo) ([Wash Tips (Bravo)]) and Wash Tips (Vertical Pipetting Station) ([Wash Tips (Vertical Pipetting Station)]) tasks wash pipette tips using a number of aspirate and dispense actions.

Task is available for…	Task is available in…
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Requirements

In Bravo Diagnostics or Vertical Pipetting Station Diagnostics, as applicable, ensure that at least one of the following is set up at the correct deck location or shelf and associated with a Pump Module.

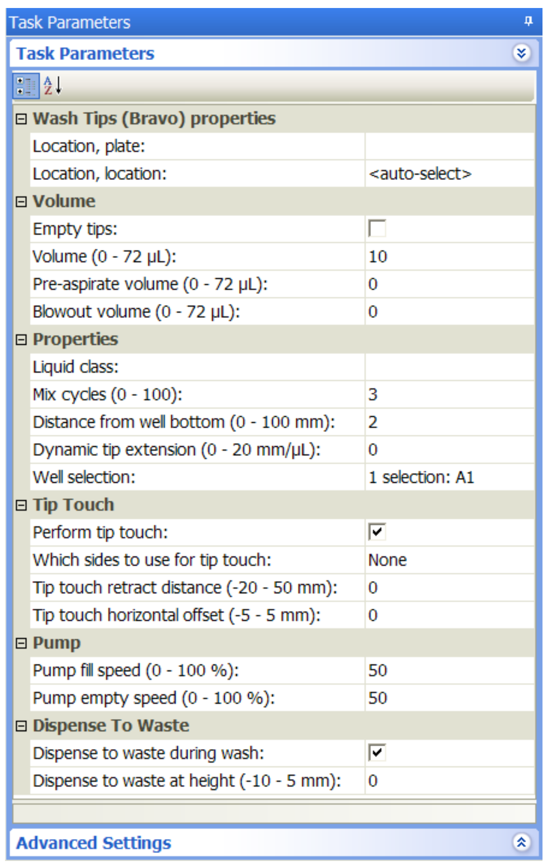
- Autofilling Reservoir
- Tip Wash Station (also known as MicroWash Reservoir)
- Open Wash Tray

Ensure that the protocol has a configured labware for the selected autofilling reservoir in addition to the other processes and configured labware required by the protocol.

Task parameters

Note: The task parameters for Wash Tips (Bravo) and Wash Tips (Vertical Pipetting Station) are identical.

After adding the Wash Tips task at the desired point in the subprocess, set the following parameters in the **Task Parameters** area:



Parameter	Description
Location, plate	The labware involved in the Wash Tips task.

Parameter	Description
Location, location	The location at which the Wash Tips task occurs. <auto-select> automatically places the labware at the first-available or appropriate location for the task. If accessories are installed on the deck or shelf, the software uses the accessory configuration information in Bravo Diagnostics or Vertical Pipetting Station Diagnostics to determine the correct location for the task.
Empty tips	The option to empty the entire contents of the pipette tips, including fluid and air. The Volume parameter is ignored if this option is selected. *Bravo 96AM Head only*. Empty tips causes any liquid in the syringes and cartridges or pipette tips to be automatically dispensed to waste at the beginning of the Wash Tips task. In addition, if the Perform tip touch check box is selected, the syringe probes, cartridge tips, or pipette tips will perform a tip touch on the sides of the wash station chimneys after dispensing to waste.
Volume (µL)	The volume of liquid to be dispensed from each pipette tip.
Pre-aspirate volume (µL)	The volume of air to be drawn before the pipette tips enter the liquid.
Blowout volume (µL)	Specifies the volume of air to dispense after the main volume has been dispensed. Typically, the blowout volume is the same as the pre-aspirate volume. *Note:* Blowout only occurs in the last quadrant dispensed for a given dispense action.
Liquid class	The liquid class associated with this liquid. **IMPORTANT** To ensure consistent pipetting, always select a liquid class for liquid-handling tasks.
Mix cycles	The number of times you want to aspirate and dispense. Each cycle consists of one aspirate action and one dispense action.

Parameter	Description
Distance from well bottom (mm)	The distance between the end of the pipette tips and the well bottoms during the Wash Tips task. **IMPORTANT** The labware definition must be accurate and the teachpoint must be precise in order for the system to position the tips at the correct distance from the well bottom.
Dynamic tip extension (mm/(µL)	The rate at which the pipette head moves during the Wash Tips task. The software calculates the distance over which the tips will move without crashing. Use dynamic tip extension to prevent spills as the pipette tips displace the liquid. To move the tips: • *At the same rate as the volume change.* Calculate dynamic tip extension (DTE) as follows: DTE = (well depth)/(well vol) = 1/A, where A is the cross-sectional area of a well with straight walls • *Faster than the volume change.* DTE > 1/A • *Slower than the volume change.* DTE < 1/A The starting and ending positions can be calculated as follows: $(V_{dispensed} * DTE) + D_{well\ bottom}$ $(V_{aspirated} * DTE) + D_{well\ bottom}$
Well selection	The wells at which the Wash Tips task occurs. Click in the parameter box, and then click the Browse button to select the wells in the Well Selection dialog box. Use this parameter only if the pipette head has fewer tips than the number of wells in the microplate, or if you are in single-row or single-column mode.
Perform tip touch	The option to touch the pipette tip on one or more sides of the well.
Which side to perform tip touch	The wall or walls for tip touch: North, South, East, West, North/South, West/East, West/East/South/North. If you also select the Dispense to waste during wash option, the tip touch is performed on the northeast side only.

Parameter	Description
Tip touch retract distance	The vertical distance the pipette tips rise before touching the sides of the wells.
Tip touch horizontal offset	The horizontal distance the tips move. The value is based on the well diameter specified by the labware definition. The value of the parameter determines the direction of movement: • *0*. Tips move a horizontal distance equal to the well radius. • *Great than 0*. Tips attempt to move past the well radius, which results in a more forceful tip touch. • *Less than 0*. Tips move a distance less than the radius of the well, resulting in a lighter tip touch.
Pump fill speed (%)	The speed, in percent of maximum speed, of liquid flow into the reservoir. For the MicroWash Reservoir, this value should be high enough for the washing liquid to just bubble over the tops of the chimneys.
Pump empty speed (%)	The speed, in percent of maximum speed, of liquid flow out of the the reservoir. For the MicroWash Reservoir, this value should be slightly higher than that of the inflow pump to prevent an overflow.
Dispense to waste during wash	The option to move the tips by a specified offset (defined in the Labware Editor) and dispense used fluid outside of the reservoir chimney. This option applies only to reservoirs that have chimneys.
Dispense to waste at height (mm)	The height at which the dispense action occurs. For example, during the dispense action, the tips move up to clear the chimneys, move the offset distance, and then lower to the distance you specified. If you want the lower the tips by 10 mm, specify –10 mm.

Example: Wash the pipette tips after the liquid-handling tasks on the Bravo Platform

Goal

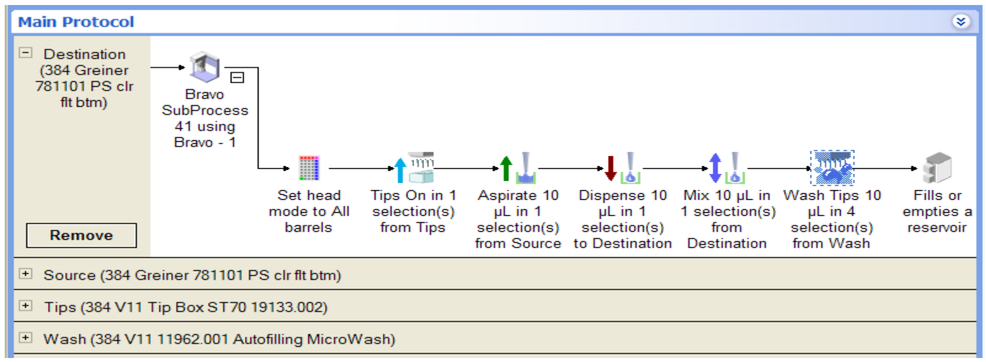
After some liquid-handling tasks, wash the pipette tips in preparation for reuse.

Implementation

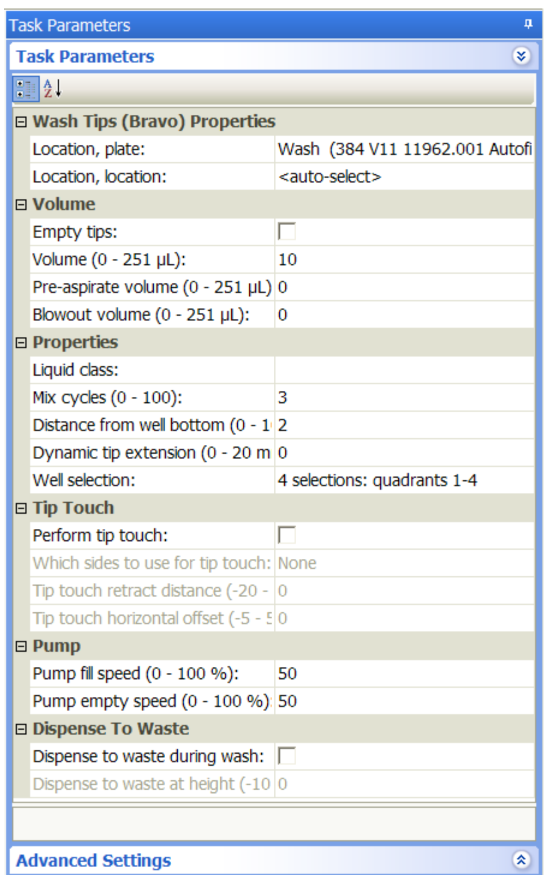
In Bravo Diagnostics, set up the Tip Wash Station on a deck location. Associate the reservoir with one or more pump modules.

In the protocol, add a configured labware for the Tip Wash Station in addition to the other processes and configured labware required by the protocol. In the example shown, the configured reservoir is called Wash.

In the Bravo subprocess where the liquid-handling tasks are specified, a Wash Tips task is added at the end of a cycle of liquid-handling tasks.



In the Wash Tips Task Parameters area, Wash (the name for the Tip Wash Station) is selected so that the task is performed in this labware.



Related information

For information about...	See...
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Aspirate task	“Aspirate (Bravo, Vertical Pipetting Station)” on page 392
Dispense task	“Dispense (Bravo, Vertical Pipetting Station)” on page 451
Set Head Mode task	“Set Head Mode (Bravo)” on page 529
Tips On task	“Tips On (Bravo, Vertical Pipetting Station)” on page 542
Tips Off task	“Tips Off (Bravo, Vertical Pipetting Station)” on page 539
Pump Reagent task	“Pump Reagent (Bravo, Vertical Pipetting Station)” on page 521
Microplate-handling tasks	“Setting parameters for microplate-handling tasks” on page 277
Microplate-storage tasks	“Setting parameters for microplate storage tasks” on page 349
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

11
Setting parameters for scheduling tasks

This chapter contains the following topics:

Connect to Encore

Description

The Connect to Encore task (Connect to Encore) establishes communication with the Encore Multispan software.

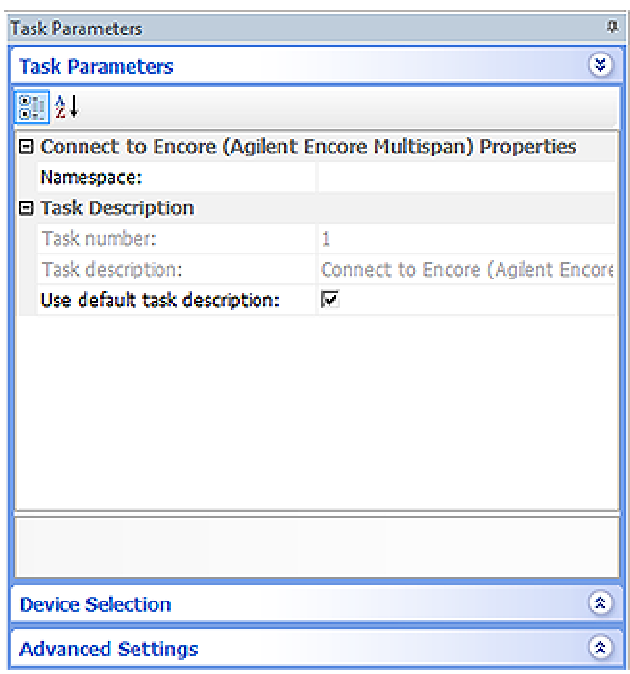
You use the Connect to Encore task if you are running an Encore Multispan software and you want to process labware at VWorks-controlled devices. For information about integrating VWorks-controlled devices with an Encore Multispan System, see the *Encore Multispan System knowledge base*.

Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Note: Although you can add the Connect to Encore task at any point in a protocol, it is typically added in the Startup Protocol.

Task parameters

After adding the Connect to Encore task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Namespace	The unique name that identifies the VWorks protocol. The Namespace parameter allows the Encore Multispan software to start the correct VWorks process if you are running more than one VWorks protocol, and processes in different protocols have the same name. If you are running only one VWorks protocol, the Namespace parameter is not used. The Namespace value is referenced by the VWorks task in the Encore Multispan protocol. For more information, see the *Encore Multispan System knowledge base*.
Task number	The number that indicates the position of the task in the protocol.
Task description	The description of the task.
Use default task description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

Related information

For information about...	See...
Running VWorks-controlled devices with the Encore Multispan System	*Encore Multispan System knowledge base*
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
User Message task	"User Message" on page 599
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387

Define Plate Set

Description

The Define Plate Set task (Define Plate Set) allows you to create an array variable to represent a group of process plates that will be processed using the same tasks in a looping routine. You can reference the plate set using the Location, plate parameter in a task.

Task is available for...	Task is available in...
Any device	Main Protocol, not in a subprocess

Requirements

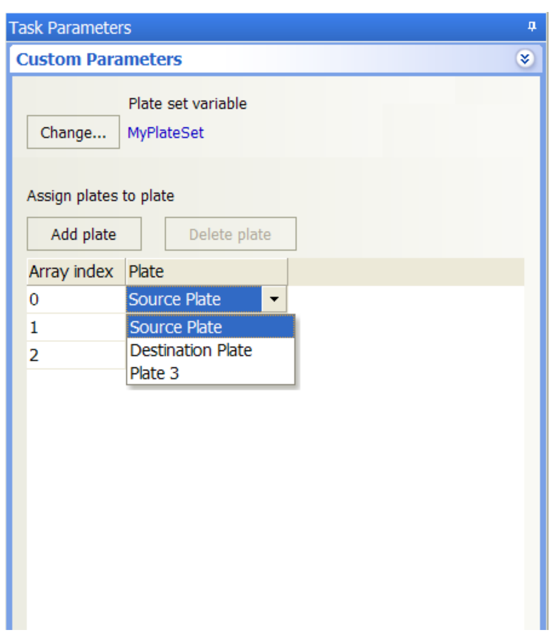
You must use the Define Plate Set task with the Loop and Loop End tasks, and it must be added in the protocol as follows:

- Before the Loop and Loop End tasks that uses it.
- In the top-level process, not in the subprocess that contains the Loop and Loop End tasks.
- At the beginning of the process, and the process is not spawned.

The Loop tasks allow you to specify the starting array index value, how frequently the variable will increment or decrement, and the amount by which the index will increment or decrement.

Custom parameters

After adding the Define Plate Set task at the desired point in the protocol, define the plate set in the **Custom Parameters** area:



Parameter	Description
Plate set variable	The name of the variable assigned to the plate set. The name must conform to JavaScript rules for variable names: start with an alphabetical character or the underscore symbol, followed by combinations of alphabetical characters, numbers, and underscore symbols. To define a new name, click **Change,** type the new name, and then click **OK.**
Add plate	The command that allows you to add a process plate to the plate set. To add a process plate, click **Add plate.** A new row appears in the Array table and an index value is automatically assigned. In the **Plate** field, select a process plate from the list.
Delete plate	The command that deletes the selected process plate.
Array table	The list of process plates in the plate set. The table consist of: • *Array index*. The value that identifies the process plate in the array. • *Plate*. The process plate name.

Referencing a plate set in a task

To reference a plate set in a task:

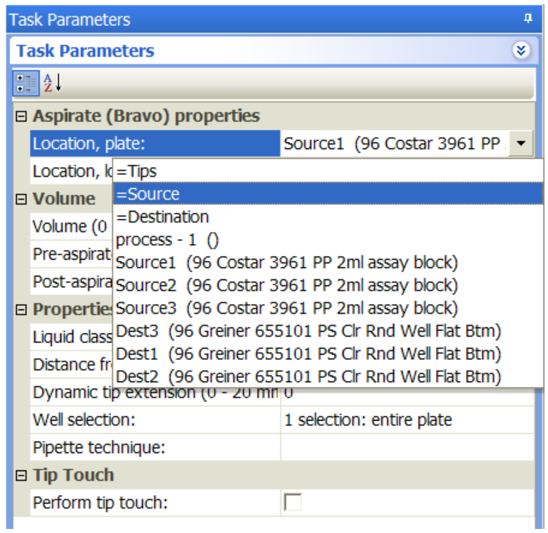
1 Select the task in the protocol.

2 In the **Task Parameters** area, select the plate-set variable from the **Location, plate** list.

If the variable does not appear in the list, make sure:

- The task is inside of a loop.
- The Loop task specifies the plate set name in its Custom Parameters area.

The following example shows the Aspirate task parameters. Three plate-set variables were defined earlier in the protocol and are available for selection: Tips, Source, and Destination. The example shows the Source variable selected.



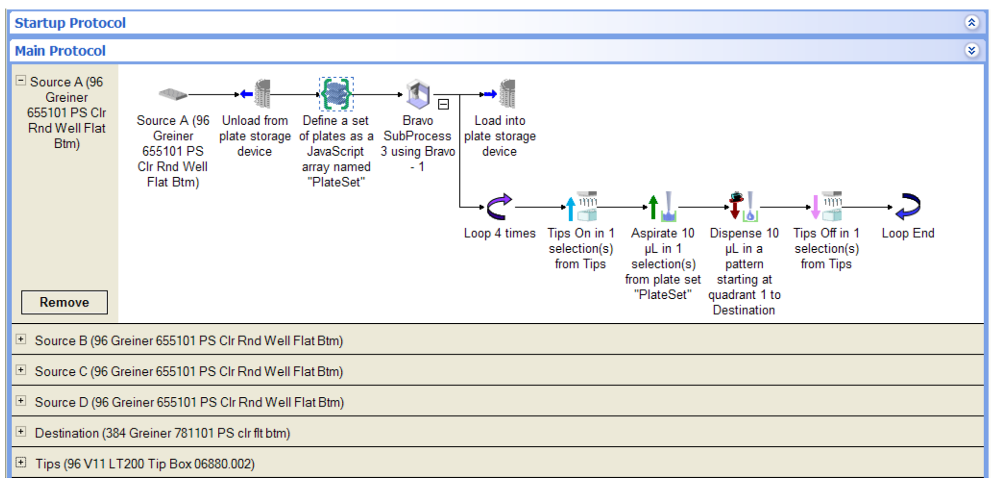
Example: Define a plate set for processing on the Bravo Platform

Goal

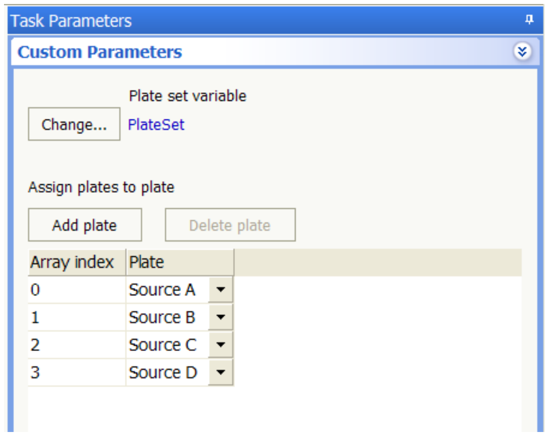
Define a plate set of source microplates (Source A, B, C, and D). Aspirate from each source microplate into a destination microplate.

Implementation

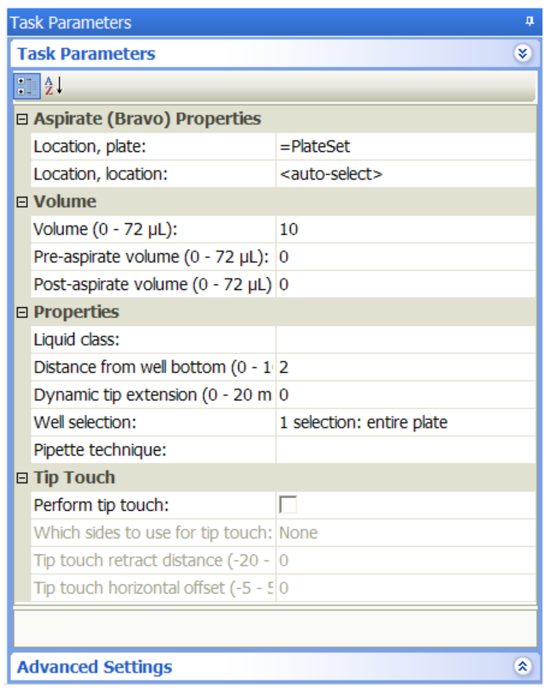
Add a Define Plate Set task at the beginning of the protocol.



The plate set is called PlateSet.



Instead of adding four separate Aspirate tasks for each source microplate, add only one Aspirate task in the liquid-handling subprocess. In the Aspirate Task Parameters area, PlateSet is selected.



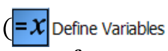
Related information

For information about...	See...
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Loop and Loop End tasks	“Loop and Loop End” on page 574
Spawn Process task	“Spawn Process” on page 595
Process Control task	“Process Control” on page 586
Task parameter variables	“Using simple variables” on page 77

For information about...	See...
Microplate-handling tasks	“Setting parameters for scheduling tasks” on page 557
Microplate-storage tasks	“Setting parameters for scheduling tasks” on page 557
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Define Variables

Description

The Define Variables task (

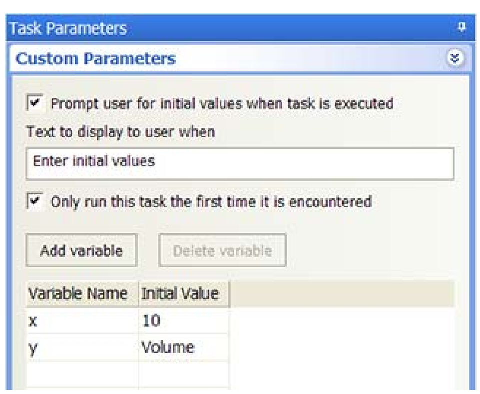
) allows you to create variables and assign initial values. You can reference the variables in a task.

You can use the Define Variables task with the Loop and Loop End tasks. The Loop tasks allow you to specify the starting variable value, how frequently the value will increment or decrement, and the amount by which the value will increment or decrement.

Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Custom parameters

After adding the Define Variables task at the desired point in the protocol, define the variables in the **Custom Parameters** area:



Parameter	Description
Prompt operator for initial values when task is run	The option to display all the variables in the Define Variables task and ask the operator to specify initial values for the variables. *Note:* This option is equivalent to the **User data entry into variable** option in the User Message task.
Text to display to operator when prompted	The instructions or description that displays in the prompt dialog box.
Only run this task the first time it is encountered	The option to run this task the first time it appears in the protocol run. Select this option if the protocol is run multiple times or the task is in a loop, and you only want to run it the first time.
Add variable	The command that adds a new variable in the Variable table. To add a new variable, click **Add variable.** Double-click the **Variable Name** field and type the variable name. Double-click the **Initial Value** field and type the starting value. *Note:* The starting value specified in this task overrides the starting value the operator provides during run time.
Delete variable	The command that deletes the selected variable.
Variable table	The list of variables. The table consists of: • *Variable Name*. The name of the variable. • *Initial Value*. The starting value of the variable.

Referencing variables in a task

To reference variables in a task:

1 Select the task.
2 In the **Task Parameters** area, type the =x in the parameter field, where x is the variable name. For more information about using variables, see “Using simple variables” on page 77.

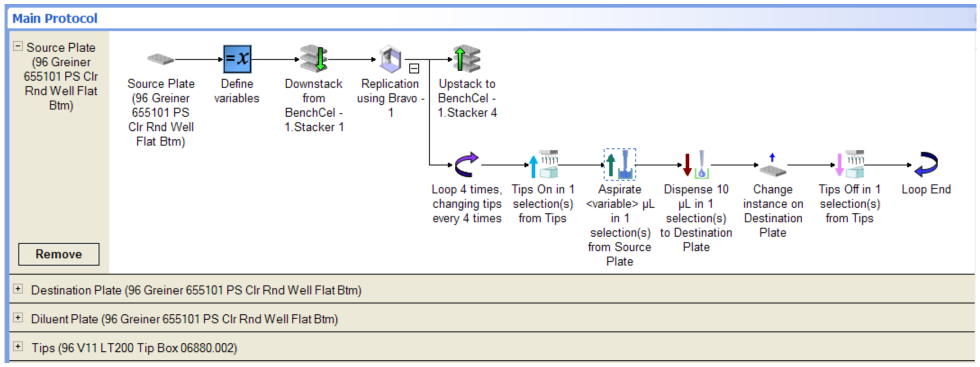
Example: Define a plate set for processing on the Bravo Platform

Goal

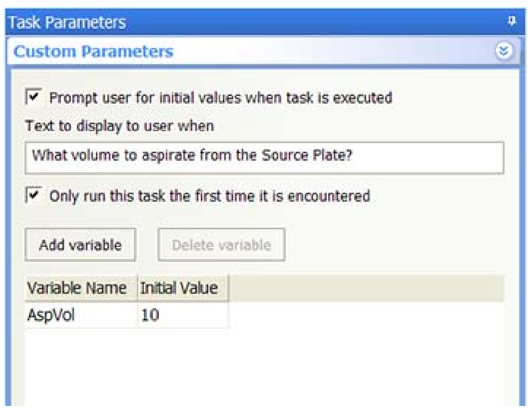
Define a variable to represent the aspirate volume. Request the operator to provide the initial value of the aspirate volume.

Implementation

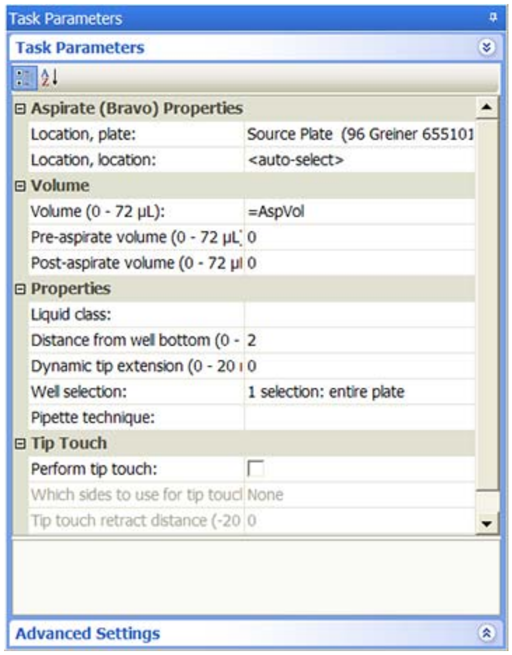
Add a Define Plate Set task at the beginning of the protocol.



The AspVol variable is defined and it represents the aspirate volume. Notice the options are selected to request input from the operator at the start of the protocol run. The operator's input will override the initial value specified in the Custom Parameters area.



In the Aspirate Task Parameters area, the AspVol variable is selected for the aspirate volume.



Related information

For information about...	See...
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Loop and Loop End tasks	“Loop and Loop End” on page 574
Spawn Process task	“Spawn Process” on page 595
Process Control task	“Process Control” on page 586
Task parameter variables	“Using simple variables” on page 77
Microplate-handling tasks	“Setting parameters for scheduling tasks” on page 557
Microplate-storage tasks	“Setting parameters for scheduling tasks” on page 557
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Change Instance

Description

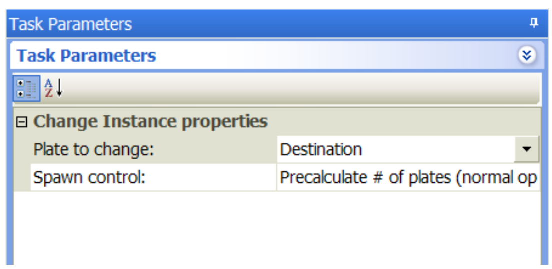
The Change Instance task (Change Instance) allows you to change a plate instance within a loop. For example, you can use the Change Instance task within the Loop and Loop End tasks to aspirate from one source microplate and dispense into different instances of a destination microplate.

Task is available for...	Task is available in...
Any device	Main Protocol

Do not include the Change Instance task in a subprocess of a spawned process that is running as a subroutine. For example, if process 1 spawns process 2 as a subroutine (Spawn as subroutine option) and process 2 contains a subprocess, the Change Instance task will not work in the subprocess.

Task parameters

After adding the Change Instance task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Plate to change	The process plate whose instance you want to change.

Parameter	Description
Spawn control	The spawn options: • *Precalculate # of plates (normal operation)*. Select this option if the number of microplates can be calculated before the run starts. For example, the calculation can be based on the Number of times to loop value in the Loop task. Do not select this option if the Number of times to loop is a variable, or the looping and the Change Instance task is scripted. • *Spawn new plate when task runs*. Select this option if the Number of times to loop is a variable, or you are scripting the looping and the Change Instance task. The software will not precalculate the number of microplates to use in the Change Instance task. Instead, every time this task runs, a new microplate is brought into the system. • *Do not spawn new plates*. Select this option to script the number of microplates in the run without having to wait for the Change Instance task to execute to bring in new microplates.

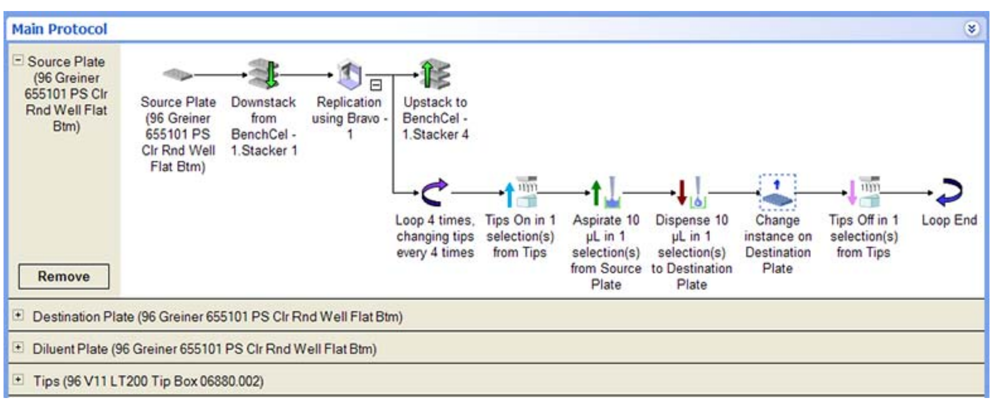
Example: Dispense into multiple destination microplates

Goal

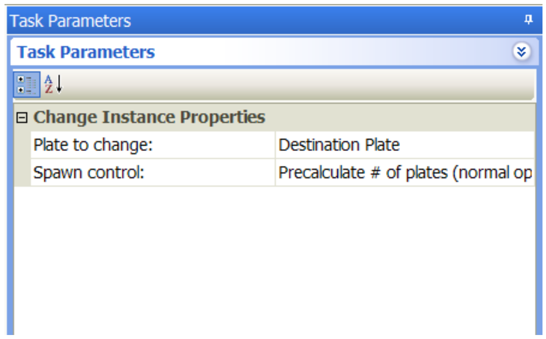
Aspirate from one source microplate and dispense into multiple destination microplates.

Implementation

Within the liquid-handling subprocess loop, add a Change Instance task after the Dispense task.



In the Change Instance Task Parameters area, the Destination Plate is selected.



Related information

For information about...	See...
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Loop and Loop End tasks	“Loop and Loop End” on page 574
Microplate-handling tasks	“Setting parameters for scheduling tasks” on page 557
Microplate-storage tasks	“Setting parameters for scheduling tasks” on page 557
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Group Begin and Group End

Description

The Group Begin ({ Group Begin) and Group End (} Group End) tasks are used together to:

- Group a number of tasks within a subprocess.
- Constrain the grouped tasks to run in the specified sequence.
- Prevent downstream tasks from starting until the grouped tasks are finished.

Note: If the tasks are not grouped, the software will determine the fastest task sequence based on available resources.

During the protocol run, the software will make sure the labware needed by the grouped tasks are on the device locations before running the first task in the group. In addition, while the grouped tasks are running, only the microplates required by the grouped tasks will be allowed to be moved onto, off of, or within the device.

Use the Group Begin and Group End tasks if you want to control the sequence of liquid-handling tasks to make sure each microplate is handled the same way in the same sequence.

Note: To further control the tasks, especially those with time limits, you can specify time constraints between dependent tasks. See "Specifying time constraints between dependent tasks" on page 57.

Task is available for...	Task is available in...
Bravo Platform	Main Protocol, Bravo Subprocess
Vertical Pipetting Station	Main Protocol, Vertical Pipetting Station Subprocess

Requirements

You cannot add a Change Instance task within the grouped tasks.

Task parameters

The Group Begin and Group End tasks do not have task parameters.

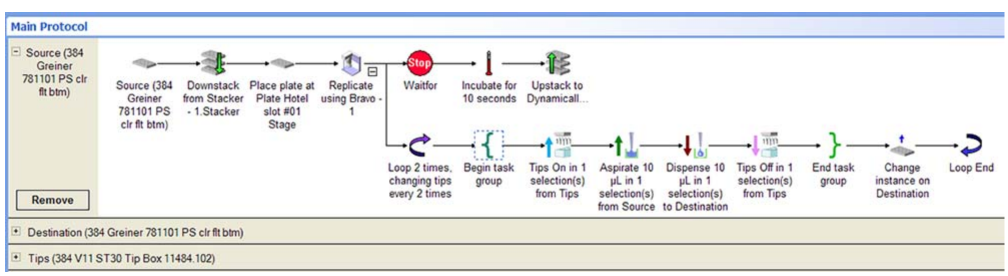
Example: Group liquid-handling tasks

Goal

Make sure all liquid-handling tasks are finished before incubation begins.

Implementation

Add the Group Begin and Group End tasks around the liquid-handling tasks. Notice that the Change Instance task is not within the group.



Related information

For information about...	See...
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Microplate-handling tasks	"Setting parameters for scheduling tasks" on page 557
Microplate-storage tasks	"Setting parameters for scheduling tasks" on page 557

For information about...	See...
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

JavaScript

Description

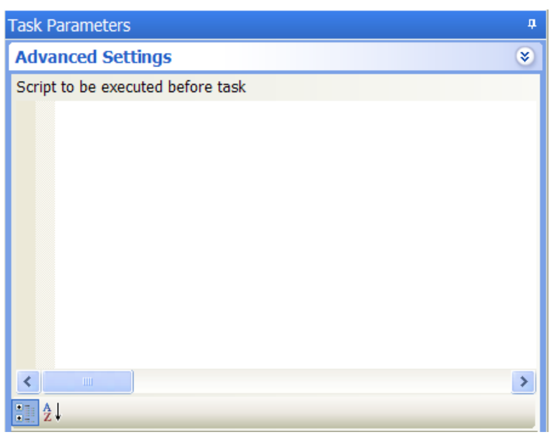
The JavaScript task (JavaScript) runs the specified JavaScript at the desired point in the protocol. Typically, you use the JavaScript task to run a program that is independent of any task.

Note: If you want to run a JavaScript that skips a task or changes the parameters of the task, write the JavaScript code in the Advanced Settings area of that task. See "Using JavaScript" on page 84 for more information.

Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Advanced Settings

After adding the JavaScript task at the desired point in the protocol, write the code in the **Advanced Settings** area:



Related information

For information about...	See...
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Writing JavaScript in a protocol	"Using JavaScript" on page 84
Define Variables task	"Define Variables" on page 564
Task parameter variables	"Using simple variables" on page 77
Adding start and finish protocol scripts	"Using start and finish protocol scripts" on page 127

For information about...	See...
JavaScript utilities	“Using JavaScript utilities” on page 128
Microplate-handling tasks	“Setting parameters for scheduling tasks” on page 557
Microplate-storage tasks	“Setting parameters for scheduling tasks” on page 557
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Loop and Loop End

Description

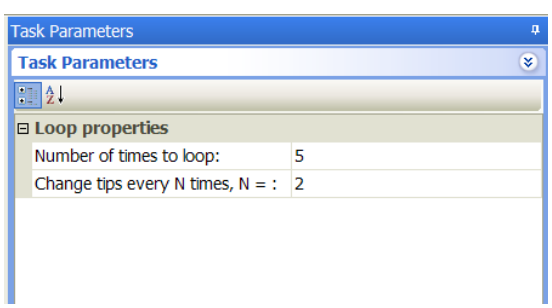
The Loop (Loop) and Loop End (Loop End) tasks are used together to repeat a set of tasks that are within the two tasks.

Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Task parameters

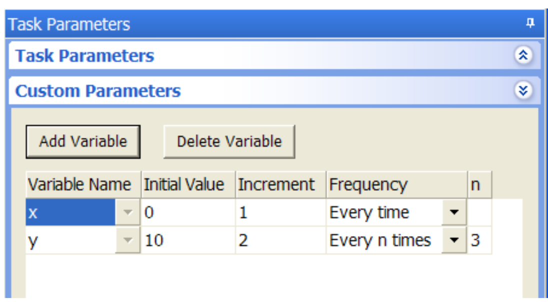
Note: Loop End does not have any task parameters.

After adding the Loop task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Number of times to loop	The number of times you want to run the tasks within the loop.
Change tips every N times, N=	The number of times to run the tasks in the loop before changing the tips. For example, if N is set to 2, the Tips On task will be run every odd number of the loop: first, third, fifth, and so on. The Tips Off and Change Instance tasks that operate on the tip box used by the Tips On task will run every even number of the loop: second, fourth, sixth, and so on. *Note:* If the loop does not contain any tasks that require pipette tips, the software ignores this parameter.

If you want to change the value of variables during the looping process, click **Custom Parameters** and set the following:



Parameter	Description
Add Variable	The command that adds a variable in the Variable table. To add a variable, click **Add Variable.** A blank row appears in the table. Select the desired value for each column.
Delete Variable	The command that deletes the selected variable from the table.
Variable table	The table that lists the variables to be incremented or decremented in the loop.

Parameter	Description
Variable name	The variable you want to increment or decrement. Select the variable from the list. The variables are defined in the Define Variables task, the Define Plate Set task, or the JavaScript code in the Advanced Settings area. If the variable is new and was not previously defined elsewhere, and an initial value is given, the variable will be defined the moment the Loop task runs.
Initial value	The starting value to be used the first time through the loop. For variables, the Initial value is the starting value of the variable. For plate sets, the Initial value is the starting array index value. *Note:* The starting variable value specified in this task overrides the starting variable value the operator provides during run time.
Increment	The amount you want the variable to increment or decrement.
Frequency	The frequency of the increment or decrement: • *Every time*. The variable should increment or decrement every loop. • *Every n times*. The variable should increment or decrement every specified number of times. • *After first time*. The variable should increment or decrement starting from the second loop. • *After last time*. The variable should increment or decrement after the last loop.
n	The number of times to loop before a variable increments or decrements. This field is available only if you selected Every n times.

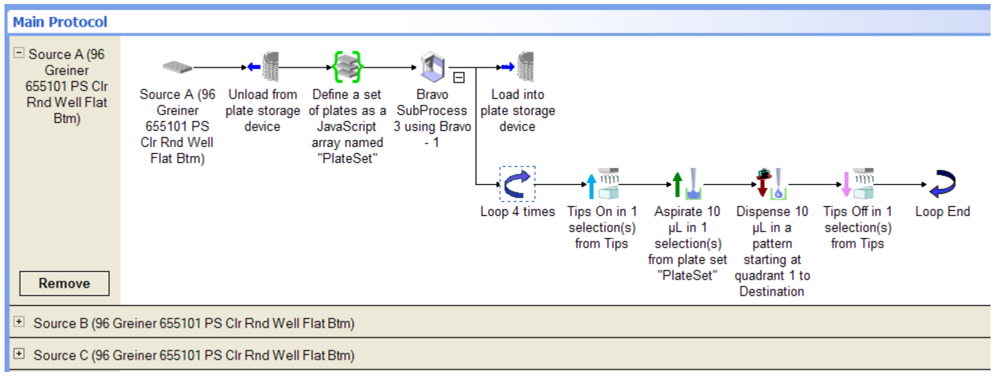
Example: Use the Loop task for plate set indexing

Goal

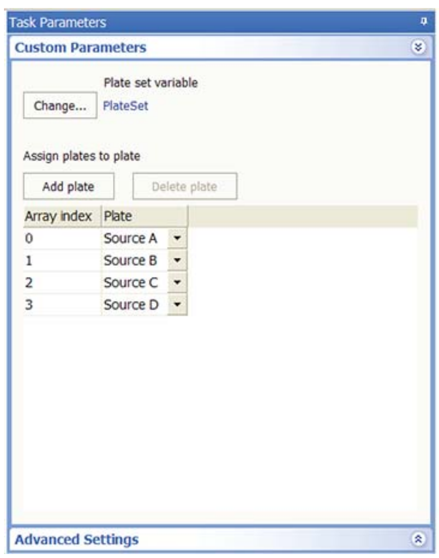
In a protocol where a plate set variable is defined, use the Loop task to specify the starting plate set (or array) index, how frequently the variable will increment or decrement, and the amount by which the index will increment or decrement.

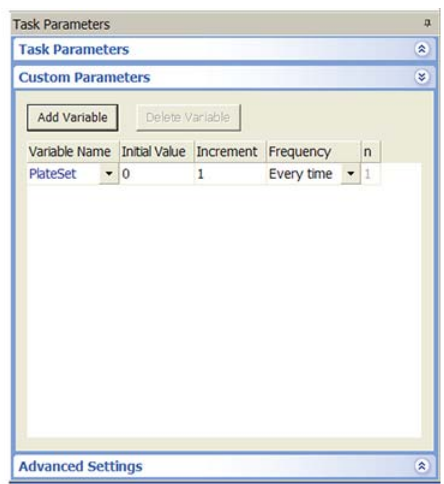
Implementation

In the following protocol example, a plate set is defined and used in the liquid-handling tasks.



In the Define Plate Set Task Parameters area, an array of source microplates is defined and indexed (0, 1, 2, and 3). In the Loop Custom Parameters area, the PlateSet variable is selected. The Initial Value is set to 0 to match the first PlateSet index (0). The PlateSet variable will increment by 1 every time through the loop. So the first time through the loop, PlateSet index 0, or SourceA is processed. The second time through the loop, PlateSet index 1, or Source B is processed, and so on.





Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Define Plate Set task	"Define Plate Set" on page 560

For information about...	See...
Task parameter variables	“Using simple variables” on page 77
Microplate-handling tasks	“Setting parameters for scheduling tasks” on page 557
Microplate-storage tasks	“Setting parameters for scheduling tasks” on page 557
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
Scheduling tasks	“Setting parameters for scheduling tasks” on page 557

Print

Description

The Print task (Print) prints barcode labels using the Microplate Labeler. The task does not apply labels to labware.

Use this task if you want to print labels and manually apply the labels to labware such as tubes or other containers. You can also apply the labels to pages in your lab notebook for record keeping purposes.

Task is available for...	Task is available in...
Microplate Labeler (standalone only)	Startup Protocol Cleanup Protocol

Requirements

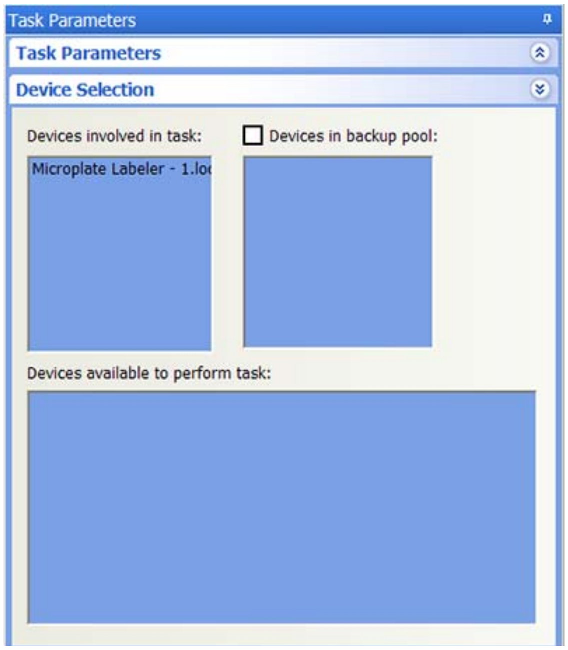
The requirements for the Print task is identical to the requirements for the Print and Apply task. See “Print and Apply” on page 318.

Selecting devices for the task

You must select a device for the Print task before you can set the task parameters. After adding the Print task in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

To select a device for the task:

Double-click the desired device in the **Devices available to perform task** area to move it to the **Devices involved in task** area.



Setting the task parameters

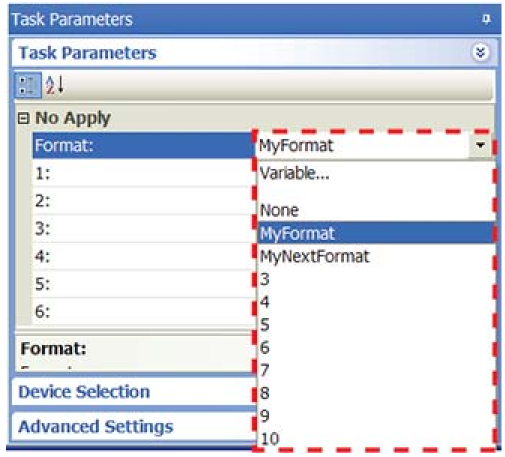
IMPORTANT Make sure the label formats are uploaded to the printer. Be sure to initialize the Microplate Labeler device before you set the task parameters.

After selecting the device to use for the Print and Apply task, you can set the parameters in the **Task Parameters** area. The area lists the four sides of a microplate (south, west, north, and east). For each side, you can select a label format and specify the data that will substitute for the text and barcode fields in the label format.

CAUTION Format selection and field information are saved with the protocol. If the formats on the printer are changed, initializing the device will overwrite the information in the protocol. For example, suppose you created a protocol and selected a format called MyFormat. Later, MyFormat was deleted from the printer. The next time you initialize the device and open the protocol, MyFormat will not appear in the protocol.

To set the task parameters:

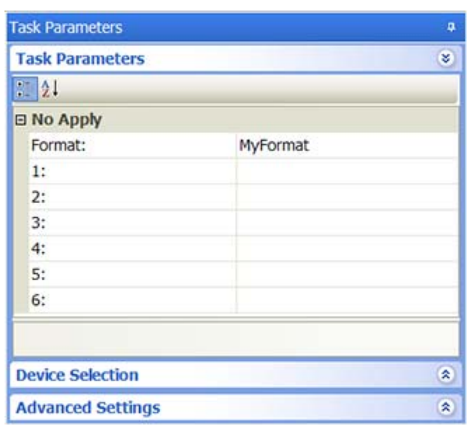
1 Select the barcode format in the **Format** list:



Format selection	Description
None	Indicates no label will be printed.
Format name or number	Uses a format that was set up in Microplate Labeler Diagnostics. *Note:* If you do not see a list of formats, make sure the label formats are not empty (formats must contain at least one field), the formats are uploaded to the printer, and the Microplate Labeler device is initialized.

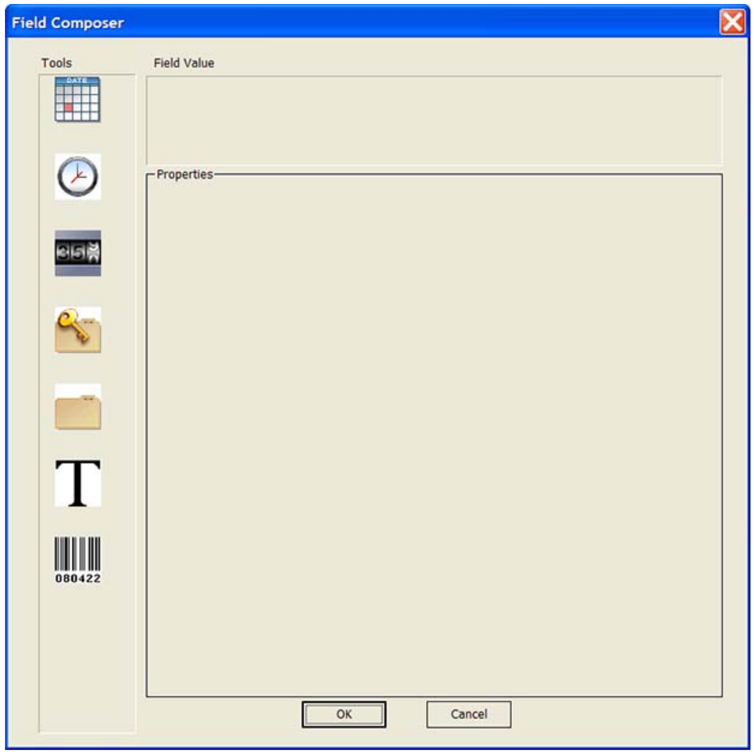
As soon as you select a format, fields appear in the Task Parameters area. You can specify the information you want to print in these fields.

Note: The number of fields that appear depends on the format you select.



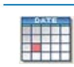
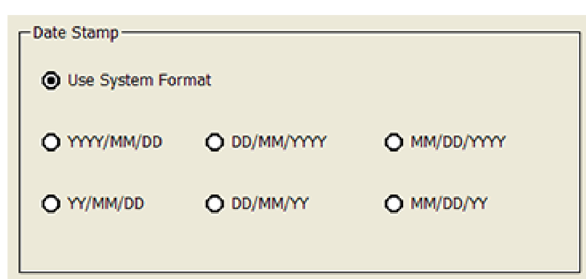
2 Click a field, and then click the ... button that appears. The Field Composer dialog box opens.

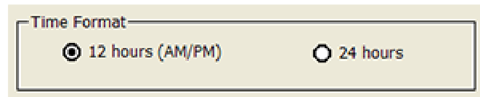
The Field Composer allows you to specify the information to print on the barcode label. For example, you can print the current date and time.



3 In the **Tools** area, double-click one or more of the following icons to specify the information to be printed on the barcode label. The selected icon appears in the **Field Value** area.

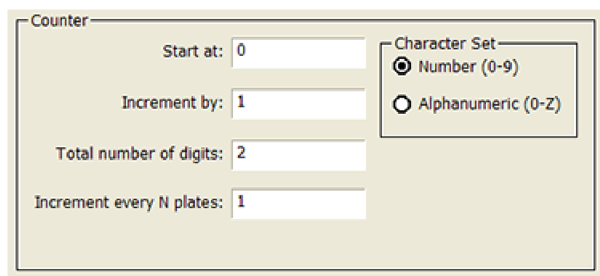
IMPORTANT For field limitations, such as the maximum number of characters permitted or symbology-dependent limitations, check the format you set up in Microplate Labeler Diagnostics. See also the *G5404B Microplate Labeler User Guide*.

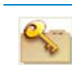
Icon	Description
	Prints the current date. Click the icon in the **Field Value** area. In the **Properties** area, select the desired date format. YYYY is the year, MM is the month, and DD is the day. *Note:* The Use System Format option uses the local computer’s date format. 

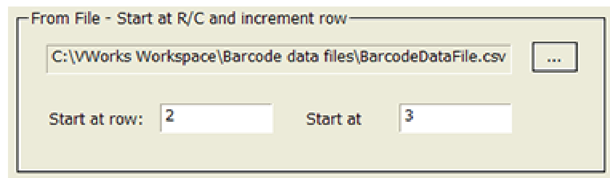
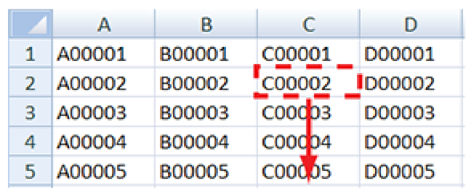
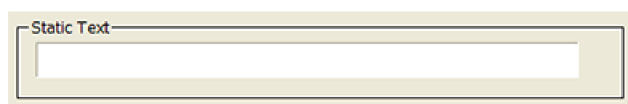
<table>
<tr><th>Icon</th><th>Description</th></tr>
<tr><td></td><td>
Prints the current time.
Click the icon in the Field Value area. In the Properties area, select the desired data format: 12 hours (AM/PM) or 24 hours.


</td></tr>
<tr><td></td><td>
Prints a numeric or alphanumeric value that can be incremented.
Click the icon in the Field Value area. Set the following in the Properties area:

Character Set. The option to use either numeric or alphanumeric characters.
Start at. The starting value.
Increment by. The amount by which the value increments.
Total number of digits. The total number of digits or characters, including leading 0s.
Increment every N plates. The increment value. For example, 1 increments the value every microplate.



</td></tr>
<tr><td></td><td>Not used. If you add this icon, the software will ignore it.</td></tr>
</table>

Icon	Description
	Prints the values in the specified barcode input file. Click the icon in the **Field Value** area. In the **Properties** area, click the ... button to select the input file. Specify the row and column number of the starting value. During the run, the software automatically increments to the next row to print the next value. For example, Start at row is set at 2, Start at col is set at 3. During the run, the software starts with the value in row 2 column 3 (C00002). Then, the software moves to row 3 column 3 (C00003), row 4 column 3 (C00004), row 5 column 3 (C00005), and so on.   (A: A00001–A00005; B: B00001–B00005; C: C00001–C00005; D: D00001–D00005; rows 1–5)
	Prints the text you specify. Click the icon in the **Field Value** area. In the **Properties** area, type the text you want to print on every microplate. 
	Not used. If you add this icon, the software will ignore it.

4 When you are finished, click **OK.** The information you specified appears in the Task Parameters area.

Example

Goal

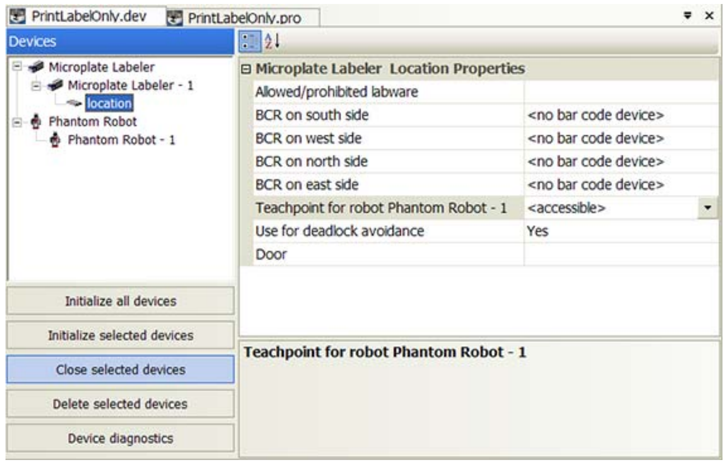
Using the Microplate Labeler, print values read from a barcode data file. The software should start reading the file from row 1 column 1.

The software should pause after it prints a label to permit you to remove the label from the device and manually apply it to a page in your lab notebook.

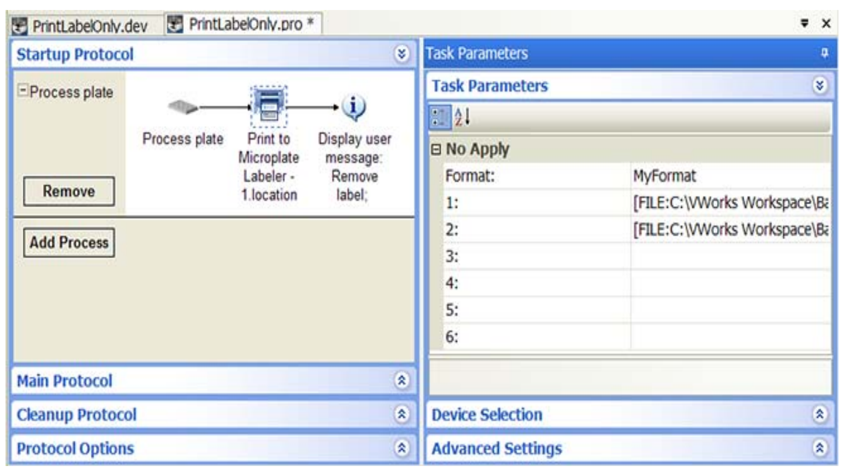
Implementation

Note: This example assumes that the Microplate Labeler is set up correctly and the format, MyFormat, is already defined and loaded to the printer. MyFormat contains two fields. Field 1 is a human-readable text field. Field 2 is a barcode field.

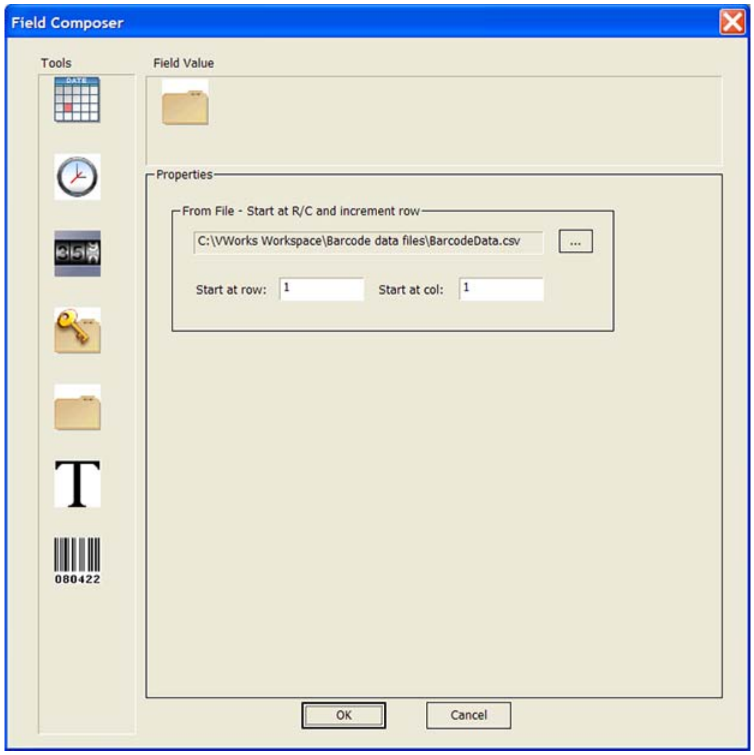
Create a device file that contains both the Microplate Labeler and a Phantom Robot, as shown. In the **Microplate Labeler Location Properties** area, make sure the **Teachpoint for Phantom Robot** is **<accessible>.**



Create a Startup Protocol as shown and add a Print task and a User Message task. The User Message task creates the pause to permit you to remove the label from the device. When setting the Print and Apply task parameter, select MyFormat.



For each of the two format fields (1 and 2), open the Field Composer dialog box and double-click the data file icon in the **Tools** area to add it to the Field Value area. In the **Properties** area, locate and select the data file. In the **Start at row** box, type 1. In the **Start at col** box, type 1.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Print and Apply task	"Print and Apply" on page 318
Microplate-handling tasks	"Setting parameters for scheduling tasks" on page 557
Microplate-storage tasks	"Setting parameters for scheduling tasks" on page 557
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Process Control

Description

The Process Control task (Process Control) is used to initiate a specific process within the same protocol based on an upstream condition or the type of labware entering the process.

Task is available for...	Task is available in...
Any device	Main Protocol

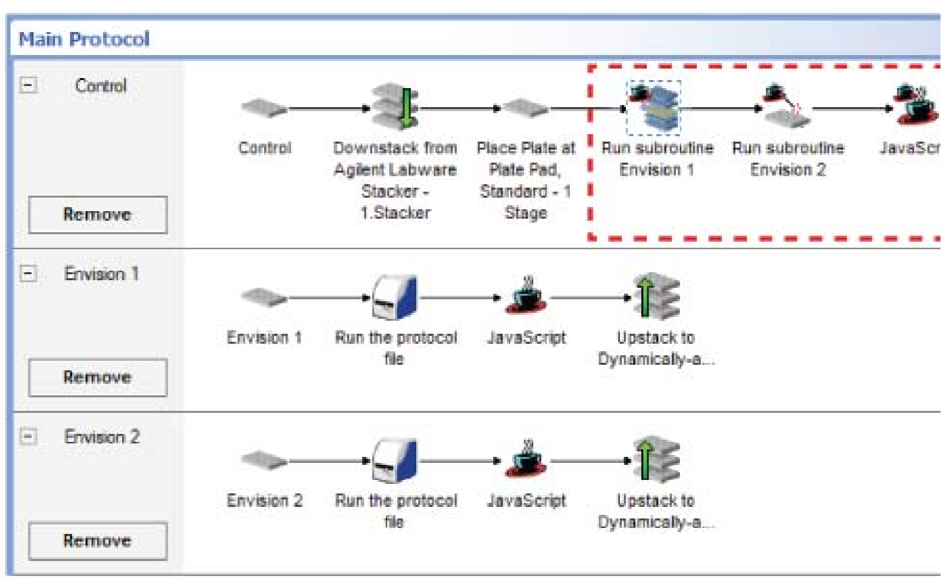
When to use Process Control instead of Spawn Process

The Process Control task extends the functions of a Spawn Process task with simplified implementation in a protocol. Instead of adding multiple Spawn Process tasks in one process, you can simplify the protocol by adding a single Process Control task.

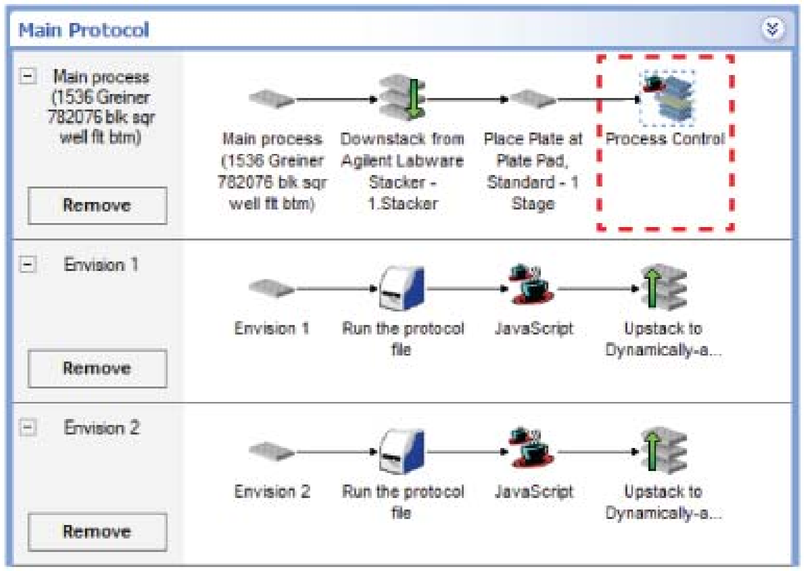
For example, suppose you have two devices in a device pool. You want to design the protocol such that if the first device is busy, the system will use the second device.

You can write the protocol in one of two ways:

- Add two Spawn Process tasks, one for using each device. The required JavaScript will be distributed across the two Spawn Process tasks and a separate JavaScript task, thus making debugging and future updates difficult.

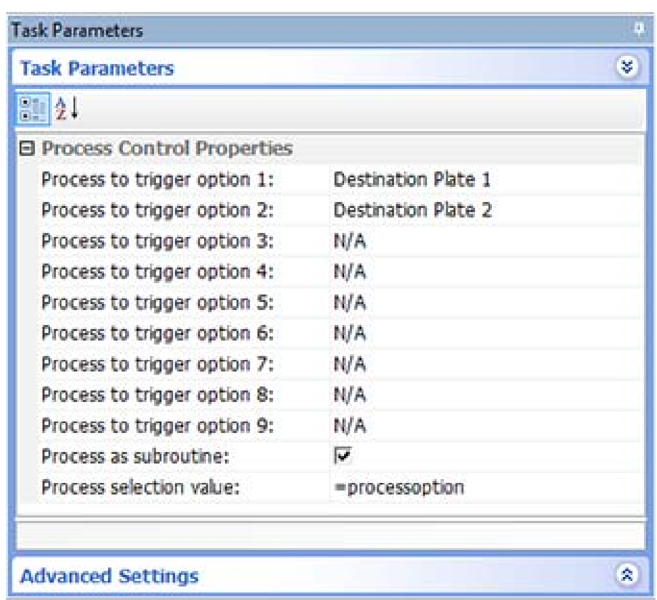


- Add a single Process Control task that controls the use of both devices. The required JavaScript is consolidated in the Process Control task, thus facilitating debugging and future updates.



Task parameters

After adding the Process Control task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Process to trigger option 1–9	One of the processes that will be initiated when the run reaches the Process Control task. Select the process that you want to initiate. For example, if you have two potential processes to initiate, Destination1 and Destination2, select Destination1 for Process to trigger option 1 and Destination2 for Process to trigger option 2. *Note:* The **Process selection value** determines which process will be initiated.
Process as subroutine	The option to run the initiated process as a subroutine (or subprocess) of the current process. Select the check box if you have data, such as barcode information, from the current process that you want to pass to the initiated process. Clear the check box if the two processes can run in parallel when the current process reaches the Process Control task.
Process selection value	The value that determines which process to initiate. For example, if you specify 5, then the process you selected for Process to trigger option 5 will initiate. You can also specify a variable. During the protocol, the value of the variable (1, 2, 3,..., or 9) is passed to the Process Controls task from an upstream task or from the Process Control task JavaScript. The value determines which process to run. If the value is 1, the specified process for **Process to trigger option 1** will initiate; if the value is 2, the specified process for **Process to trigger option 2** will initiate; and so on.

Example

Goal

Downstack microplates from the Labware Stacker, scan the microplate barcode, read the microplate using an available Envision Reader, and then upstack the microplate to an available Labware Stacker.

Use the pool of two Envision Readers during the protocol run:

- If the first reader is busy, use the second reader.
- If the second reader is busy, use the first reader.
- If both readers are busy, wait 2 seconds and check each reader again.

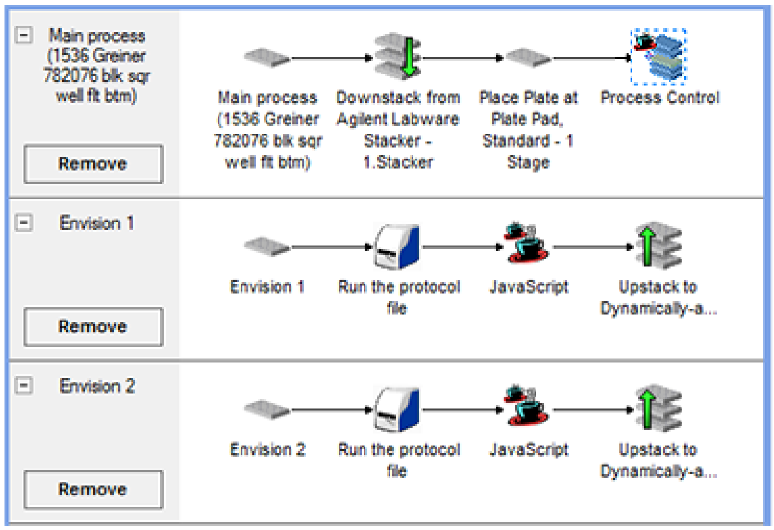
Set variables for the following purposes:

- Determine which Envision Reader to use. In the example, the variable used for this purpose is `procoption`.
- Determine the state of the first Envision Reader. Valid values are true (busy) and false (available for use). In the example, the variable used for this purpose is `bEnvision1Busy`.
- Determine the state of the second Envision Reader. Valid values are true (busy) and false (available for use). In the example, the variable used for this purpose is `bEnvision2Busy`.

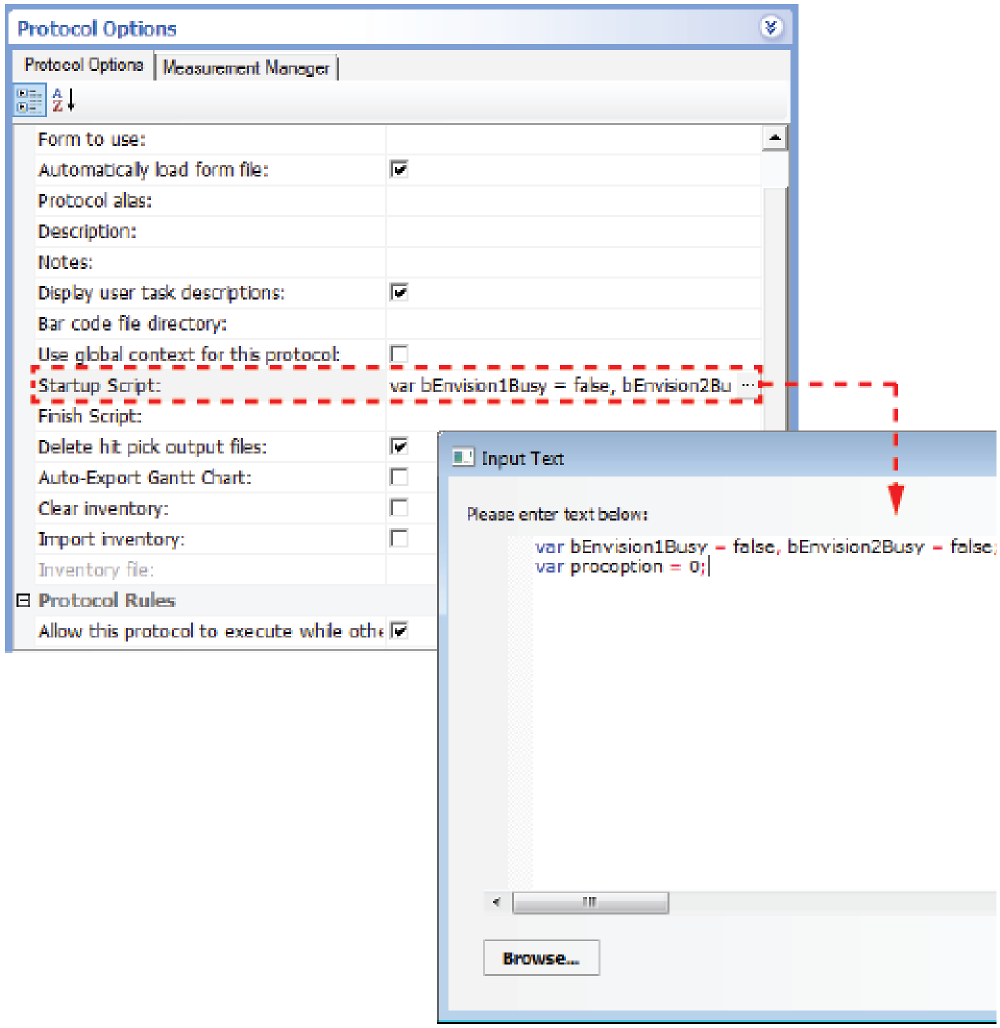
Implementation

Create a protocol as shown.

Note: The barcode reader is attached to the platepad, so a Place Plate task is added to scan the microplate barcode at the platepad.

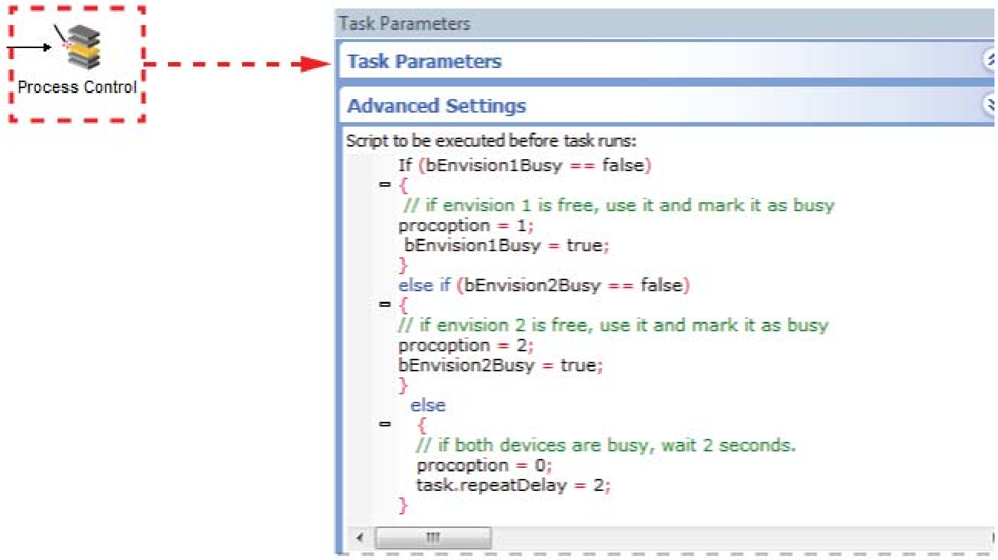


In the **Protocol Options Startup Script**, add JavaScript to reset all relevant variables at the start of the run. Set the Envision Reader busy state (`bEnvision1Busy` and `bEnvision2Busy`) to false, and set the `procoption` to 0.



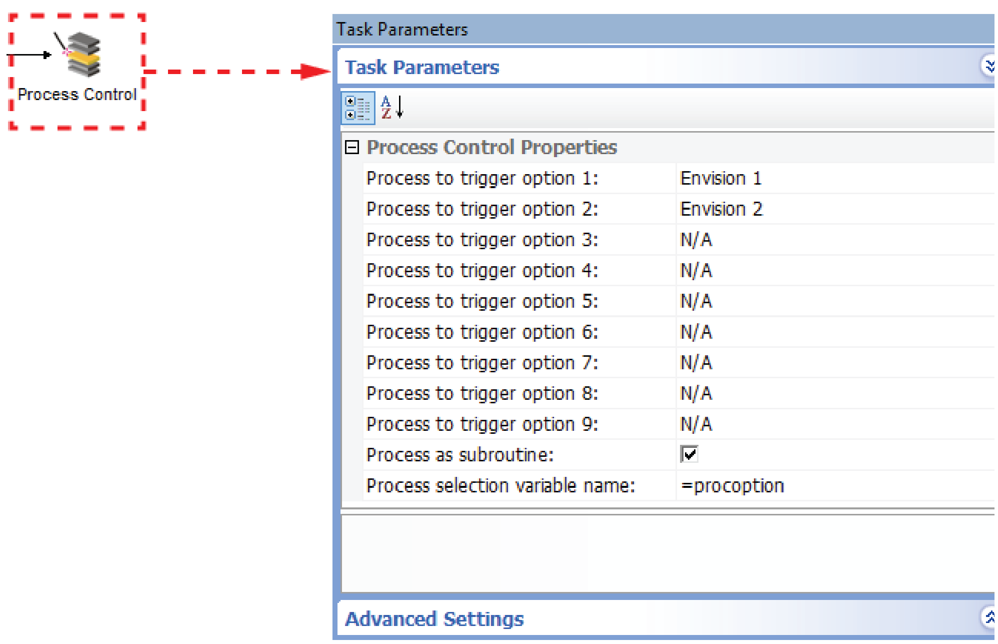
Before setting the task parameters, add JavaScript to the **Process Control** task to determine which Envision Reader to use. (The script is run before the Process Control task is performed.) The script logic is as follows:

- The system checks to see if the first Envision Reader is available. If it is, `procoption` is set to 1.
- If the first Envision Reader is busy, the system checks the second Envision Reader. If it is available, `procoption` is set to 2.
- If both devices are busy, the system will wait 2 seconds before checking the devices again.

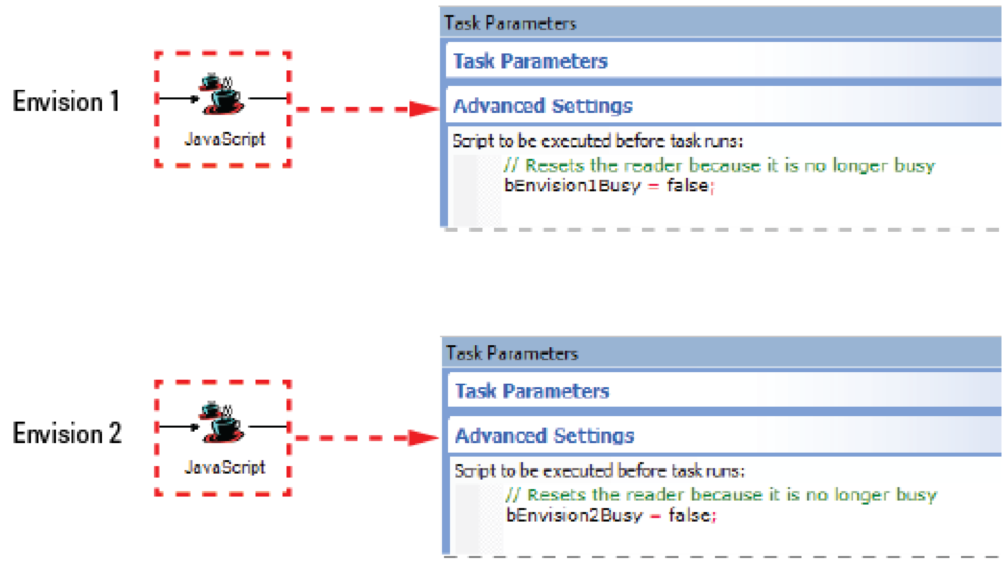


Set the **Process Control Task Parameters** as shown.

- Notice that the process selection variable is `procoption`. This variable determines which process to initiate. The value of this variable is passed from the Process Control JavaScript. If `procoption` = 1, the task will initiate option 1: Envision 1. If `procoption` = 2, the task will initiate option 2: Envision 2.
- Select **Process as subroutine**. The microplate barcode information is retained during the Envision processes.



Add the JavaScript as shown in each Envision Reader process to set the Envision Reader busy state to false. After the microplate-reading is finished, the devices are available for other labware.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Wait For task	"Wait For" on page 603
Microplate-handling tasks	"Setting parameters for scheduling tasks" on page 557
Microplate-storage tasks	"Setting parameters for scheduling tasks" on page 557
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Signal

Description

The Signal task (Go Signal) permits another process that is currently in the wait state to continue to the next task. The wait state is from the Wait For task in the other process.

Note: Each Signal task can be used with multiple Wait For tasks.

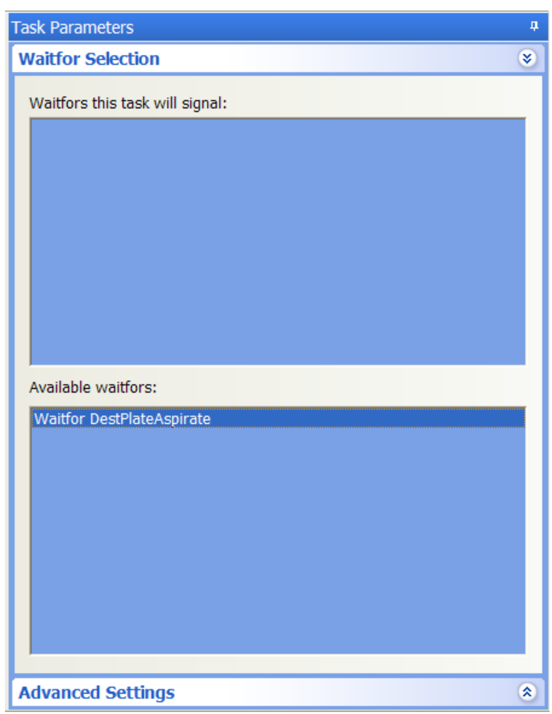
Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Requirements

You must first add the Wait For task at the desired point in the protocol before adding the Signal task.

Waitfor Selection

After adding the Signal task at the desired point in the protocol, select the corresponding Waitfor task in the **Waitfor Selection** area. Double-click the Waitfor task name in the **Available waitfors** area. The selected Waitfor task name appears in the **Waitfors this task will signal** area.



Example

See the example in "Wait For" on page 603.

Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Wait For task	"Wait For" on page 603
Microplate-handling tasks	"Setting parameters for scheduling tasks" on page 557
Microplate-storage tasks	"Setting parameters for scheduling tasks" on page 557
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Spawn Process

Description

The Spawn Process task (Spawn Process) is used to initiate another process within the same protocol. For example, to reduce evaporation, you can use Process Control task to deliver a certain labware into the system only when they are ready to be processed. You can also use the Process Control task with JavaScript code to use operator-supplied information to initiate a new process or funnel incoming labware into a different process depending on the barcode.

Task is available for...	Task is available in...
Any device	Main Protocol

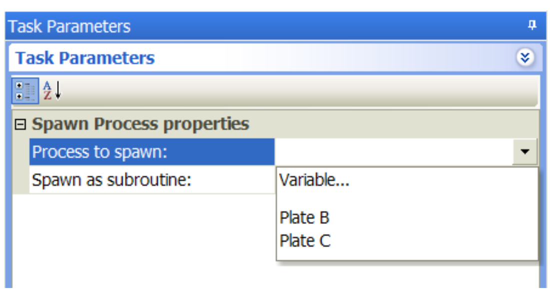
When to use Spawn Process instead of Process Control

The Spawn Process task provides a simple way to initiate another process in the same protocol. Use the task if you want to initiate only one or two processes in the protocol, and the spawning of the processes do not depend on varying run-time conditions.

If you want to initiate multiple processes depending on changing run-time conditions, use the Process Control task instead. See “Process Control” on page 586.

Task parameters

After adding the Spawn Process task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
Process to spawn	The process that starts to run when the current process reaches the Spawn Process task. Select from the list of processes.

Parameter	Description
Spawn as subroutine	The option to run the spawned process as a subroutine (or subprocess) of the current process. **IMPORTANT** If the spawned subprocess specifies a different labware type, the software ignores the labware specification. Select the option if you want the software to start the spawned process when the Spawn Process task is reached, and continue the current process after the spawned process is finished. Clear the check box if the two processes can run in parallel when the current process reaches the Spawn Process task.

Example 1

Goal

Replicate Source Plate on the Bravo Platform. Deliver the Diluent Plate from the BenchCel Microplate Handler into the system only when the Destination Plate is ready for dilution.

Implementation

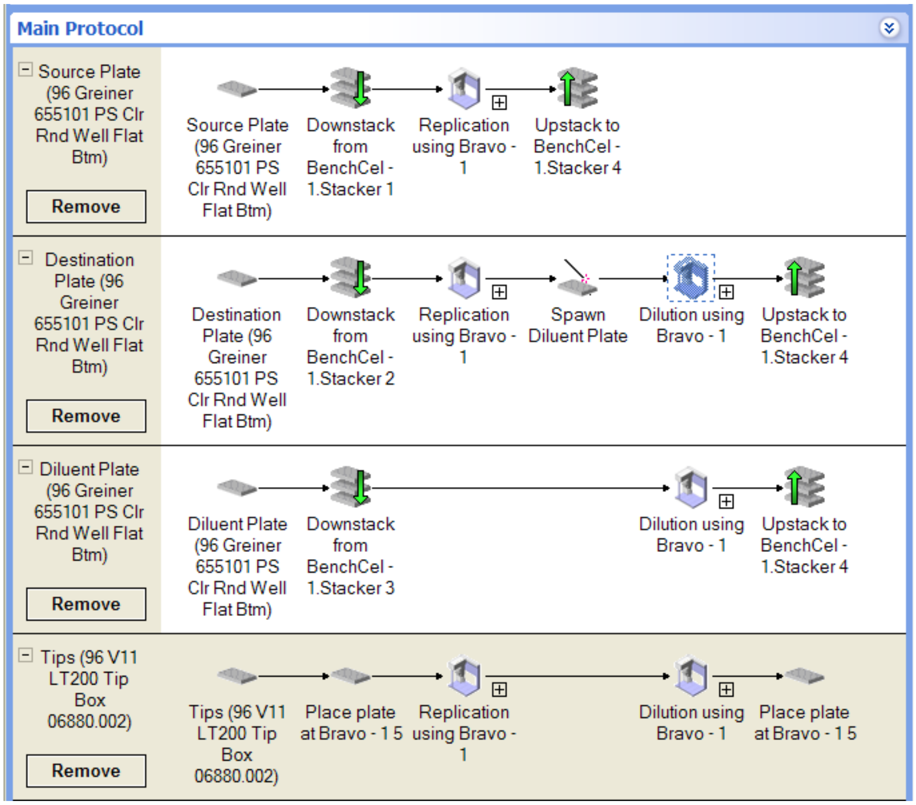
The Bravo Platform and BenchCel Microplate Handler in the example are configured as follows:

- The replication process will occur on the Bravo Platform.
- The Source Plate, Destination Plate, and Diluent Plate will be stored in different stacks in the BenchCel Workstation. At the end of the process, the microplates are stored in an empty stack in the BenchCel Microplate Handler.

When writing the protocol:

- Add a process for the Source Plate as shown. In the Source Plate process, add a Bravo Subprocess (renamed Replication in the example) that aspirates contents from the Source Plate and dispenses into the Destination Plate.
- Add a process for the Destination Plate as shown. In the Destination Plate process, add two Bravo Subprocesses:
  - The first subprocess (renamed Replication in the example) reflects the movement of contents from the Source Plate and into the Destination Plate. This is a copy of the tasks in the Source Plate process.
  - The second subprocess (renamed Dilution in the example) aspirates contents from the Diluent Plate and dispenses into the Destination Plate.
- Add a process for the Diluent Plate as shown. In the Diluent Plate process, add a subprocess (renamed Diluent in the example) that reflects the movement of contents from the Diluent Plate into the Destination Plate. This is a copy of the tasks in the Diluent subprocess in the Destination Plate process.
- Add a configured labware for the Tip Box as shown.

- In the Destination Plate process, add a Spawn Process task between the Replication subprocess and the Dilution subprocess. Doing so holds the Diluent Plate in the BenchCel stack until the Replication subprocess is finished and the Destination Plate is ready to receive the diluent.



Example 2

Goal

The same goal as Example 1, except:

- Ask the operator to specify the number of replications.
- Control the spawning of the Source Plate and Destination Plate processes based on the operator input.

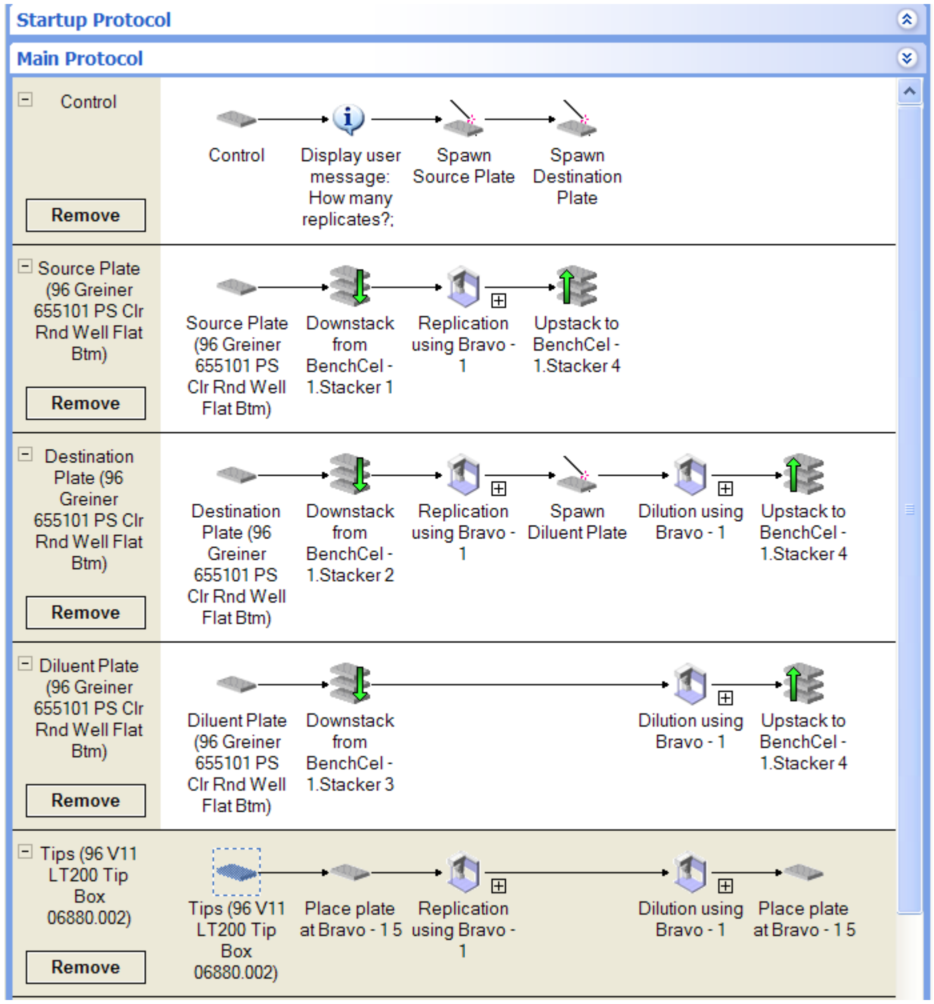
Implementation

The Bravo Platform and the BenchCel Microplate Handler are configured as described in Example 1. Set up the Source Plate, Destination Plate, and Diluent Plate processes as described in Example 1.

Add a new process called Control at the top of the Main Protocol as shown. Add the following tasks in the Control process:

- *User Message*. Asks the operator to specify the number of replications.
- *Spawn Process*. Starts the Source Plate process. In addition, provide JavaScript code in the Advanced Settings area to incorporate the input from the operator during the run. For example, if the operator specified two replications, the JavaScript code should use that value to produce two replications during the run.

- *Spawn Process*. Starts the Destination Plate process. Provide JavaScript code in the Advanced Settings area to incorporate the input from the operator during the run.



Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
Task parameter variables	"Using simple variables" on page 77
Microplate-handling tasks	"Setting parameters for scheduling tasks" on page 557
Microplate-storage tasks	"Setting parameters for scheduling tasks" on page 557

For information about...	See...
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

User Message

Description

The User Message task (User Message):

- Displays reminder messages. For example, the message can remind the operator to change pipette heads, empty the waste container, empty a reservoir, or replace labware.
- Prompts operators for variable values.

The protocol run is paused until the operator clicks Continue, or supplies the requested variable value and clicks Continue.

Note: Except for those that request variable input, user messages do not appear when running a protocol in simulation mode.

IMPORTANT Remove all user messages from protocols that will run unattended.

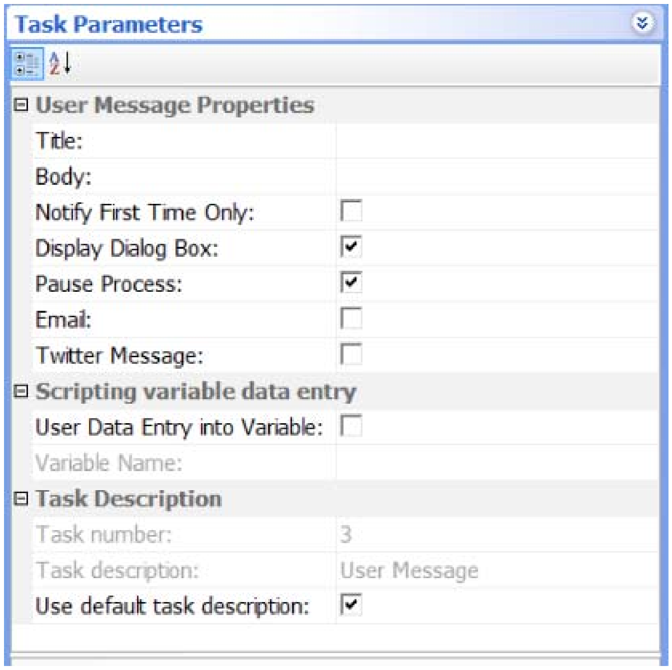
Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

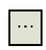
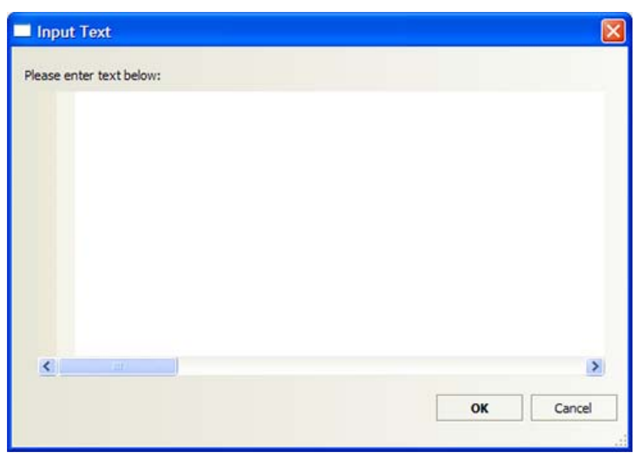
Requirements

If you want the User Message task to request variable input, add the variable in the desired tasks. For information, see "Using simple variables" on page 77.

Task parameters

After adding the User Message task at the desired point in the protocol, set the following parameters in the **Task Parameters** area:



Parameter	Description
User Message Properties	
Title	The text that appears in the titlebar of the message dialog box.
Body	The text that displays in the message dialog box. ***To enter the message using the Input Text dialog box:*** **1** Click the Body field in the Task Parameters area, and then click the  button that appears. The Input Text dialog box opens. **2** Type the message that you want to display, and then click OK. 

Parameter	Description
Notify First Time Only	The option to display the message the first time it appears in the protocol if you are running the protocol multiple times, or if the User Message task is in the loop.
Display Dialog Box	The option to display the message dialog box when the User Message task runs.
Pause Process	The option to pause the run when the User Message task runs. This option is only available when the Display Dialog Box option is selected.
Email	The option to send an email when the User Message task runs. The email subject is the message title and the email body is the text displayed in the message dialog box. To receive email, you must set up email notification. See “Setting up email notification” on page 232 for instructions on how to set up email notification. *Note:* If the email setup is incorrect, an error will be recorded in the main log but will not cause the run to pause.
Twitter Message	The option to send a message to a Twitter account when the User Message task run. The message contains the text displayed in the message dialog box. To receive a Twitter message, you must set up a Twitter account. See “Setting up automatic online notification” on page 235 for instructions on setting up online notification.
Scripting Variable Data Entry	
User Data Entry into Variable	The option to prompt the operator for a variable value.
Variable name	The name of the variable.
Task Description	
Task Number	The number that indicates the position of the task in the protocol.
Task Description	The description of the task.
Use Default Task Description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

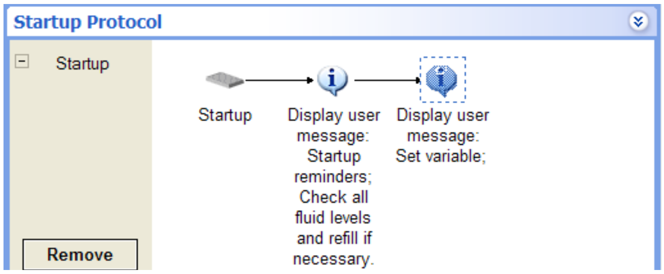
Example

Goal

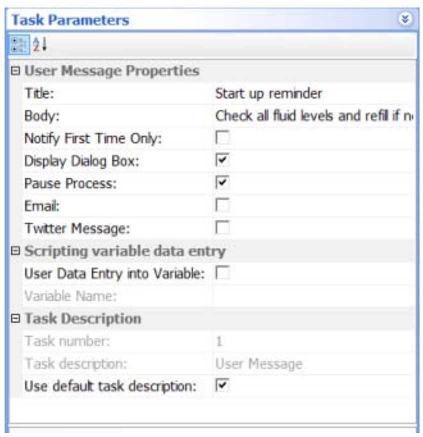
Display messages at the beginning of the protocol run to remind the operator to check fluid levels and to set the aspirate volume.

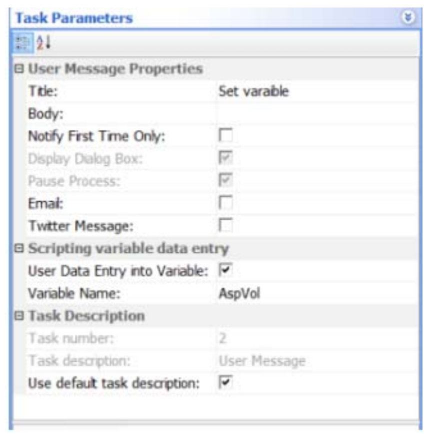
Implementation

In the Startup Protocol, add two User Message tasks. The first task asks the operator to check fluid levels. The second message asks the operator for the aspirate volume. AspVol is the variable defined for the aspirate volume.



The task parameters for both User Message tasks are as follows:



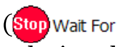


Related information

For information about...	See...
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Using simple variables	“Using simple variables” on page 77
Adding user message prompts	“Adding user message prompts” on page 82
Defining variables	“Define Variables” on page 564
Microplate-handling tasks	“Setting parameters for scheduling tasks” on page 557
Microplate-storage tasks	“Setting parameters for scheduling tasks” on page 557
Liquid-handling tasks	“Setting parameters for liquid-handling tasks” on page 387
I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269

Wait For

Description

In a multi-process protocol, the Wait For task (

) pauses the process that contains the task and waits for the go-ahead signal from another process before continuing to the next task. The go-ahead signal comes from the Signal task in the other process.

Note: Multiple Wait For tasks can be used with a single Signal task.

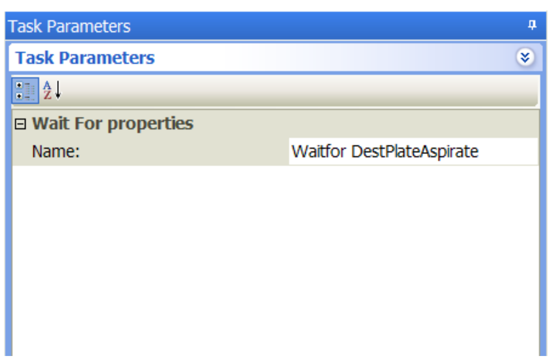
Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Requirements

The Wait For task is always used with the Signal task. You must first add the Wait For task at the desired point in the protocol before adding the Signal task.

Task parameters

After adding the Wait For task at the desired point in the protocol, double-click the corresponding Wait For task in the **Task Parameters** area:



Parameter	Description
Name	The name you want to assign the Wait For task. Type a name that describes the waitfor condition.

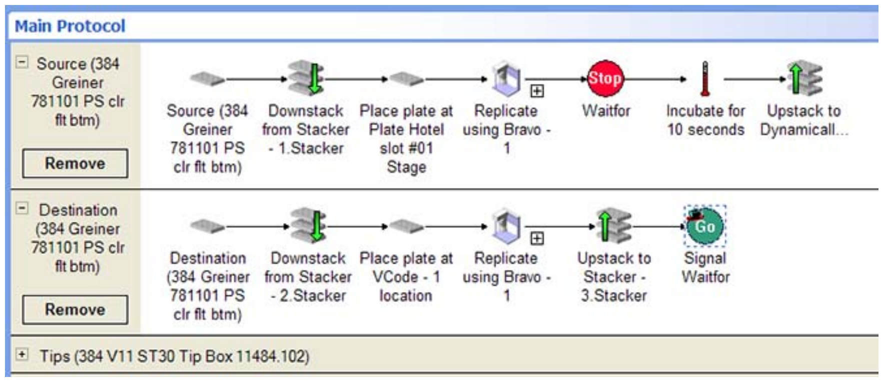
Example: Finish all liquid-handling tasks before incubation

Goal

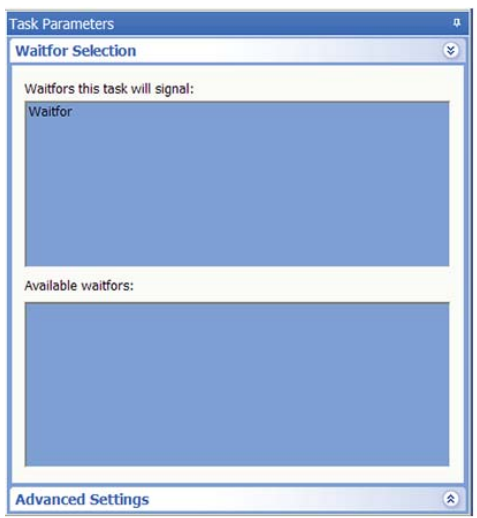
Make sure all liquid-handling tasks are finished before storing the source microplates.

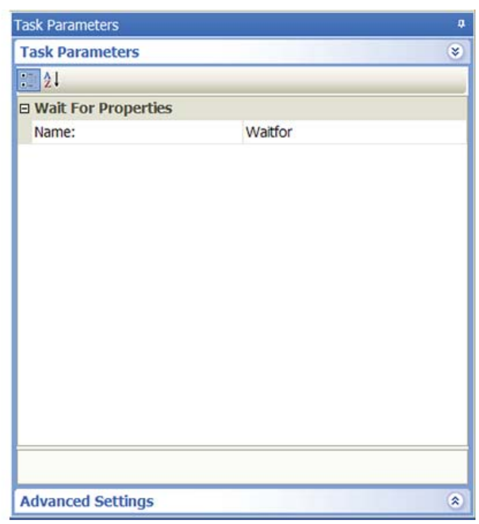
Implementation

In the protocol where the source microplate and destination microplates are separate processes, add a Wait For task in the source microplate process, after the liquid-handling tasks. Add a Signal task in the destination microplate process after the liquid-handling tasks.



The Wait For and Signal task parameters are set up as follows:





For other examples, see “Dismount” on page 301 and “Mount” on page 307.

Related information

For information about...	See...
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
Signal for task	“Signal” on page 593

For information about...	See...
Microplate-handling tasks	"Setting parameters for scheduling tasks" on page 557
Microplate-storage tasks	"Setting parameters for scheduling tasks" on page 557
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269

Wait for User (Bravo)

Description

The Wait For User task (Wait For User (Bravo)) pauses the protocol and waits until the operator presses the go button on the pendant. When the operator presses the go button, the protocol resumes.

You use the Wait For User task if you want to perform a task manually at the desired point in the protocol. For example, you can use the Wait For User task to pause the run so that you can manually replace a microplate on the deck. When you are finished, you press the go button on the pendant.

Note: Alternatively, you can use the User Message task to insert a pause in a protocol. Use the User Message task instead of the Wait For User task if you have easier access to the computer than the pendant during a run.

Task is available for...	Task is available in...
Any device	Startup Protocol Main Protocol Cleanup Protocol

Task parameters

The Wait For User task does not have task parameters.

Example

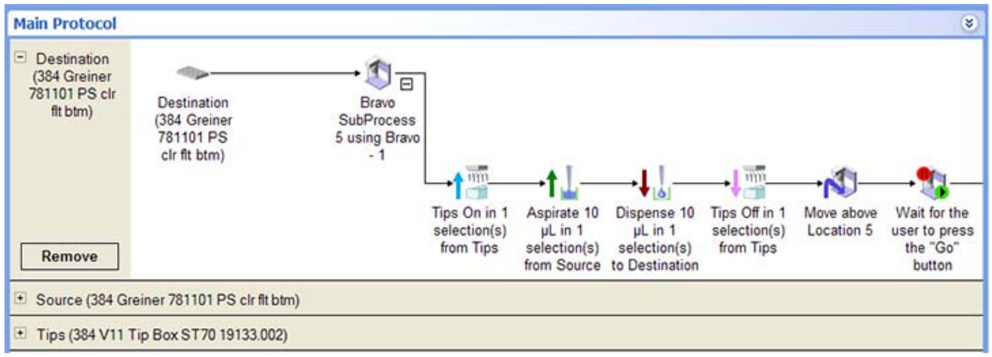
Goal

After the first dispense task, move the pipette head out of the way. Pause the run so that you can replace the destination microplate on the Bravo deck. Press the Go button on the pendant after the new destination microplate is placed.

Implementation

A Move to Location task is added after the first Dispense task to move the pipette head away from the current destination microplate. In the following example, because the destination microplate is at location 1, the pipette head is moved to location 5.

A Wait for User task is added after the Move to Location task to pause the run.

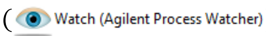


Related information

For information about...	See...
Adding devices	• "Adding devices" on page 25 • Device user guide
Adding tasks in a protocol	"Adding and deleting tasks" on page 53
User Message task	"User Message" on page 599
Microplate-handling tasks	"Setting parameters for scheduling tasks" on page 557
Microplate-storage tasks	"Setting parameters for scheduling tasks" on page 557
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387
Scheduling tasks	"Setting parameters for scheduling tasks" on page 557

Watch (Process Watcher)

Description

The Watch task (

) monitors the ...\VWorks Workspace\Inputs\Encore folder for new file activity. When the Watch task detects a new file in the folder, it runs the specified JavaScript file.

You use the Watch task if you are running an Encore Multispan System and you want to process labware at VWorks-controlled devices. For information about integrating VWorks-controlled devices with an Encore Multispan System, see the *Encore Multispan System* *knowledge base*.

Task is available for...	Task is available in...
Process Watcher device	Startup Protocol Main Protocol Cleanup Protocol

The Process Watcher device is a virtual device that monitors communication from the Encore Multispan System. For more information, see the *Encore Multispan System* *knowledge base*.

Note: Although you can add the Watch task at any point in a protocol, it is typically added in the Main Protocol.

Requirements

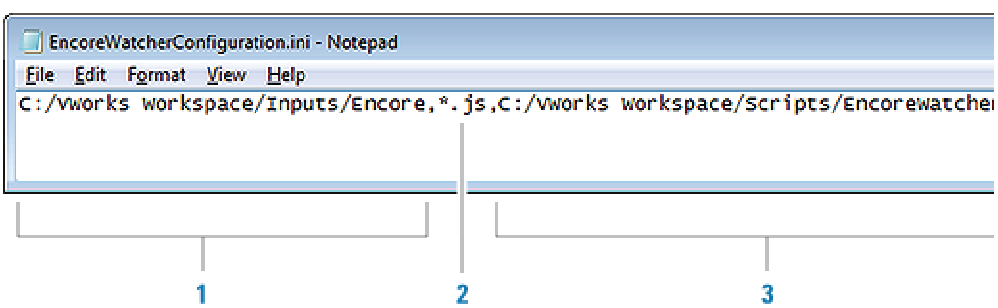
The Watch task requires the following:

- "EncoreWatcherConfiguration.ini file" on page 607
- "EncoreWatcher.js file" on page 608

EncoreWatcherConfiguration.ini file

The Watch task uses the information in the EncoreWatcherConfiguration.ini file to determine which folder to monitor and what JavaScript file to run.

The EncoreWatcherConfiguration.ini file is supplied with the VWorks software and resides in the C:\VWorks Workspace\INIs folder. The file specifies the following:



Item	Value	Description
1	Folder to monitor	The name and full path of the folder that the Watch task will monitor during the protocol run. The default folder specification is: C:/VWorks Workspace/Inputs/Encore *Note:* The path separators are forward slashes (/).
2	Input file to detect	The file that the Watch task will look for in the C:\VWorks Workspace\Inputs\Encore folder. The input file is any file that has the .js file name extension and contains the following information: • The VWorks protocol process to run • Optional data from the Encore Multispan protocol, such as task parameters for use by VWorks tasks The Encore Multispan software adds the .js file in the Encore folder just before the robot moves labware to the VWorks-controlled device.
3	JavaScript file to run	The name and full path of the JavaScript file that will run after the input file is detected. A default JavaScript file, EncoreWatcher.js, is supplied with the VWorks software and resides in the C:\VWorks Workspace\Scripts folder. For more information, see "EncoreWatcher.js file" on page 608.
4	JavaScript function to run	The JavaScript function that the EncoreWatcher.js file will run. By default, the function is WatcherMain.

You can use the default configuration file and modify items **1**, **3**, and **4**. You can also store the configuration file in a different folder. If you decide to store the configuration file in another folder, be sure to specify the location when you set the Watch task parameters.

EncoreWatcher.js file

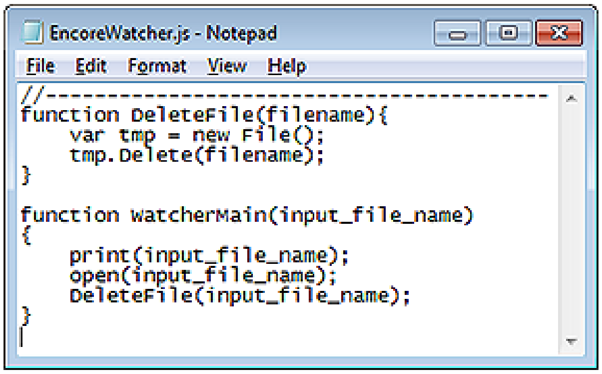
After the Watch task detects an input file in the ...VWorks Workspace\Inputs\Encore folder, it will run EncoreWatcher.js (item **3** in the configuration file) to do the following:

- Read the input (.js) file to get the VWorks protocol process information and any optional data (such as task parameter values).

  Note: To parse the optional data, you need to add the Parse Encore Spawn Parameter task in the VWorks protocol. For information about where to add the task in the protocol, see the *Encore Multispan System knowledge base*.

- Delete the input (.js) file in the ...VWorks Workspace\Inputs\Encore folder.

The following figure shows the contents of the default EncoreWatcher.js file. The code deletes the detected input file and runs the WatcherMain JavaScript function.



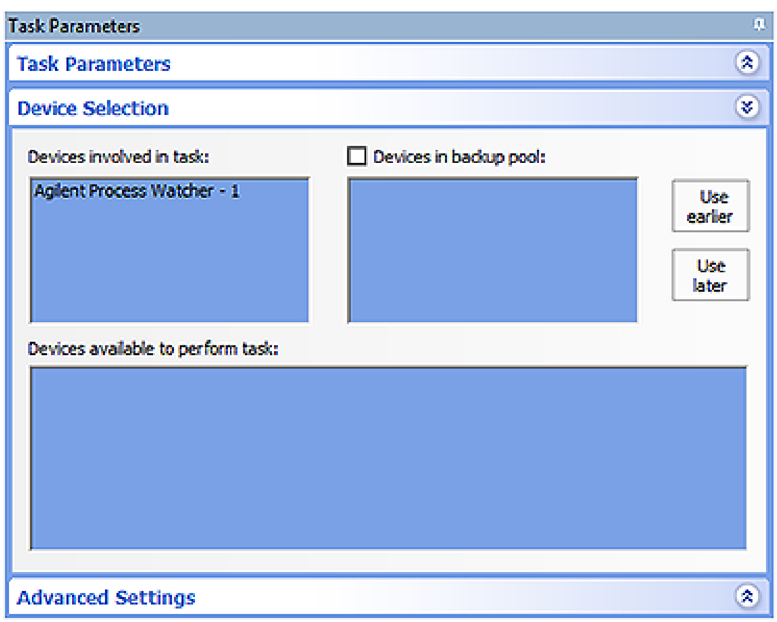
If you want to modify the EncoreWatcher.js file or create your own JavaScript file, see "Guidelines for creating your script" on page 687. (The guidelines are written for the Watcher tool. However, the principles of the guidelines are the same for the Watch task JavaScript file.)

Selecting devices for the task

Before setting the Watch task parameters, you must select a device for the task. After adding the Watch task in the protocol, select the task, and then click **Device Selection** in the **Task Parameters** area.

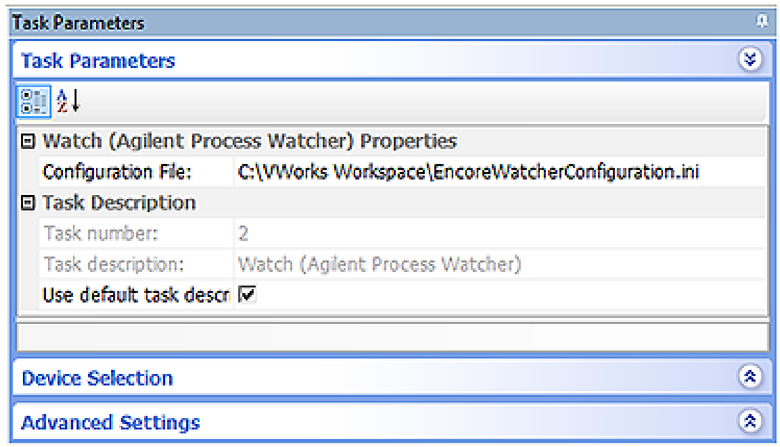
To select a device for the task:

Double-click the Process Watcher device in the **Devices available to perform task** area to move it to the **Devices involved in task** area.



Task parameters

After selecting the device for the Watch task, set the following parameters in the **Task Parameters** area:



Parameter	Description
Configuration File	The full path to the EncoreWatcherConfiguration.ini file. By default, it is stored in C:\VWorks Workspace. For more information, see “EncoreWatcherConfiguration.ini file” on page 607.
Task number	The number that indicates the position of the task in the protocol.
Task description	The description of the task.
Use default task description	The option to use the default task description or provide your own description for the task. Select the check box to use the default description. Clear the check box to provide your own description.

Related information

For information about...	See...
Running VWorks-controlled devices with the Encore Multispan System	*Encore Multispan System knowledge base*
Adding devices	• “Adding devices” on page 25 • Device user guide
Adding tasks in a protocol	“Adding and deleting tasks” on page 53
User Message task	“User Message” on page 599

For information about...	See...
Microplate-handling tasks	"Setting parameters for microplate-handling tasks" on page 277
Microplate-storage tasks	"Setting parameters for microplate storage tasks" on page 349
Liquid-handling tasks	"Setting parameters for liquid-handling tasks" on page 387

12
Specifying pipetting techniques

This chapter contains the following topics:

About the Pipette Technique Editor

Pipette techniques

You can define a Pipette Technique to use different pipetting methods. Different applications can benefit from different pipetting methods. For example, in multiplexed microplates, dispensing at an offset from the well center can improve distribution of the fluid. In cell-based assays, moving the pipette to the side of the wells minimizes the removal of the cells from the center or reading area of the well.

Pipette Technique Editor

You use the Pipette Technique Editor to define any number of Pipette Techniques. After you create a technique, it becomes available for the following tasks in any Bravo or Vertical Pipetting Station Subprocess:

- Aspirate
- Dispense
- Mix
- Pin Tool

Related information

For information about...	See...
Creating and editing Pipette Techniques	"Creating and editing pipetting techniques" on page 615
Managing Pipette Techniques	"Managing pipetting techniques" on page 619
Pipette Technique files	"Storing Pipette Technique files" on page 620
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
Mix task	"Mix (Bravo, Vertical Pipetting Station)" on page 497
Pin Tool task	"Pin Tool (Bravo, Vertical Pipetting Station)" on page 511

Creating and editing pipetting techniques

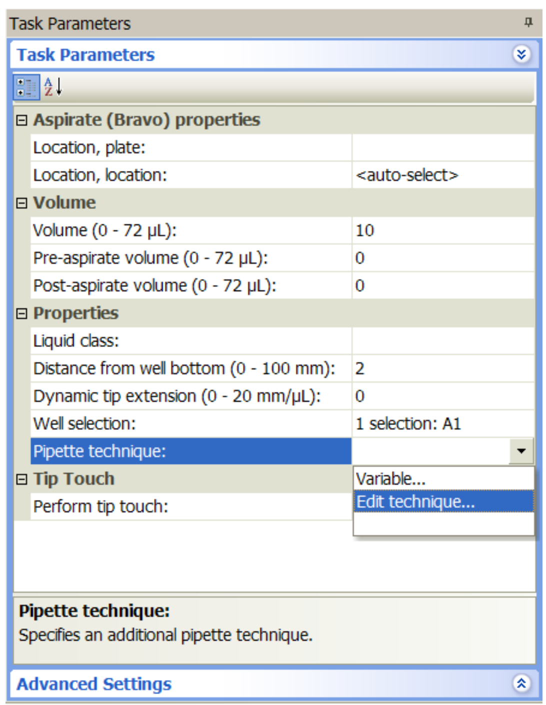
Creating a pipette technique

You can create a Pipette Technique in two ways:

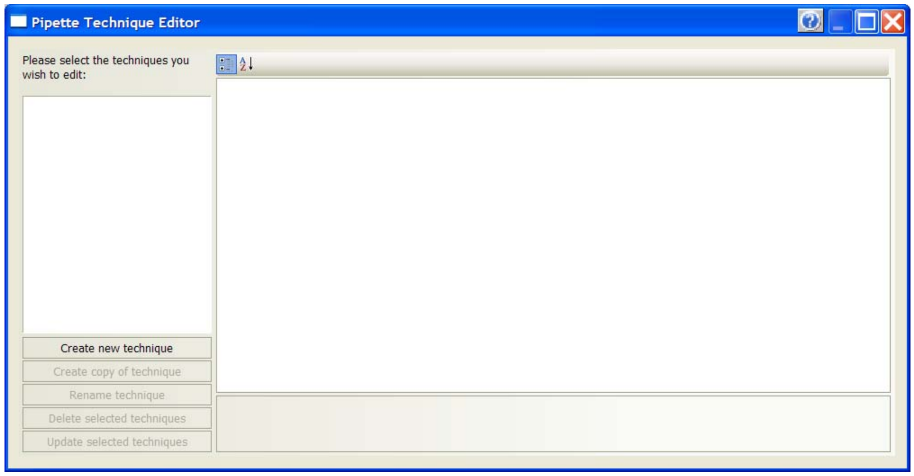
- When setting task parameters in a protocol
- Using the Tools menu

To create a Pipette Technique when setting task parameters:

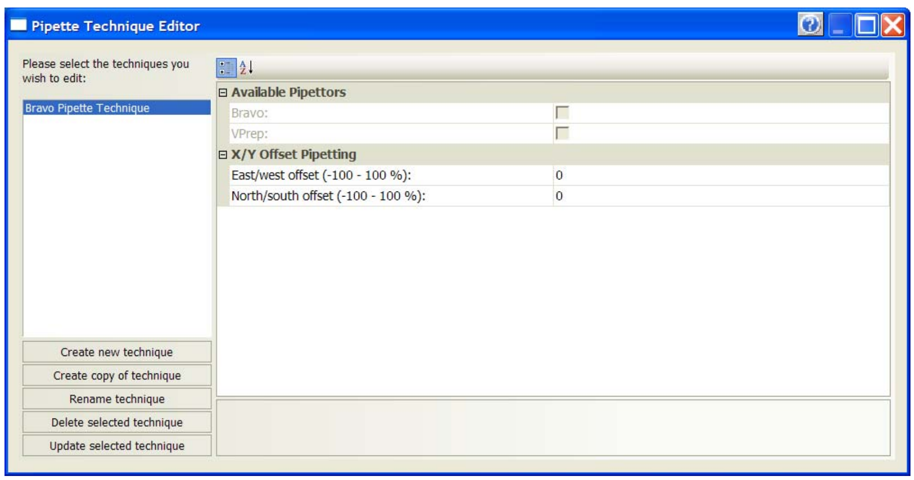
1. In the **VWorks** window, create a new protocol.
2. Add the **Subprocess (Bravo)** or **Subprocess (Vertical Pipetting Station)** task.
3. Add one of the following tasks:
   - Aspirate
   - Dispense
   - Mix
   - Pin Tool
4. In the **Task Parameters** area, select **Edit technique** in the **Pipette Technique property** list.



The Pipette Technique Editor dialog box opens.



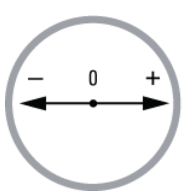
5 Click **Create new technique**, type a technique name, and then click **OK.** The new technique name appears in the Pipette Technique Editor dialog box. In addition, the **X/Y Offset** parameters appear to the right of the technique name.



6 In the **East/west offset (-100 - 100%)** box, type the distance (in percent of well radius) you want the pipette to move in the X direction:

- 0 does not move the pipette horizontally.
- Positive value moves the pipette to the right.
- Negative value moves the pipette to the left.

Note: The pipette offset directions do not correspond to the robot jog directions in Bravo and Vertical Pipetting Station Diagnostics.

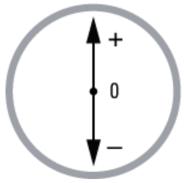


7 In the **North/south offset (-100 - 100%)** box, type the distance (in percent of well radius) you want the pipette to move in the Y direction.

- 0 does not move the pipette forward or backward.

- Positive value moves the pipette backward.
- Negative value moves the pipette forward.

Note: The pipette offset directions do not correspond to the robot jog directions in Bravo and Vertical Pipetting Station Diagnostics.



8 When you are finished, click **Update** the selected technique.

To create a Pipette Technique using the Tools menu:

1 In the **VWorks** window, on the **Tools** menu, click **Pipette Technique Editor.** The Pipette Technique Editor dialog box opens.

2 Click **Create new technique,** type a technique name, and then click **OK.** The new technique name appears in the Pipette Technique Editor dialog box. In addition, the X/Y Offset parameters appear to the right of the technique name.

3 Set the parameters in the **X/Y Offset Pipetting** table.

4 When you are finished, click **Update** the selected technique.

Editing a pipette technique

You can edit a Pipette Technique in two ways:

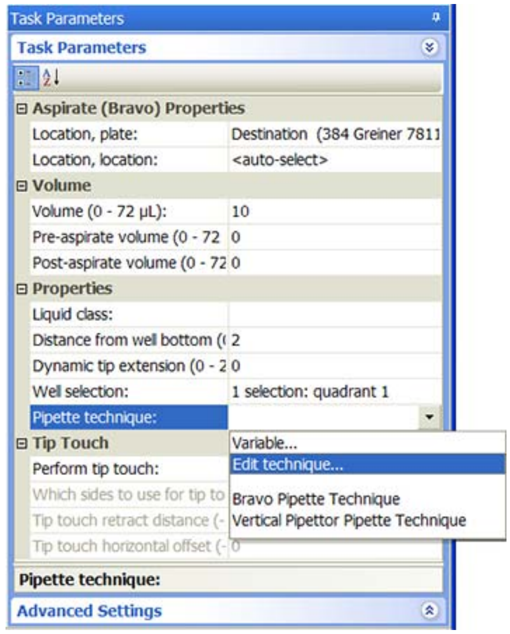
- When setting task parameters in a protocol
- Using the Tools menu

To edit a Pipette Technique when setting task parameters:

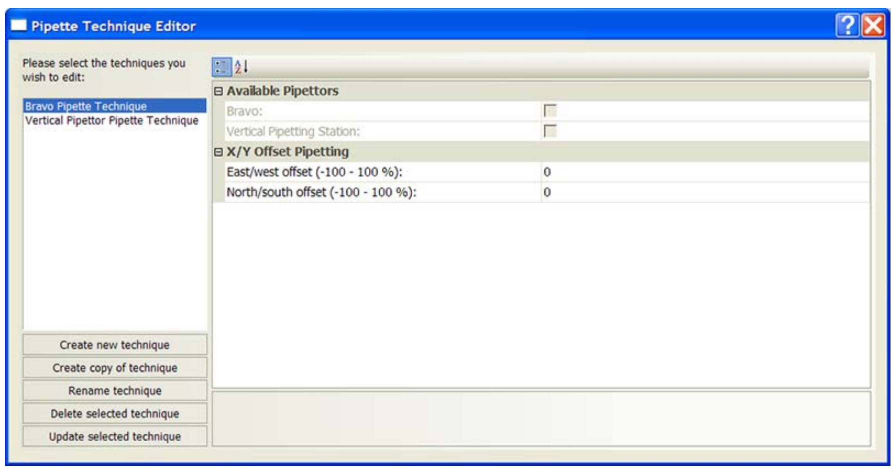
1 In the **VWorks** window, open the protocol.

2 In the **Main Protocol** area, select the task for which the Pipette Technique will change.

3 In the **Task Parameters** area, select **Edit technique** in the **Pipette Technique property** list.



The Pipette Technique Editor dialog box opens.



4 Select the **Pipette Technique** on the left side of the dialog box.

5 Make the desired changes in the **X/Y Offset Pipetting** table.

6 When you are finished, click **Update selected technique**.

To edit a Pipette Technique using the Tools menu:

1 In the **VWorks** window, on the **Tools** menu, click **Pipette Technique Editor.** The Pipette Technique Editor dialog box opens.

2 Select the **Pipette Technique** on the left side of the dialog box.

3 Make the desired changes in the **X/Y Offset Pipetting** table.

4 When you are finished, click **Update selected technique**.

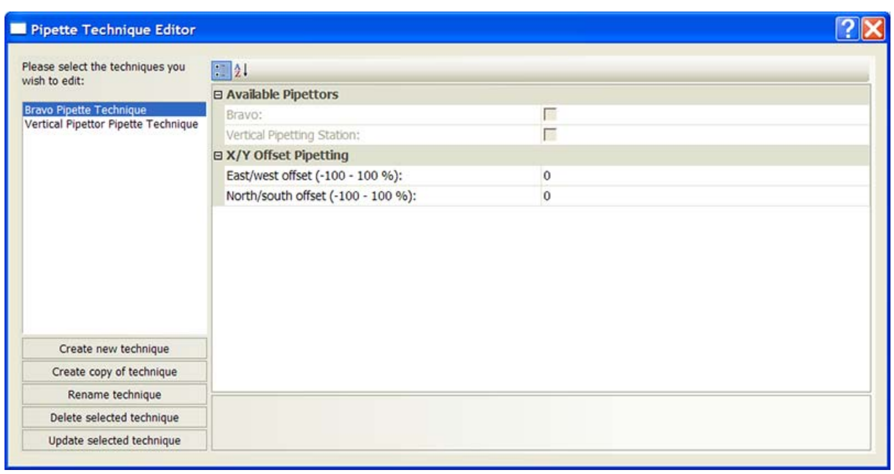
Related information

For information about...	See...
Pipette Techniques	"About the Pipette Technique Editor" on page 614
Managing Pipette Techniques	"Managing pipetting techniques" on page 619
Pipette Technique files	"Storing Pipette Technique files" on page 620
Aspirate task	"Aspirate (Bravo, Vertical Pipetting Station)" on page 392
Dispense task	"Dispense (Bravo, Vertical Pipetting Station)" on page 451
Mix task	"Mix (Bravo, Vertical Pipetting Station)" on page 497
Pin Tool task	"Pin Tool (Bravo, Vertical Pipetting Station)" on page 511

Managing pipetting techniques

To copy, rename, and delete an existing Pipette Technique:

1 In the **VWorks** window, on the **Tools** menu, click **Pipette Technique Editor.** The Pipette Technique Editor dialog box opens.



2 Select the Pipette Technique on the left side of the dialog box.

3 Click one of the following:

- *Create copy of technique*. The software prompts you to type a new name for the duplicated technique, and then creates a copy of the selected technique and saves it using the new name.

- *Rename technique*. The software prompts you to type a new name for the selected technique, and then saves the technique using the new name.
- *Delete selected technique*. The software deletes the selected technique.

Related information

For information about...	See...
Pipette Techniques	"About the Pipette Technique Editor" on page 614
Creating and editing Pipette Techniques	"Creating and editing pipetting techniques" on page 615
Pipette Technique files	"Storing Pipette Technique files" on page 620

Storing Pipette Technique files

Default storage location

By default, the VWorks software stores Pipette Technique files in the following folder:

...\VWorks Workspace\pipette techniques

Changing the storage location

To change the location of the Pipette Technique files:

1. In the **VWorks** window, choose **Tools** > **Options.** The Options dialog box opens.
2. In the **Options** area, click the **Pipette technique editor root** field, and then click the ... button that appears. The Browse for Folder dialog box opens.
3. Locate and select a folder for the Pipette Technique files.
4. Click **OK** to save the new location.

Related information

For information about...	See...
Pipette Techniques	"About the Pipette Technique Editor" on page 614

For information about...	See...
Creating and editing Pipette Techniques	"Creating and editing pipetting techniques" on page 615
Managing Pipette Techniques	"Managing pipetting techniques" on page 619

13
Maintenance and troubleshooting

This chapter contains the following topics:

Backing up and restoring files

Backing up files

You should regularly back up the following files in case they become damaged or lost:

- Protocols and associated files
- System files

You should store the backup files on a different computer or storage device.

Backing up protocols and associated files

You can back up files if you have administrator or technician privileges.

You can use the **Files > Export** command to back up protocols and associated files. For instructions, see "Exporting and importing protocols and associated components" on page 626.

Backing up system files

You use the Backup Manager to back up system files. The Backup Manager creates a copy of the following information and stores them in the .vbk file:

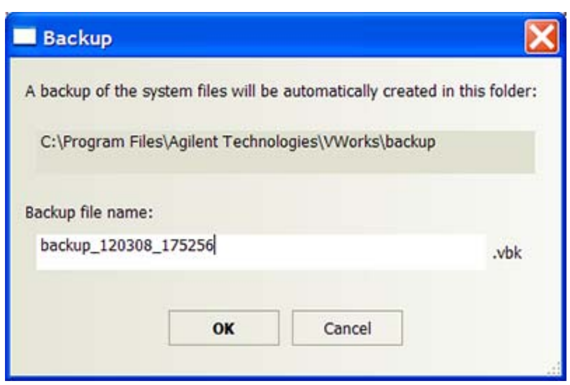
- Existing state of the inventory databases
- Labware definitions and liquid classes
- Pipette techniques

Make sure the inventory database is up-to-date before you back up the system files.

If you want to back up the inventory database without backup the entire set of system files, see *VWorks Automation Control Setup Guide*.

To back up system files:

1 In the **VWorks** window, select **Tools > Backup Manager > Backup.** The Backup dialog box opens.



2 Type a name for the backup file.

Note: The backup file location is the ...\VWorks\backup folder. You cannot change this location.

3 Click **OK.** The .vbk file is created and stored in the ...\VWorks\backup folder.

Restoring system files

Restoring protocols and associated files

You can restore files if you have administrator or technician privileges.

You can use the **Files > Import** command to restore protocols and associated files. For instructions, see "Exporting and importing protocols and associated components" on page 626.

Restoring system files

You use the Backup Manager to restore system files. The Backup Manager restores the following information in the databases and Windows registry:

- Existing state of the databases
- Labware definitions and liquid classes
- Pipette techniques

To restore system files:

1. In the **VWorks** window, select **Tools > Backup Manager > Restore.** The Open dialog box appears.
2. Locate and select the .vbk file, and then click **Open.** A message dialog box opens and asks you if you want to replace the existing labware definition, liquid classes, and pipette techniques.
3. Do one of the following:
   - Click **Yes** to overwrite existing labware definition, liquid classes, and pipette techniques.
   - Click **No** to cancel the restore process.

Related information

For information about...	See...
Exporting and importing protocols and associated files	"Exporting and importing protocols and associated components" on page 626
VWorks software components: protocol, device file, and so on	"Relationship of VWorks components" on page 4
Viewing logs	"Viewing logs" on page 631
Contacting Agilent Technologies and Reporting problems	"Reporting problems" on page 661

Exporting and importing protocols and associated components

Reasons for exporting and importing files

You can export and import protocols and associated components to transport protocols between computers, back up and recover protocols and associated files, and to facilitate troubleshooting problems with protocols and other files when seeking assistance from Automation Solutions Technical Support.

You can export or import the following:

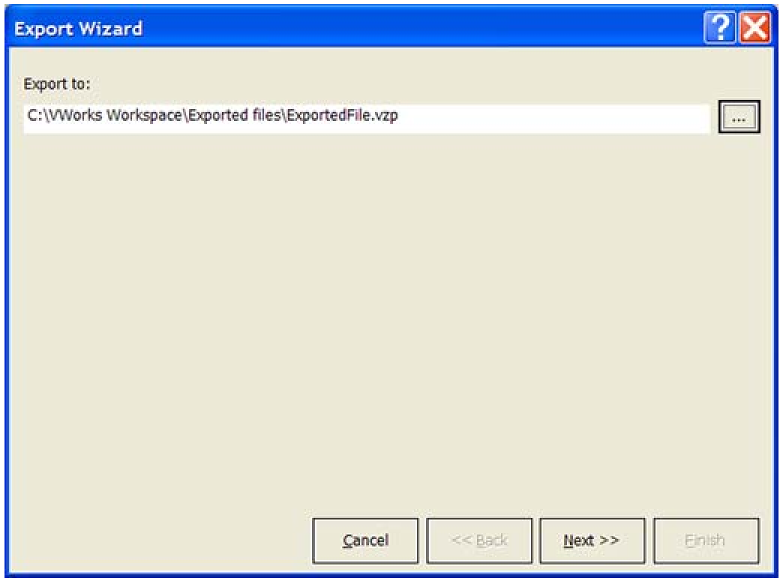
- Protocol file
- Form file
- Runset file
- Device file, including the device profile and teachpoint file
- Macros used in the exported protocol or the entire macro library (.mlb file)
- Labware definitions
- Liquid classes
- Pipette techniques
- Hit-picking files
- Plate map files
- Barcode files
- Error library
- Log files

Exporting files

You can export files if you have administrator, technician, or operator privileges.

To export the system files:

1 In the **VWorks** window, select **File > Export.** The Export Wizard dialog box opens.



2 Follow the instructions in the wizard to do the following:

a Specify the location and name of the .vzp file, which contains the exported and compressed files.

b Select the protocol, runset, or form you want to export. You do not need to select the associated device file and other files. The software automatically exports all the associated files with the protocol.

c *Optional*. If a form (.VWForm file) is specified in the protocol's options, verify the **Include all Form files specified in Protocol options** is selected (default) to include the form in the export. If you do not want to include the form, clear the **Include all Form files specified in Protocol options** check box.

IMPORTANT Any custom image files that you provide for the form, must be in the same folder as the .VWForm file that you are exporting.

d *Optional*. If the protocol is associated with barcode files, select **Export barcode files**.

e *Optional*. If the protocol is associated with an error library file, select **Export error library file**.

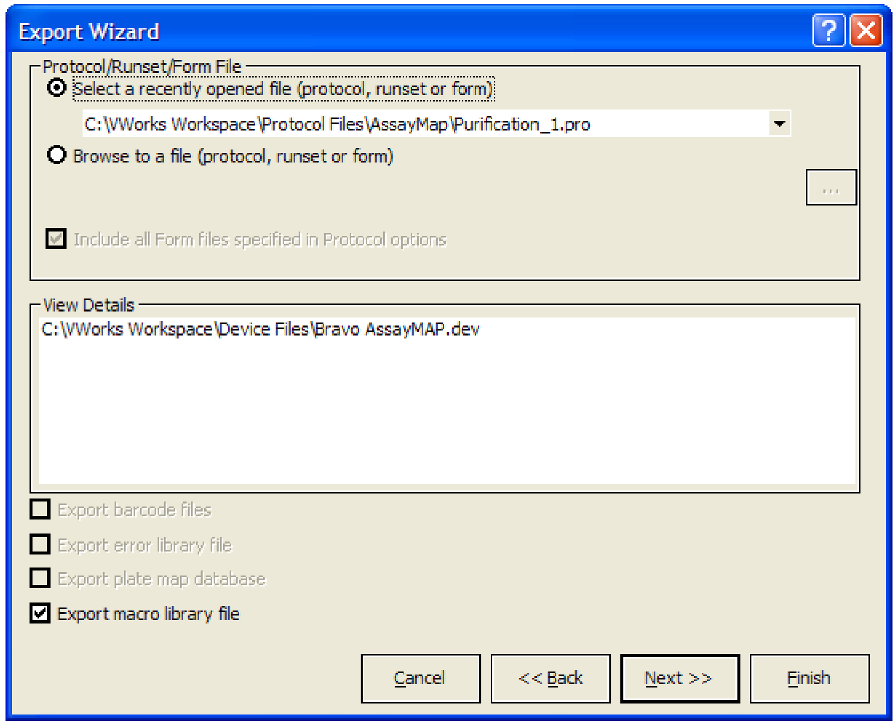
f *Optional*. If the protocol is associated with a plate map database, select **Export plate map database**.

g *Optional*. If the protocol contains task macros that have the same names as macros in the Available Macros area in the VWorks window (macro library), do one of the following:

- Select the **Export macro library file** check box to export the macro library information for those macros from the protocol that are also in the macro library. After the .vzp file is imported, this set of macros can be added to the macro library on the destination computer.
- Clear the **Export macro library file** check box to export the protocol version of the macros only. After the .vzp file is imported, the macros will appear in the protocol only.

Note: The Export macro library file check box is available only for the macros in the protocol that are also in the macro library. If you want to export the entire macro library, you can attach the .mlb file on the Additional files list page of the Export Wizard.

h Select the log files you want to include.



3 When you are finished, click **Finish.** The .vzp file appears in the specified location.

You can move the .vzp file to another computer and import it into the VWorks software, or send the .vzp file to Automation Solutions Technical Support to report a problem.

Importing files

You can import files if you have administrator or technician privileges.

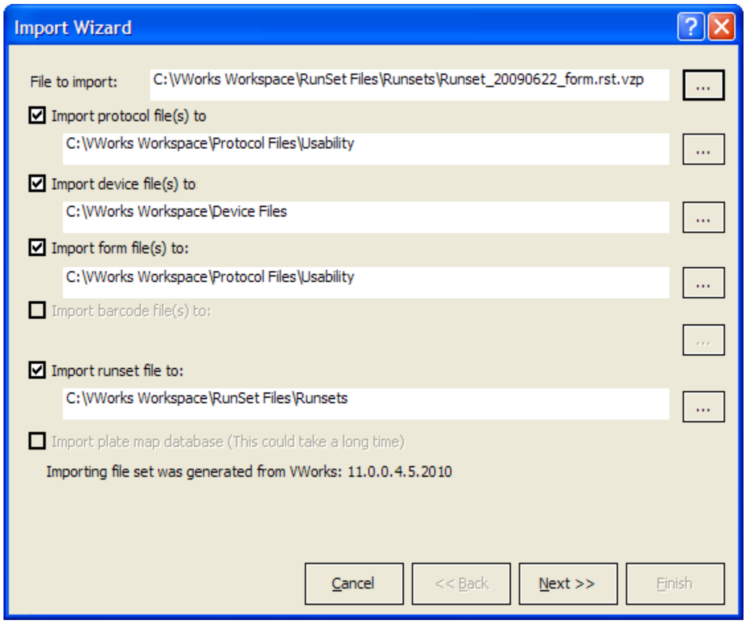
To import the system files:

1 In the **VWorks** window, select **File > Import.** A message dialog box opens and asks if you want to back up the existing labware definitions, liquid classes, pipette techniques, and plate-map databases.

2 Do one of the following:

- Click **Yes** to back up the information. In the **Backup** dialog box, type a name for the backup file (.vbk), and then click **OK.** The .vbk file is created and stored in the ...\VWorks\backup folder.
- Click **No** to start the import process.

The Import Wizard dialog box opens.



3 Follow the instructions in the wizard to:

a Select the .vzp file you want to import. The .vzp file contains the protocol and associated files.

b Select the location to store the imported protocol file, device file, barcode files, and runset file, if applicable.

c *Optional*. Select the option to import the associated plate-map database.

d Select the labware entries and classes you want to import. If a labware entry or class has the same name as an existing entry or class in the database, you have the following options:

- *Labware classes only*. Append new labware entries to the existing labware class.
- Replace existing labware entries or classes with newly imported entries and classes.

> **CAUTION** Protocols that rely on existing labware entries or classes might be affected.

- Create a new file for the newly imported labware entries and classes. You can choose to append the import date on the name of either the new labware file or the existing labware file. The imported protocol will use the imported labware definitions. When running existing protocols, you have to use the old labware file.

e Select the pipette technique files.

f Select the hit-picking input and format files.

g Select the device profiles.

h Select the error handlers and log files.

i If macros were associated with the exported protocol, review the list of exported macros. An asterisk highlights any macros that are already in the VWorks macro library on the destination computer.

You can choose to overwrite the macros of the same name or append the import date on the name of the imported macros.

4 When you are finished, click **Finish.** The files are imported in the specified locations. If you selected the option, the imported protocol opens in the VWorks window.

Related information

For information about...	See...
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661
VWorks software components: protocol, device file, and so on	"Relationship of VWorks components" on page 4
Viewing logs	"Viewing logs" on page 631
Backing up system files	"Backing up and restoring files" on page 624
Macros in protocols	"Using macros to create protocols" on page 133
Protocol forms	"Creating protocol forms for operators" on page 153

Viewing logs

About the logs

The VWorks software records events that occur and stores the information in the following logs:

- *Main Log*. Contains all of the actions that occur in the software. For more information, see "Main Log" on page 631.
- *Pipette Log*. Contains all pipetting events and error information. For more information, see "Pipette Log" on page 633.
- *Time Constraints Log*. Contains all information about time-limited tasks. For more information, see "Time Constraints Log" on page 634.

The log text colors can help you distinguish between different types of messages and events. For more information, see "Log text colors" on page 637.

In addition, you have the option of backing up and validating the log files for regulatory compliance. For more information, see "Backing up and validating the log files" on page 637.

Main Log

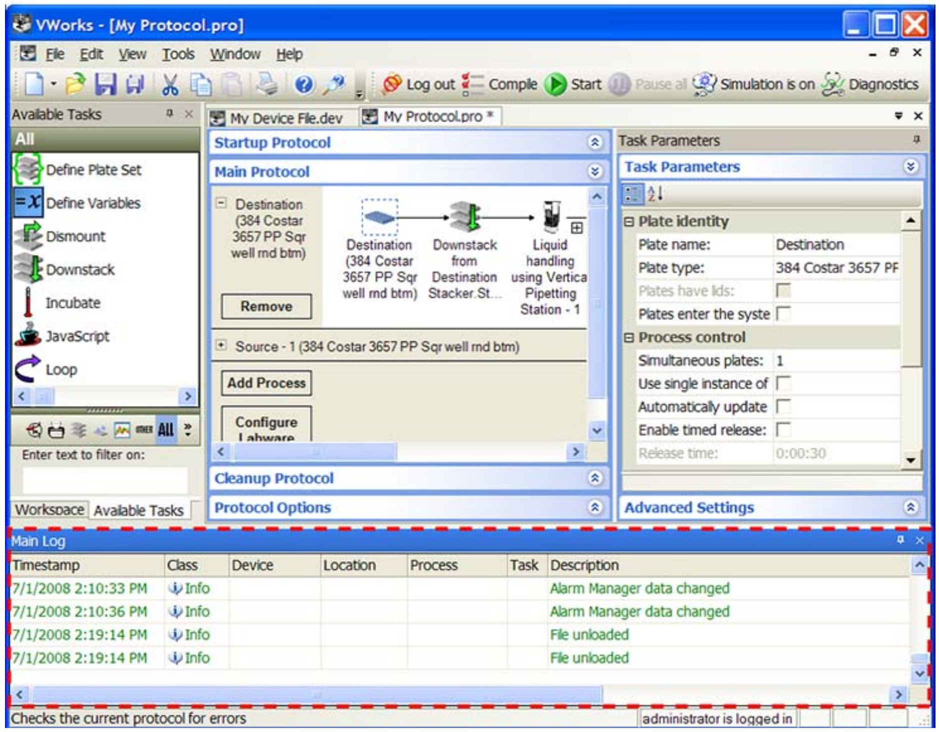
The Main Log records all available event and error information. You can view the Main Log in the following:

- VWorks window
- VWorks log file

Viewing the Main Log in the VWorks window

To view the Main Log in the VWorks window:

1. In the **VWorks** window, select **View > Main Log.**
2. Review the information in the Main Log tab.



Viewing the Main Log in a text editor

The VWorks log file is a text file with the following file name:

VWorks_log(*date_time*).log

To view the Main Log in a text editor:

1. In the **Windows Explorer** window, locate and select the VWorks_log(*date_time*).log file. The location is specified in the Options (Tools > Options) dialog box.
2. Open the file using a text editor such as Notepad.

Main Log contents

The Main Log contains the following information:

Column name	Description
Timestamp	Time and date of the event or error.
Class	The type of event or error message: • *Info*. General information. For example, System start up, or Simulation mode toggled on. • *Event*. A software action. For example, Scheduler started, or Move plate. • *Error*. An error that can stop the software and must be resolved. For example, Location incompatible with labware. • *Warning*. An error that might permit the software to continue. For example, Task requires that tips be on the pipette head.

Column name	Description
Device	Device at which the event or error occurred.
Location	Location where the event or error occurred. For example, Bravo location 9.
Process	Name of the protocol process and subprocess that is running.
Task	Task at which the event or error occurred.
Description	Description of the action that is being recorded.
File name	Name and location of the protocol file.
Session ID	Login session number.

Pipette Log

The Pipette Log records all pipetting events and error information. You can view the Pipette Log in the following:

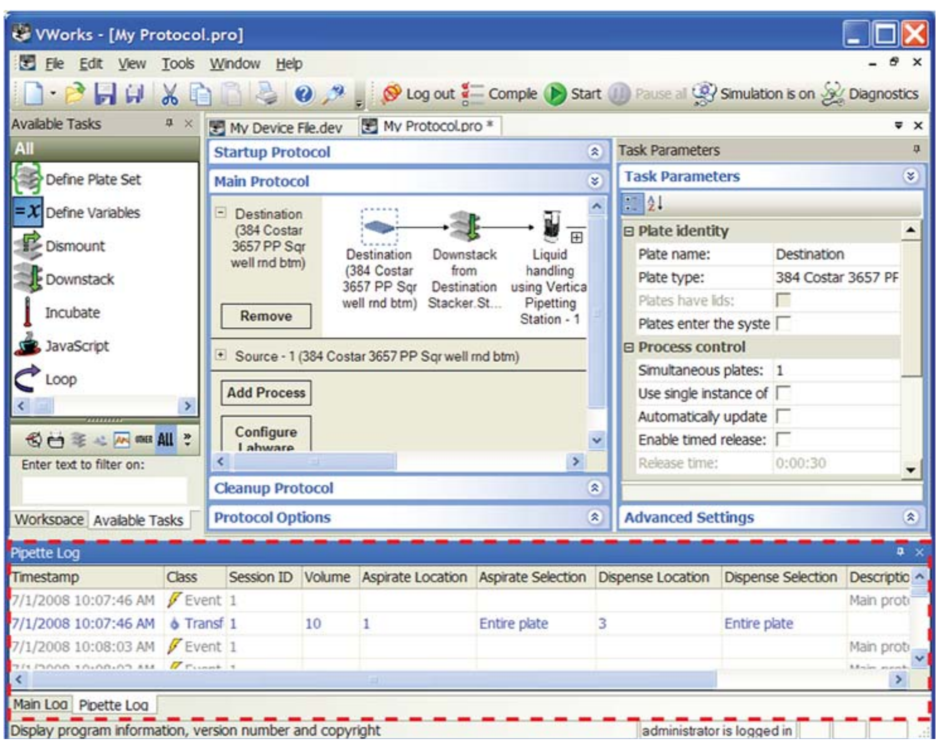
- VWorks window
- VWorks pipette log file

Viewing the Pipette Log in the VWorks window

To view the Pipette Log in the VWorks window:

1 In the **VWorks** window, select **View > Pipette Log.**

2 Review the information in the Pipette Log tab.



Viewing the Pipette Log in a text editor

The VWorks log file is a text file with the following file name:

VWorks_pipette_log(*date_time*).log

To view the Pipette Log in a text editor:

1 In the **Windows Explorer** window, locate and select the VWorks_pipette_log(*date_time*).log file. The location is specified in the Options (Tools > Options) dialog box.

2 Open the file using a text editor such as Notepad**.**

Pipette Log contents

The Pipette Log contains the following information:

Column name	Description
Timestamp	Time and date of the pipetting event or error.
Class	The type of event or error message: • *Event*. A software action. For example, when a protocol is started, the event is logged. • *Transfer*. A pipetting event.
Session ID	Login session number.
Volume	The volume of liquid transferred.
Aspirate Location	The location at which the Aspirate task occurred.
Aspirate Selection	The wells from which fluid was drawn for the Aspirate task.
Dispense Location	The location at which the Dispense task occurred.
Dispense Selection	The wells into which fluid was dispensed.
Description	The description of the action that is being recorded.
File name	The name and location of the protocol file.

Time Constraints Log

The Time Constraints Log records any time-limited task events and errors. If you did not specify time constraints, no information appears in the log.

You can view the Time Constraints Log in the following:

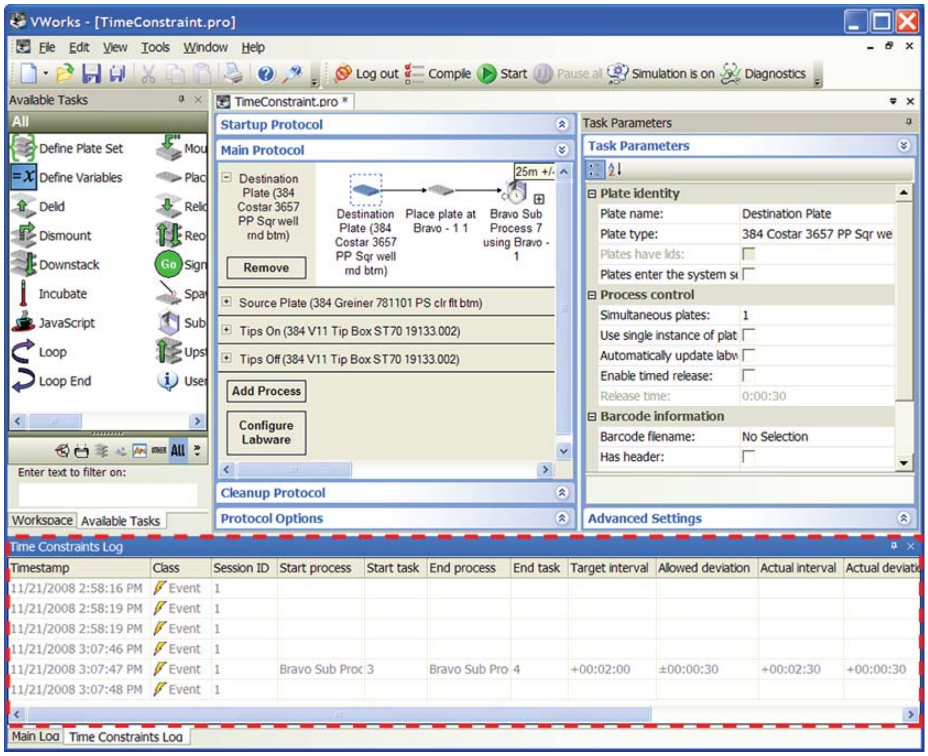
- VWorks window
- VWorks time constraint log file

Viewing the Time Constraints Log in the VWorks window

To view the Time Constraints Log in the VWorks window:

1 In the **VWorks** window, select **View > Time Constraints Log.**

2 Review the information in the **Time Constraints Log** tab.



Viewing the Time Constraints Log in a text editor

The VWorks log file is a text file with the following file name: VWorks_time_constraints_log(*date_time*).log

To view the Time Constraints Log in a text editor:

1. In the **Windows Explorer** window, locate and select the VWorks_time_constraints_log(*date_time*).log file. The location is specified in the Options (Tools > Options) dialog box.
2. Open the file using a text editor such as Notepad.

Time Constraints Log contents

The Time Constraints Log contains the following information:

Column name	Description
Timestamp	Time and date of the event or error.
Class	The type of event or error message: • *Info*. General information. For example, System start up, or Simulation mode toggled on. • *Event*. A software action. For example, Scheduler started, or Move plate. • *Error*. An error that can stop the software and must be resolved. For example, Location incompatible with labware. • *Warning*. An error that might permit the software to continue. For example, Task requires that tips be on the pipette head.

Column name	Description
Session ID	The login session number.
Start Process	The name of the protocol process or subprocess that contains the first of the two dependent tasks. This field also includes the instance number to help you identify the plate instance. For example, Process - 1 1 means Process - 1, instance 1. Start Process is paired with Start Task to identify the first of the two dependent tasks.
Start Task	The task number that identifies the first of the two dependent tasks. For example, 1 is the first task in the process, and 5 is the fifth task in the process. Start Task is paired with Start Process to identify the first of the two dependent tasks in the protocol.
End Process	The name of the protocol process or subprocess that contains the second of the two dependent tasks. This field also includes the instance number to help you identify the plate instance. For example, Process - 3 1 means Process - 3, instance 1. End Process is paired with End Task to identify the second of the two dependent tasks.
End Task	The task number that identifies the second of the two dependent tasks. For example, 3 is the third task in the process, and 10 is the tenth task in the process. End Task is paired with End Process to identify the second of the two dependent tasks in the protocol.
Target interval	The time specified in the Edit Time Constraints dialog box.
Allowed deviation	The time tolerance specified in the Edit Time Constraints dialog box.
Actual interval	The length of time of the first task during the run.
Actual deviation	The difference between the Target interval and the Actual interval.

Column name	Description
Error	The result of the following: |Actual deviation| – |Allowed deviation| If the result is less than 0, Error is set to 0. For example: • Actual deviation is 66. Allowed deviation is 60. So Error is 66 – 60, or 6. • Actual deviation is 30. Allowed deviation is 60. So Error is 30 – 60, or –30. Because the result is less than 0, Error is set to 0.
Description	The description of the action that is being recorded.
File name	The name and location of the protocol file.

Log text colors

Text in the logs appear in the following colors:

Color	Description
Grey	Standard events with a date stamp or operator-added notes
Blue	Liquid transfer events
Orange	Warnings
Red	Error
Green	General information
Purple	Debug information (shown only if the Debug log level is greater than 0 in Tools > Options)

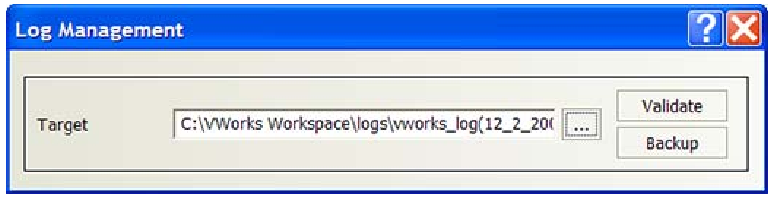
Backing up and validating the log files

You can back up and validate the log files to comply with regulatory requirements. The validation process checks the log file to see if it has been altered.

Backing up log files

To back up the log files:

1. With the protocol file open in the **VWorks** window, select **Tools > Log Management.** The Log Management dialog box opens.
2. Click the ... button, and then locate and select the log file you want to back up. The log file location is specified in Tools > Options. The file path and name appear in the **Target** box.



3 Click **Backup.** The Please name backup log file dialog box opens.

4 Select the folder in which you want to store the backup copy. In addition, type a name for the backup file.

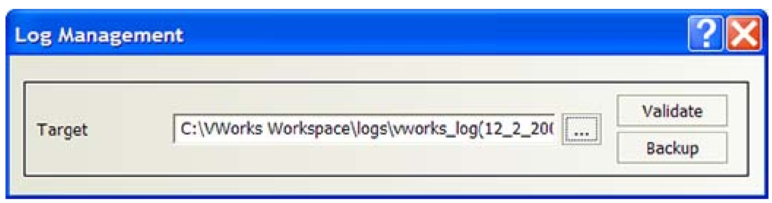
5 Click **Save.** A backup copy of the log file is created.

Validating log files

To validate a log file:

1 With the protocol file open in the VWorks window, select **Tools > Log Management.** The Log Management dialog box opens.

2 Click the ... button, and then locate and select the log file you want to validate. The log file location is specified in Tools > Options. The file path and name appear in the **Target** box.



3 Click **Validate.** The software checks the log file and displays a message that explains whether the validation is successful. Validation is successful only if the file has not been altered.

Related information

For information about...	See...
Setting system-wide options	“Setting general and view options” on page 224
Specifying log file storage locations	“Setting log file directories” on page 222
Exporting and importing protocol and associated files	“Exporting and importing protocols and associated components” on page 626
Contacting Agilent Technologies and reporting problems	“Reporting problems” on page 661

Resolving device initialization errors

Device initialization process

When you start the VWorks software, the software loads the driver files for all the devices in the automation system. A record of this process is displayed in the Main Log.

When you open a protocol, the device file associated with that protocol opens. The device file tells the software which devices are connected to the system. For some devices, an initialization step tests the communication between the VWorks software and the device.

Resolving initialization errors

During the device initialization process, the software displays the list of devices it is expecting to find. Devices are removed from the list as the software determines that the devices are ready.

If a problem occurs during initialization, an error message appears and explains the problem.

To resolve the problem:

1. Make sure the device is turned on.
2. Make sure the communication cable is connected properly.
3. Make sure the communication cable is connected to the correct COM port.
4. Check the device profile to make sure it is set up correctly for communication.
5. If applicable, follow the instructions in the error message to fix the communication problem.
6. Click **Retry** to re-initialize the device.
7. If the problem persists, contact Automation Solutions Technical Support.

Related information

For information about...	See...
Turning on the device	Device user guide
Editing the device profile	Device user guide
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661

Resolving compilation error messages

Compilation warnings and error messages

During the protocol-compiling process, the software reports errors in the Main Log. You can use the information to troubleshoot the protocol.

The software displays two types of messages during the compiling process:

- Warning messages
- Error messages

Note: If you are logged in with technician, operator, or guest privileges, you are unable to continue with the protocol. If you are logged in with administrator privileges, the dialog box allows you to run the protocol despite the errors.

Warning messages

A warning message alerts you to an error that should be fixed. If the error is left unresolved, the software can still continue. For example, a pipetting task requires that tips be on the pipette head, but there was no Tips On task preceding the pipetting task.

Error messages

Error messages alert you to situations where a protocol or device will fail. You must resolve the problem to continue the run.

Errors are generated when:

- Operating parameters are out of range, denoted by red text in the Task Parameters area.
- A task wants to use a labware that does not exist in the system.
- Volumes in pipette steps do not match, such as when a dispense volume is greater than a previous aspirate volume
- A Signal task has no associated Wait For task

Related information

For information about...	See...
Compiling protocols	"Compiling the protocol" on page 63
Viewing logs	"Viewing logs" on page 631
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661

Disabling and enabling tasks

When to disable tasks

You can disable tasks when you test run a protocol. Disabling one or more tasks allows you to:

- Run a protocol but skip tasks that use problem devices or locations. Doing so allows you to verify that the rest of the protocol is working properly while you troubleshoot the problem devices and locations.
- Skip tasks that are not needed for certain runs. You can reuse the same protocol for a number of situations without creating a number of different protocols.

Note: You can also skip tasks by disabling the device assigned to the task in the device file. This allows you to run alternative protocol configurations using a single device file. See "Disabling and enabling a device in the device file" on page 215 for more information.

Disabling tasks

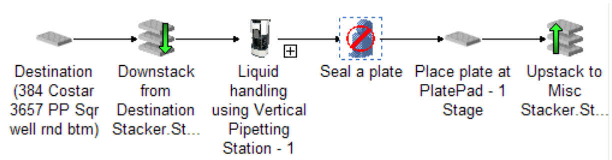
To disable a task:

1 Right-click the task that you want to disable.

2 In the shortcut menu that appears, select **Disable task.**

A red circle and strike line appears on the icon to indicate that it is a disabled task. The software will bypass the task when the protocol is running.

In the following example, the Seal a Plate task is disabled in the protocol.



Enabling tasks

You can enable tasks individually or you can enable all tasks using one command.

To enable a task:

1 Right-click the disabled task.

2 In the shortcut menu that appears, select **Enable task.**

To enable all tasks in a protocol:

1 Right-click any task in the protocol.

2 In the shortcut menu that appears, select **Enable all tasks.**

Related information

For information about...	See...
The Halt on barcode misreads option in the Options dialog box	"Setting general and view options" on page 224
Viewing logs	"Viewing logs" on page 631
Running protocols in simulation mode	"Simulating the protocol run" on page 64

Using breakpoints to monitor and troubleshoot tasks

About breakpoints

A breakpoint is a point in the protocol where you want the system to pause operation. Adding a breakpoint at a task pauses the run or simulation and opens the Debugger dialog box before the system performs the actual task. The Debugger dialog box provides JavaScript information for the corresponding task.

Breakpoints enable you to do the following:

- You can monitor parameter values and verify the actions of certain tasks during simulation or a dry run, and change the values if necessary.

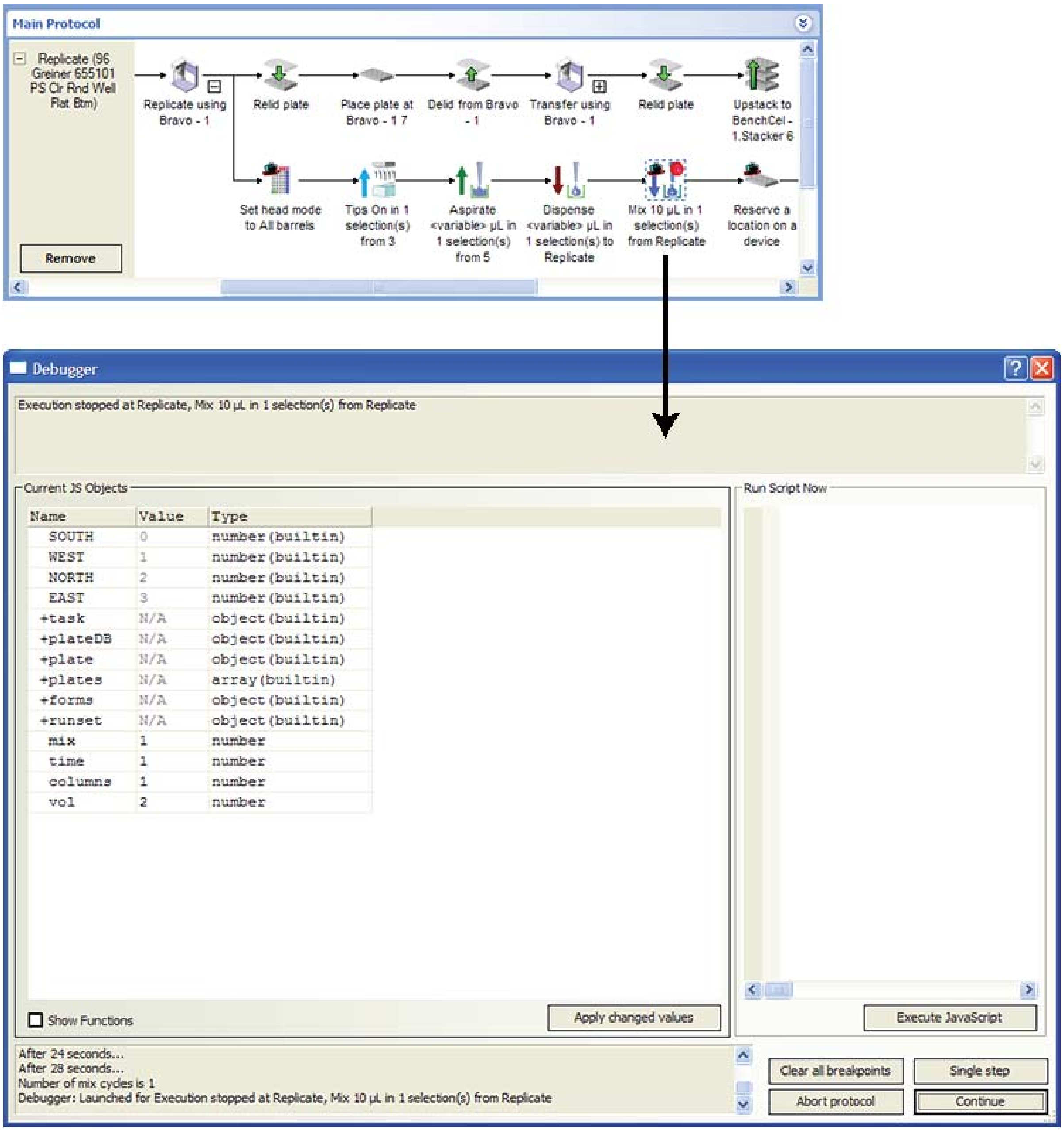
  For example, you can add a breakpoint at an aspirate task to pause the system just before the aspirate task starts. During the pause, you can verify the task parameter values in the JavaScript engine, after the task's script executes, and before the parameters are applied to the task.
- If you declare your own JavaScript variables for certain tasks, you can monitor the values by placing breakpoints at those tasks.
- You can troubleshoot tasks and fine-tune task parameter values.
- The Debugger dialog box also enables you to write and execute scripts in real time that are not necessarily associated with the selected task.

During the protocol run, the following sequence occurs when the software reaches a breakpoint:

Step	Description
1	The protocol pauses at the task with the breakpoint.
2	The JavaScript associated with the selected task executes.
3	The Debugger dialog box opens and displays the JavaScript for that task.
4	In the Debugger dialog box, you can modify values in the JavaScript engine.
5	The software applies the relevant task parameters from the JavaScript engine to the scheduler. If you changed task parameter values in the Debugger dialog box, the task will use the updated values when you continue the run.
6	You can continue running the selected task and the rest of the protocol or abort the run.

Note: If the breakpoint is on a task within a loop or if the protocol is run multiple times in one session, the protocol will pause each time it reaches that task.

Figure A breakpoint in a protocol and the corresponding Debugger dialog box

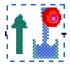


Adding breakpoints

You can add as many breakpoints as you want in a protocol. In addition, you can add breakpoints while a protocol or simulation is running.

To add a breakpoint:

1. In the **Protocol** area, right-click the task at which you want the breakpoint.
2. In the shortcut menu that appears, click **Set breakpoint.** A red dot appears on the task icon.



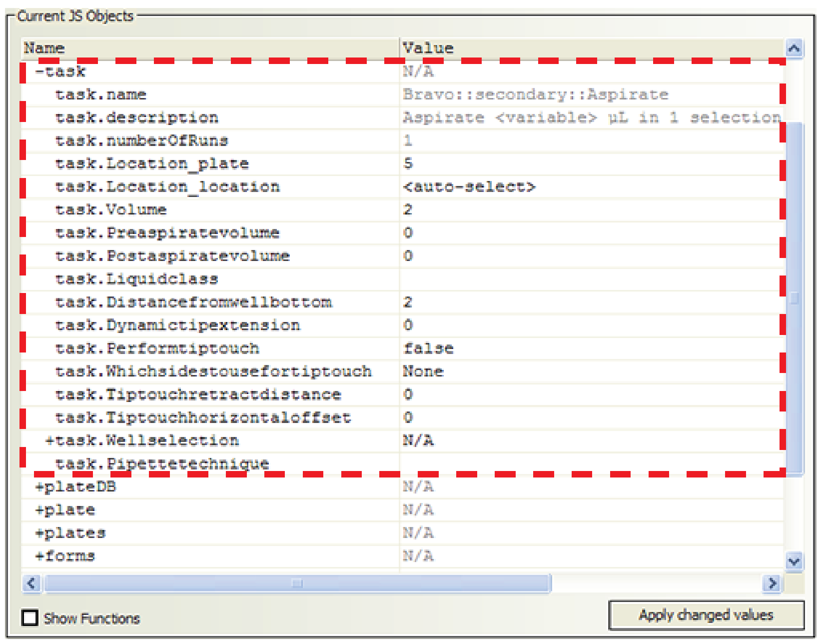
Using a breakpoint during a simulation or a dry run

To use a breakpoint in a protocol:

1 Start a simulation or dry run of the protocol that contains the breakpoint. When the software reaches the task that has the breakpoint, the Debugger dialog box opens.

2 Under **Current JS Objects** in the Debugger dialog box, look in the **Name** column for the object that you want to view. Click the **+** symbol next to the object name to view the corresponding properties.

For example, the following figure shows the task object properties for an aspirate task at which a breakpoint was set.



Note: To show the functions (methods) as well as the properties associated with the JavaScript object, select **Show Functions**, and then expand the object in the **Name** column.

3 In the **Value** column, change a property value, if required. For example, you might change the `task.Volume` value from 2 to 3.

4 To apply any changed values without closing the Debugger dialog box, click **Apply changed values**.

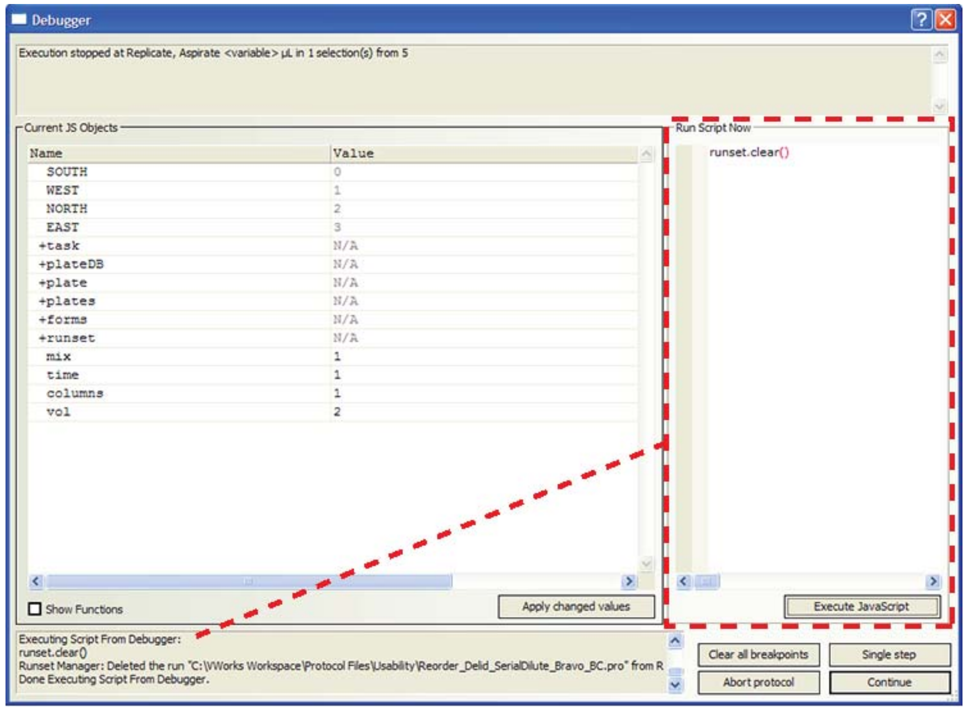
5 If you want to run a script from the Debugger dialog box:

a Type the script in the **Run Script Now** area.

b Click **Execute JavaScript**.

The status of the script appears in the bottom left of the Debugger dialog box.

For example, you could type the script `runset.clear()to` clear all the entries in the runset manager, except for currently running protocol.



6 Click the button in the Debugger dialog box that corresponds to the command you want to use next:

Button	Description
Clear all breakpoints	Removes all breakpoints in the protocol and continues the run.
Single step	Permits the protocol to perform one task at a time, pausing before each task.
Continue	Resumes the protocol run until the next breakpoint is reached.
Abort	Aborts the protocol run.

Removing breakpoints from an open protocol

You can remove breakpoints individually or you can remove all breakpoints with one command.

To remove a specific breakpoint:

1 Right-click the task that has a breakpoint.

2 In the menu that appears, click **Remove breakpoint.**

To remove all breakpoints in a protocol:

1 Right-click any task in the protocol.

2 In the menu that appears, click **Clear all breakpoints.**

Related information

For information about...	See...
Compiling protocols	"Compiling the protocol" on page 63
Simulating a protocol run	"Simulating the protocol run" on page 64
JavaScript objects in the VWorks software	"Creating a protocol: advanced topics" on page 73
Tracking the run progress of instances, processes, or devices	"Tracking the run progress of instances or devices" on page 255
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661

Resolving barcode reader error messages

About this topic

The information in this topic applies only to the following barcode readers:

- Robot barcode readers
- Barcode readers that are installed on a device such as a platepad

The information in this topic does not apply to the Microplate Labeler.

Causes of barcode reader errors

Errors generated by robot barcode readers and barcode readers that are installed on devices are generally caused by:

- Poor printing or label placement such that the barcodes are not perpendicular to the barcode laser read line
- Poor label placement such that the barcode cannot be fully read
- Missing or damaged barcode labels
- Wrong barcode labels

Note: The system does not attempt to read a barcode unless you specified that the labware has a barcode label. See "Setting plate parameters" on page 46 for instructions.

Recovering from barcode reader errors

Your ability to resolve the barcode reader errors depends on whether the **Halt on bar code misreads** option is selected in the Options (Tools > Options) dialog box.

If the **Halt on bar code misreads** option is selected:

- The protocol pauses.

- The error message is recorded in the Main Log.
- The error message dialog box opens and allows you to type the correct barcode. With the correct barcode, the software continues the run.

If the **Halt on bar code misreads** option is not selected:

- The error message is recorded in the Main Log.
- The protocol continues without pausing, so you cannot correct from the error.

A quarantine response is set up in the Error Library for a default set of barcode reader error messages. The quarantine response allows the system to continue running the protocol even though it is unable to resolve problems with the labware. For a description of the quarantine response, see "Setting up automated responses" on page 655.

Related information

For information about...	See...
The Halt on bar code misreads option in the Options dialog box	"Setting general and view options" on page 224
Viewing logs	"Viewing logs" on page 631
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661

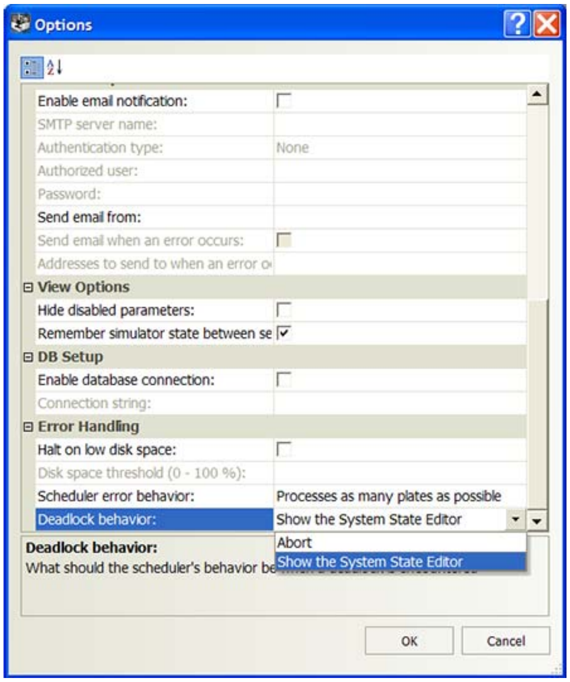
Recovering from deadlocks

About deadlocks

A deadlock is an error that occurs when the number of locations available in the system is less than the number of microplates in the system. Because the microplates cannot move to the expected locations, the protocol either aborts or pauses, depending on the **Deadlock behavior** selection in the Tools > Options dialog box.

About the System State Editor

In the Tools > Options dialog box, if the **Deadlock behavior** is **Show the System State Editor,** the run will pause when a deadlock error occurs.



During the pause:

- *Systems with an Automation Control Unit*. The Scheduler Paused dialog box appears. To access the System State Editor, click **Bypass interlock**, and then click **System State Editor** in the Bypass Interlock dialog box.
- *Systems that do not have an Automation Control Unit*. The System State Editor dialog box appears.

In the System State Editor dialog box, you can:

1. View the status of the devices and locations to assess and correct the causes of the errors. For example, if the device status indicates that a labware at a particular location is causing device problems, you can physically remove that labware from the system and restore the device to its operational state.
2. Specify in the software that a labware has been removed from certain locations or from the system. For example, after you physically remove the labware that is causing problems, you need to indicate in the software that the labware is removed and its location is now ready to accept new labware.
3. Edit or reset the status of devices and locations. If you fixed a device problem, you can indicate that the device is now operational and ready to accept labware.
4. Continue the protocol run.

If you determine that the causes of the deadlock error cannot be fixed, you can choose to abort the run or allow the existing labware to finish processing but not introduce new labware. For more information about these options, see "Pausing the run" on page 261.

You can also open the System State Editor when you pause all runs. For instructions, see "Pausing the run" on page 261.

Viewing the device and location status

To view the device and location status:

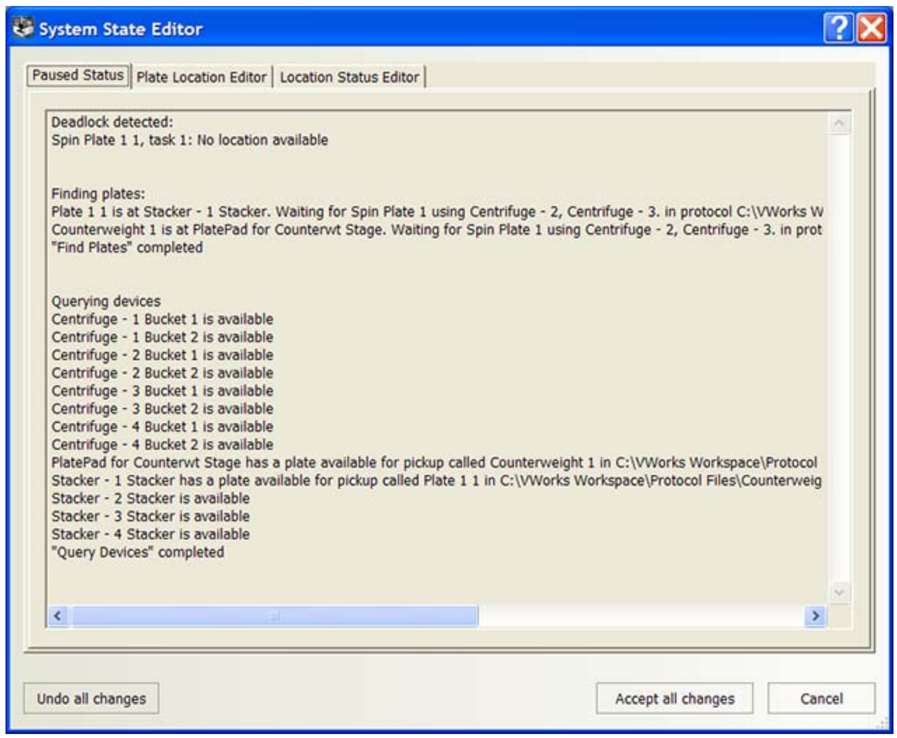
1 In the **System State Editor** dialog box, click the **Paused Status** tab.

2 Review the information in the **Paused Status** tab.

The Paused Status tab contains the following information:

- Causes of the deadlock error
- Current locations of the labware in the system and the expected locations when the deadlock error occurred
- The current status of all the devices and locations in the system

Note: The information in the Paused Status tab can be appended to the Main Log file. To do this, select **Tools > Get Status.**



Moving or removing labware from locations

The Plate Location Editor tab lists:

- All active process plates in all running protocols
- The current locations of the process plates

IMPORTANT The list does not include finished process plates. To see the finished process plates, click the Location Status Editor tab.

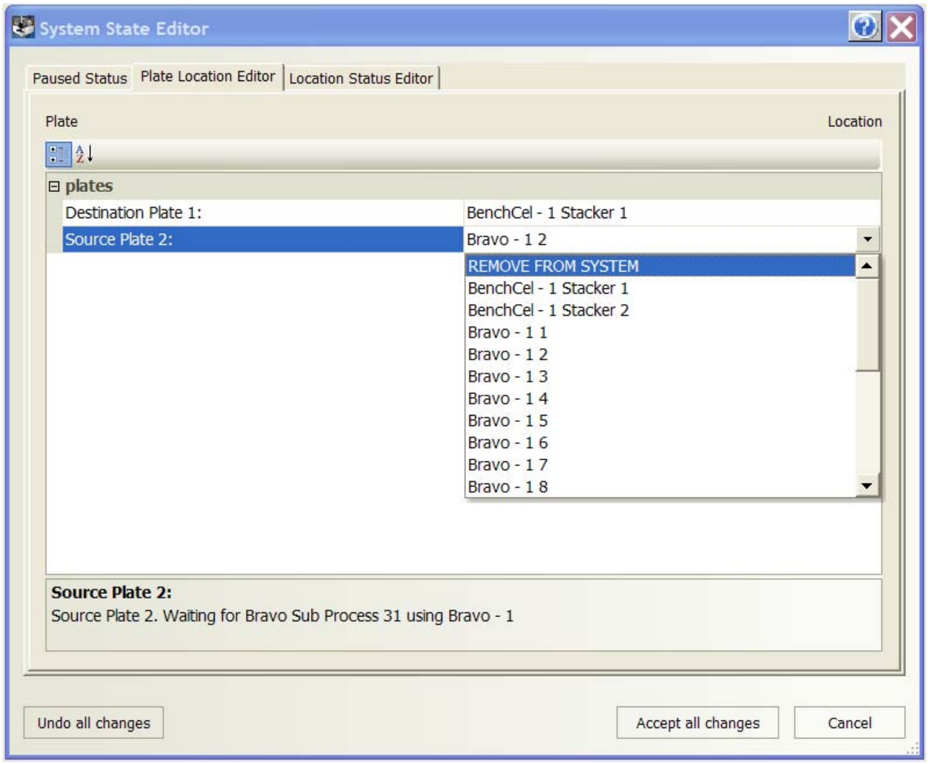
IMPORTANT You can move labware to any location except into a Labware Stacker.

To indicate that labware is moved or removed from locations:

1 In the **System State Editor** dialog box, click the **Plate Location Editor** tab.

2 To change the location of a process plate, select the new location from the list. If you want to indicate that a process plate was removed from the system, select **REMOVE FROM SYSTEM**.

Note: The software prevents you from selecting one location for two different process plates. If you select a location that is already occupied by another process plate, the software will exchange the two locations.



Editing the status of the devices and locations

CAUTION Do not change the status of device locations without fully knowing the state of all process plates and device locations. An arbitrary change can cause crashes when you resume the run. If you are uncertain of the state of the process plates and device locations, contact Automation Solutions Technical Support.

The Location Status Editor tab displays all labware in running protocols in:

- The list of every device and associated locations
- The status of the locations

Finished process plates appear in this tab. You can use the selections in this tab to remove finished process plates.

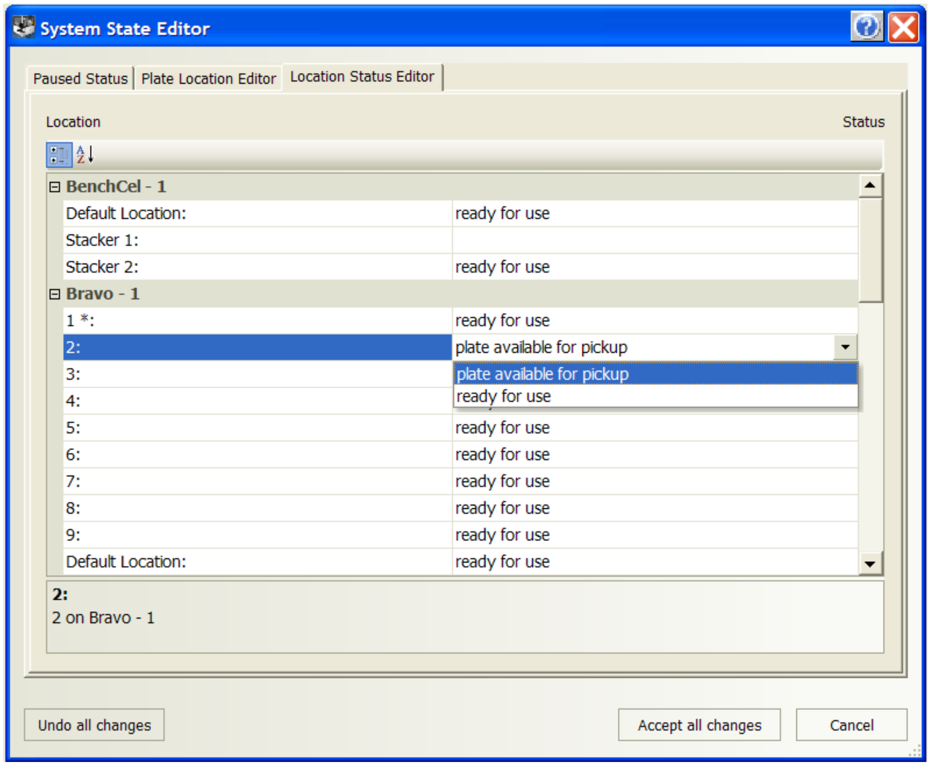
CAUTION Make sure finished process plates will not be used by downstream processes before removing them.

Except to remove finished process plates, you do not need to change the status of device locations. The changes you made in the Plate Location Editor tab are automatically reflected in the Location Status Editor tab. For example, if you removed a process plate from a location in the Plate Location Editor tab, the status of that location will change from Plate available for pickup (location is occupied) to Ready for use (location is available for use).

To remove a finished process plate:

1 In the **System State Editor** dialog box, click the **Location Status Editor** tab.

2 Find the finished process plate location, and then select the new status from the list.



Status	Description
Plate available for pickup	Retains the finished process plate in the system.
Ready for use	Removes the finished process plate from the system.

Continuing the protocol run

After you specify the labware movement or removal and the new device and location status:

1 In the **System State Editor** dialog box, click **Accept all changes.**

2 *Systems with an Automation Control Unit*. In the **Bypass Interlock** dialog box, click **Resume Run**.

3 In the **Scheduler Paused** dialog box, click **Continue** to resume the run.

The Scheduler Paused dialog box also provides other options. For details, see "Pausing the run" on page 261.

Related information

For information about...	See...
Pausing a run	"Pausing the run" on page 261
Monitoring a run	"Tracking the run progress of instances or devices" on page 255
Viewing logs	"Viewing logs" on page 631
Tracking the run progress of instances, processes, or devices	"Tracking the run progress of instances or devices" on page 255
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661

Setting up automated error responses

About the Error Library

You can use the Error Library to:

- Display or hide certain error messages during a protocol run. Hiding minor errors reduces the number of interruptions in the protocol run.
- Automate error recovery responses for selected errors to reduce the number of manual interventions in a protocol run.
- Track the frequency of certain errors for troubleshooting, establishing standard automated recovery responses, or quality-control reporting.

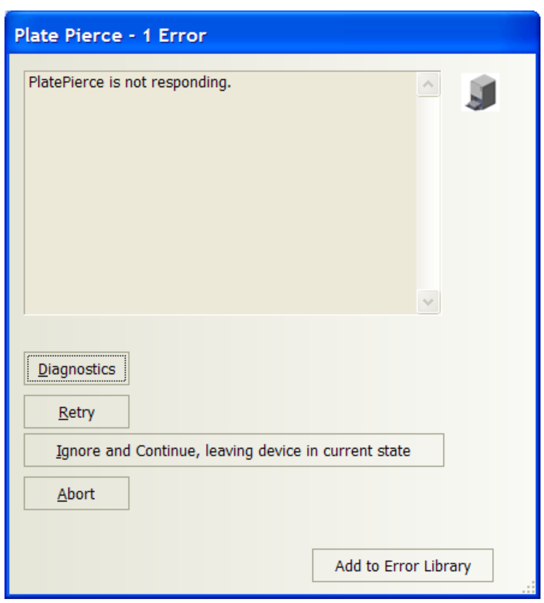
For example, you can set up a quarantine response for barcode reading, incorrect labware orientation, and wrong labware type errors. The quarantine response allows the system to continue running the protocol even though it is unable to resolve problems with the labware.

Adding errors to the library

You can add error messages to the Error Library to set up display options and automated responses. When you run a protocol and an error dialog box opens, click **Add to Error Library** to add the displayed error message to the Error Library.

To set up quarantine responses, you can perform dry runs to generate the desired error messages for the Error Library. For example, you can use labware that have poorly printed or missing barcode labels, load labware backwards in stackers, or load the wrong labware type in stackers.

Note: The software automatically records the frequency of the error in the Error Library.

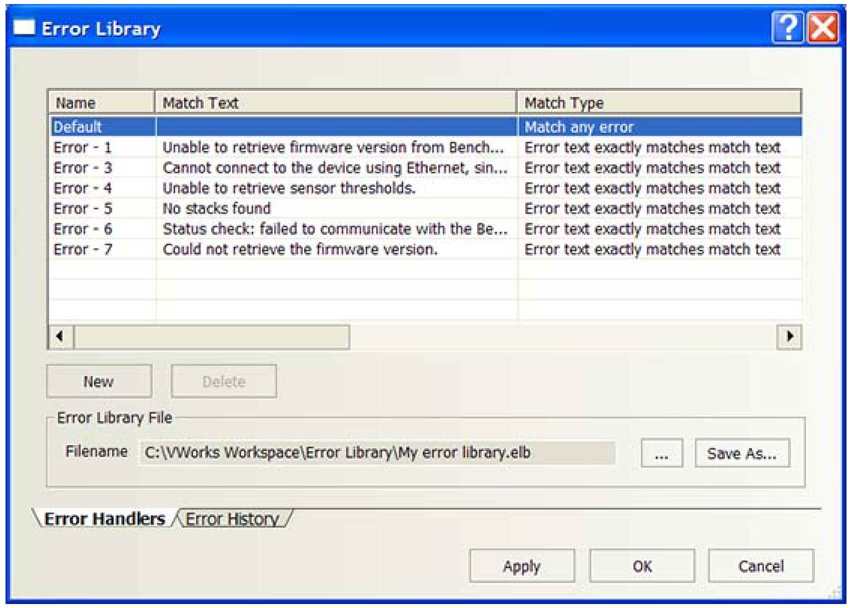


Setting up automated responses

To set up error responses:

1 In the **VWorks** window, select **Tools > Error Library.** The Error Library dialog box opens.

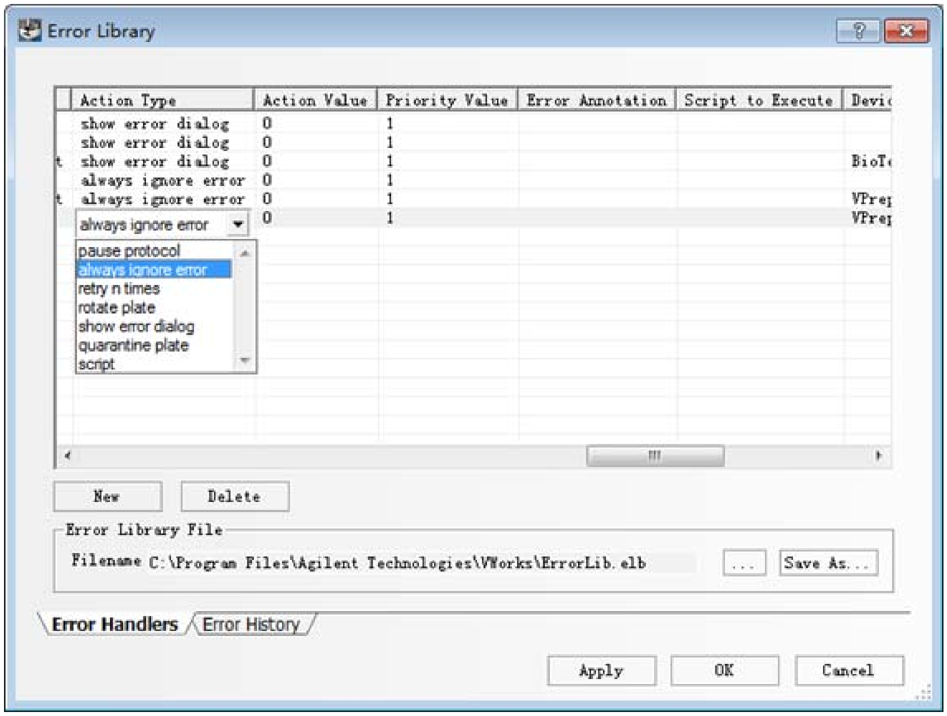
Note: The software opens the last saved error library file shown in the Error Library File area. To open a different file, click the ... button.



2 In the **Error Handlers** tab, set the following parameters to identify the error handlers. An error handler is a set of conditions that define a specific recovery response to an error.

Parameter	Description
Name	The name of the error handler. Type a name that can help you to identify it quickly in the log. *Note:* The Default error handler is used to define the default error response. It can be edited, but it cannot be deleted from the Error Library.
Match Text	The text of the error message that must be matched to activate the error handler. You can modify the text to set it up for match-text filtering.
Match Type	The match-text filtering to use: • Error text exactly matches Match Text • Error text contains Match Text

3 Set the following to determine the error response:



Parameter	Description
Action Type	The automated error response to use whenever the text-matching conditions are met: • *Pause protocol*. Pauses the protocol. • *Always ignore error*. Ignores the current command or task and continues to the next command or task in the protocol sequence. • *Retry n times*. Attempts to restart the current command or task in the run. The number of attempts is specified by the Action Value. • *Rotate plate*. Places the labware at a device that can rotate it to correct its orientation. • *Show error dialog*. Displays the error dialog box to allow the operator to determine the error response. • *Quarantine plate*. Moves the labware that caused the error to a quarantine location and continues the run. • *Script*. Runs the script specified in the **Script to Execute** box. See step 6.
Action Value	The number of times to restart the current command or task.

4 Type the **Priority** value in case the text from two errors matches the conditions in two handlers. The larger the value, the lower the priority.

The software sets the overall priority as follows:

a Exact error text match

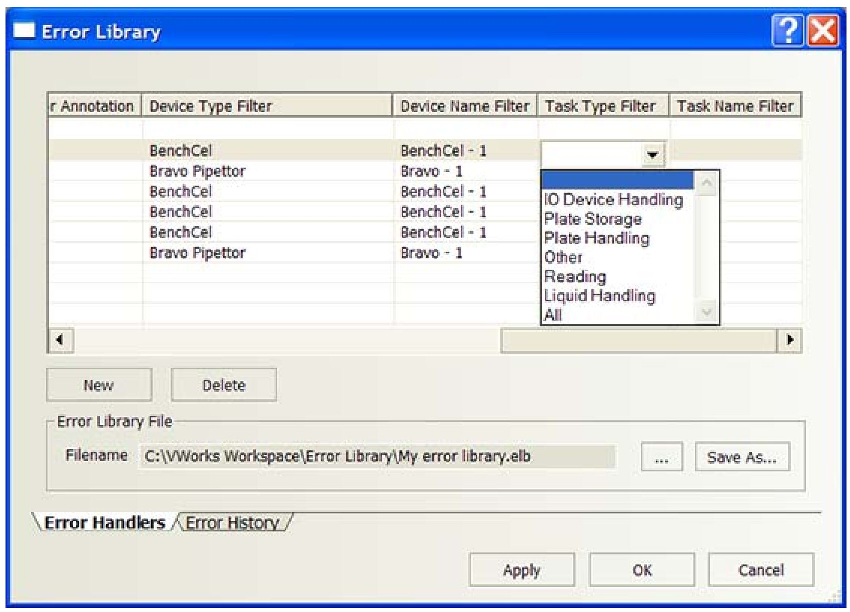
b Partial error text match (longer text matches are preferred over shorter text matches)

c Larger Priority value (for example, 1 has higher priority than 2)

5 Type additional notes about the error in the **Error Annotation** box. The text in this box is displayed with the original error text. You can use this field to explain how to fix the error.

6 Type the script you want to run in the **Script to Execute** box. This box is only active if you have selected **Script** for the **Action Type** in step 3.

7 Set up additional filters for the error message. You can select the desired value for each of the parameters in the following table. If you do not want to set up a filter for one or more of the parameters, select the empty value from the parameter list.

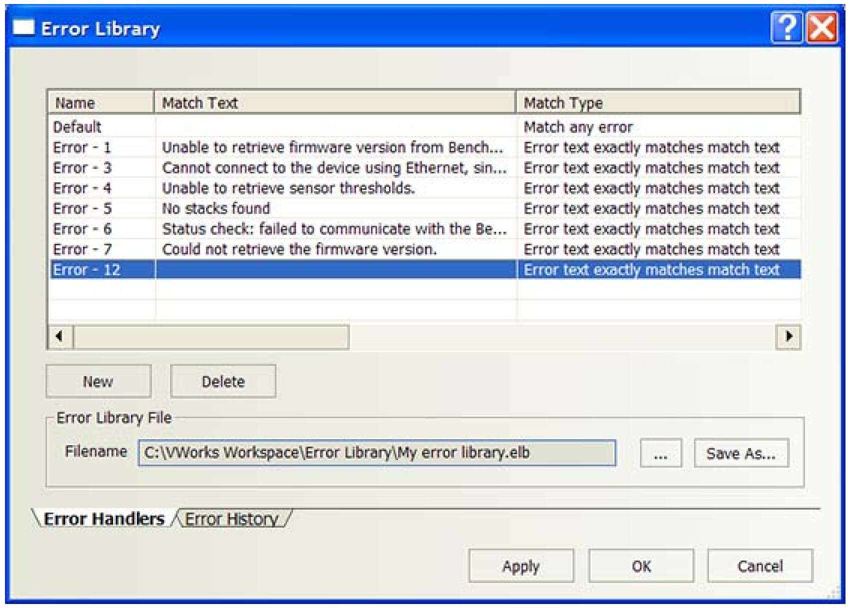


Parameter	Description
Device Type Filter	The device type, such as Bravo Pipettor.
Device Name Filter	The device name that distinguishes two devices of the same type. For example, Bravo-1 and Bravo-2.
Task Type Filter	The task category. For example, Plate Handling tasks.
Task Name Filter	The name of the task. For example, Aspirate.

Creating a new error handler

To create a new error handler:

1 In the **Error Library** dialog box, click **New.** A new row appears in the error handler table with a default name, Error - n.



2 Follow the instructions in "Setting up automated responses" on page 655 to set up the new error handler.

Note: In a newly created handler, the Match Text field is empty. Type the error text that you want to use to set up the handler.

Deleting an error handler

To delete an error handler:

1 In the **Error Library** dialog box, select the error handler you want to delete.

2 Click **Delete.** The error handler is removed from the table.

Saving the error library file

To save the error library file:

1 In the **Error Library File** area, click **Save As.** The Save As dialog box opens.

2 Locate the folder in which you want to save the file, type a name for the error library file, and then click **Save.**

Viewing and tracking error occurrences

You can view all the errors that have occurred during the protocol run that have not yet been assigned an error handler. The Error Library records the frequency of their occurrences and other information to help you determine whether you want to add them to the library and what automated response to specify.

To view errors that are not yet set up with error handlers:

1 In the **Error Library** dialog box, click the **Error History** tab.

2 Review the list of errors in the table:

Parameter	Description
Error Message	The text of the error message.
Device Type	The type of device on which the error occurred. For example, Bravo Pipettor.
Device Name	The name of the device on which the error occurred. The name distinguishes two devices of the same type. For example, Bravo-1 and Bravo-2.
Task Type	The category of the task in which the error occurred. For example, Plate Handling tasks.
Task Name	The task in which the error occurred. For example, Aspirate.
Error Times	The number of times the error occurred within the protocol.
File Name	The name of the device file containing the device.

To add an error message to the error library file:

1. Select the error message in the table.
2. Click **Add to Error Library.**
3. Follow the instructions in "Setting up automated responses" on page 655.

To clear the history table:

Click **Clear All.**

Related information

For information about...	See...
Viewing logs	"Viewing logs" on page 631
Tracking the run progress of instances, processes, or devices	"Tracking the run progress of instances or devices" on page 255
Contacting Agilent Technologies and reporting problems	"Reporting problems" on page 661

Viewing the status of the UPS

About the UPS and the APCUPS device

An uninterrupted power supply (UPS) is provided with the BioCel 1800 System and the BioCel 1200 System, and is optional in the BioCel 900 System. The UPS is powered by a battery and supplies backup power to the system in case of a power outage.

You can check the battery level and power load of the UPS using the APCUPS Diagnostics. If you set up an alarm for the UPS, an error message would display when the battery level drops below the specified level.

Before you start

To check the status of the UPS, you must add the APCUPS device in the device file.

Note: The APCUPS device does not have associated tasks that you can add in a protocol.

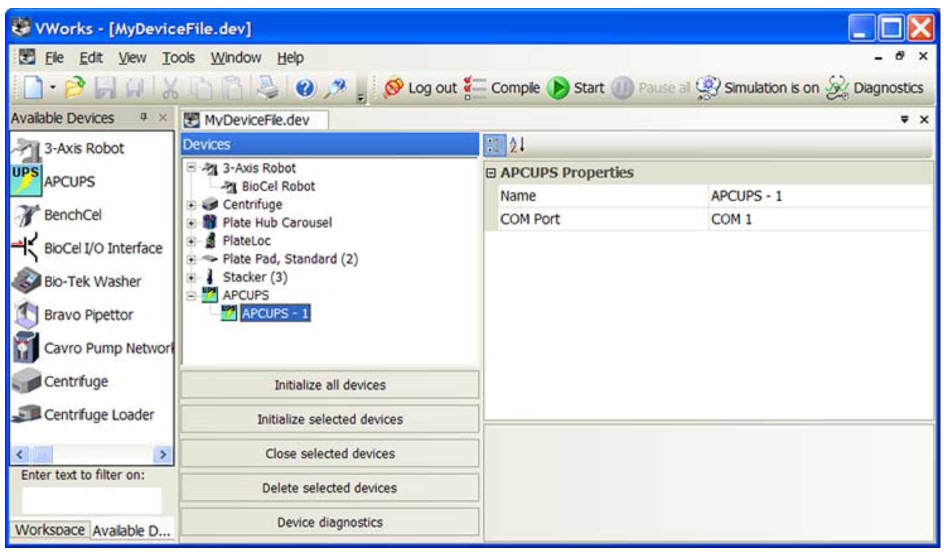
Procedure

If you have a UPS in the system, you can check the battery level and power load of the UPS at any time.

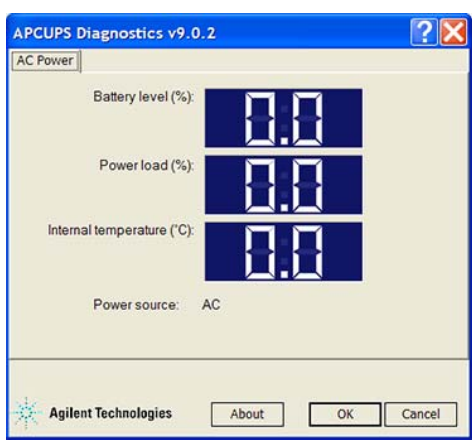
To check the status of the UPS:

1 In the **VWorks** window, open the device file that contains the UPS device.

2 In the **Device File** area, select **APCUPS**, and then click **Device diagnostics.**



The APCUPS dialog box opens.



3 Check the following:

Indicator	Value
Batter level	The amount of battery power remaining, in percent of the full level.
Power load	The amount of power drawn by the devices, in percent of the maximum load.
Internal temperature	The temperature, in degrees Celsius, of the internal battery.

Related information

For information about...	See...
Adding an alarm for the UPS	“Adding an alarm” on page 37
Contacting Agilent Technologies and reporting problems	“Reporting problems” on page 661

Reporting problems

Contacting Automation Solutions Technical Support

If you find a problem with the VWorks software, contact Automation Solutions Technical Support at one of the following:

Europe

Phone: +44 (0)1763850230

email: euroservice.automation@agilent.com

US and rest of world

Phone: 1.800.979.4811 (US only) or +1.408.345.8011

email: service.automation@agilent.com

Note: You can also send a software bug report from within the VWorks software.

Reporting hardware problems

When contacting Agilent Technologies, make sure you have the serial number of the device ready. See the device user guide for the location of the label.

Reporting software problems

When you contact Automation Solutions Technical Support, make sure you provide the following:

- Short description of the problem

- Software version number
- Error message text (or screen capture of the error message dialog box)
- Screen capture of the About VWorks software dialog box.
- Relevant software files

To find the VWorks software version number:

In the VWorks software, select **Help > About VWorks.**

To find the Diagnostics software version number:

1. Open **Diagnostics.**
2. Read the version number on the title bar of the diagnostics window.

To send compressed protocol and associated files in VZP format:

In the VWorks software, select **File > Export** to export and compress the following files:

- Protocol file
- Device file (includes the device profile and teachpoint file)
- Labware definitions
- Liquid classes
- Pipette techniques
- Hit-picking files
- Plate map files
- Barcode files
- Error library
- Log files
- Form file (*.VWForm)

Reporting user guide problems

If you find a problem with this user guide or have suggestions for improvement, send your comments using one of the following methods:

- Click the feedback button () in the online help.
- Send an email to documentation.automation@agilent.com.

Related information

For information about...	See...
Exporting a protocol and associated files	“Exporting and importing protocols and associated components” on page 626
VWorks software components: protocol, device file, and so on	“Relationship of VWorks components” on page 4
Viewing logs	“Viewing logs” on page 631

A
Quick reference

This appendix contains the following topics:

Menu commands

File menu

Command	Description
New > Protocol	Creates a new protocol. See "Creating a protocol: basic procedure" on page 13.
New > Device	Creates a new device file. See "Creating a device file" on page 26.
New > Runset	Clears the Runset Manager tab so that you can create a new runset. See "Managing runsets" on page 243.
New > Form	Opens the Form Designer so that you can create a new form to run a protocol or runset. See "Workflow for creating or editing a form" on page 160.
Open	Allows you to locate and open files, such as protocol, device, runset, and form files.
Close	Closes the displayed protocol, device, runset, or form file.
Save	Saves the changes in the protocol, device, or form file.
Save As	Opens the Save As dialog box and allows you to create a copy of the existing protocol or device file and specify a new name.
Save All	Saves all protocol, device, and form files currently open in the software.
Save Runset	Saves a runset in a folder that you specify.
Save Runset As	Opens the SaveAs dialog box, which enables you to save the runset file using a different file name and storage location.
Recent .pro files	Opens a recently opened protocol file.
Recent .dev files	Opens a recently opened device file.
Recent .rst files	Opens a recently opened runset.
Recent .VWForm files	Opens a recently opened protocol form.
Print	Prints the selected protocol, device, or form file.
Print Setup	Opens the Print Setup dialog box and allows you specify printing options.
Print Preview	Allows you to preview the printout.

Command	Description
Import	Imports a .vzp file that contains a protocol, runset, or form file and associated components. See “Exporting and importing protocols and associated components” on page 626.
Export	Exports a protocol, runset, or form file and associated components and stores them in a .vzp file. See “Exporting and importing protocols and associated components” on page 626.
Exit	Quits the VWorks software.

Edit menu

Command	Description
Cut	Removes selected text and stores it in memory.
Copy	Copies selected text and stores it in memory.
Paste	Pastes the text that is currently stored in memory.

View menu

Command	Description
Menubar	Displays or hides the menubar. When it is hidden, press ALT+V, or right-click in the protocol or device file area and select Menubar to display it.
Standard Toolbar	Displays or hides the standard toolbar.
Control Toolbar	Displays or hides the control toolbar.
Workspace	Displays or hides the Workspace area.
Available Tasks	Displays or hides the Available Tasks area.
Available Macros	Displays or hides the Available Macros area. For details on macros, see “Using macros to create protocols” on page 133.
Main Log	Displays or hides the Main Log in the log and progress area.
Pipette Log	Displays or hides the Pipette Log in the log and progress area.
Time Constraints Log	Displays or hides the Time Constraint Log in the log and progress area.

Command	Description
Status Bar	Displays or hides the status bar below the log and progress area.
Progress	Displays or hides the Progress table in the log and progress area.
Runset Manager	Displays or hides the Runset Manager in the log and progress area. For details on managing runsets, see "Managing runsets" on page 243.
Full Screen Mode	Available only if a form is open in the VWorks window and the form contains a Toggle Full Screen control. The Full Screen Mode command changes the VWorks window to a full screen display of only the form.

Tools menu

Command	Description
Labware Editor	Opens the Labware Editor. For details, see the *VWorks Automation Control Setup Guide*.
Liquid Library Editor	Opens the Liquid Library Editor. For details, see the *VWorks Automation Control Setup Guide*.
Pipette Technique Editor	Opens the Pipette Technique Editor. For details, see the *VWorks Automation Control Setup Guide*.
System State Editor	Opens the System State Editor.
Automatic Tip State Editor	Opens the Tip State Editor.
Configure Labware	Allows you to assign labware to device locations.
Backup Manager > Back up	Allows you back up system files.
Backup Manager > Restore	Allows you to restore system files.
Log Management	Allows you to back up or validate a selected log file.
Hit Pick Format Wizard	Allows you to set up the fluid transfer from a source microplate to a destination microplate.
Migrate All Files in a Folder	Allows you to migrate all files that are stored in a folder. For details, see the *VWorks Automation Control Setup Guide*.

Command	Description
Reload Plugins	Scans the Plugins folder and reloads the device plugins. The command is only available if all device files and protocol files are closed. Use the command if you have added a new device plugin in the Plugins folder and you want to reload all the plugins without restarting the software.
Open Hooks Plugin for	Allows you to open any VWorks plugins that are installed on the computer. For details on the Twitter plugin, see “Setting up automatic online notification” on page 235.
User Management	Allows you to manage users and privileges. For details, see the *VWorks Automation Control Setup Guide*.
Error Library	Opens the Error Library. For details, see “Setting up automated error responses” on page 654
Gantt Charts	Allows you to visually monitor the real-time status of processes, plate instances, and devices. For details, see “Tracking the run progress of instances or devices” on page 255
Get Status	Appends the current status to the Main Log. You can use this command at any time, including during a run.
Inventory Editor	Opens the Inventory Editor. For details, see the *VWorks Automation Control Setup Guide*.
Manage IO	Allows you to manage and set options for input and output signals.
Watcher is ON/OFF	Available only if you have a current license for the Watcher feature and the Watcher feature is configured. For details, see “Setting up and using the Watcher tool” on page 685.
Edit Form	Available only if a form is open in the VWorks window. Displays the Form Editor window that enables you to edit the form. For details, see “Workflow for creating or editing a form” on page 160.
Options	Allows you to specify global options in the software, including paths to various logs, e-mail setup, robot speed, database setup, and error handling. For details, see “Running a protocol” on page 209.

Window menu

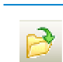
Command	Description
Cascade	Arranges the protocols and device files in the cascade style.
Tile	Arranges the protocols and device files in the tile style.
Arrange Icons	Aligns the icons in a sequential order.
Close All	Closes all open protocols and device files. If changes were made to the protocols and device files, a message will prompt you to save the changes.
Manage Windows	Opens the Window dialog box and allows you to arrange, minimize, or close the selected protocol and device files.

Help menu

Command	Description
VWorks Help	Opens the default browser window and displays the Knowledge Base.
Report a Bug	Allows you to send a software bug report to Agilent Technologies.
About VWorks	Displays the following: • VWorks software installer number • Copyright dates • License information • Product key • List of plugins installed

Toolbar buttons

Standard toolbar

Button	Description
	Creates a new device file or protocol.
	Opens the file you select.

Button	Description
	Saves the changes in the protocol or device file.
	Saves all protocol and device files currently open in the software.
	Removes selected text and stores it in memory.
	Copies selected text and stores it in memory.
	Pastes the text that is currently stored in memory.
	Prints the selected protocol, device, or form file.
	Displays the following: • VWorks software installer number • Copyright dates • License information • Product key • List of plugins installed
	Allows to you to display a help topic in context of where you are in the software. Click the Context Help button, and then click a user-interface item, such as a task icon, to display information about that item.

Control toolbar

Button	Description
Log in or Log out	Allows you to log in or log out of the software.
Compile	Compiles the selected protocol.
Start	Starts the protocol run.
Pause all	Pauses the protocol run.
Simulation is on or Simulation is off	Turns on or turns off the simulation mode.
Diagnostics	Opens the selected device diagnostics.

B
Managing digital signals

This appendix contains the following topics:

About the IO Manager

Description

The IO Manager allows you to assign signals from the Automation Control Unit, or equivalent I/O device, to:

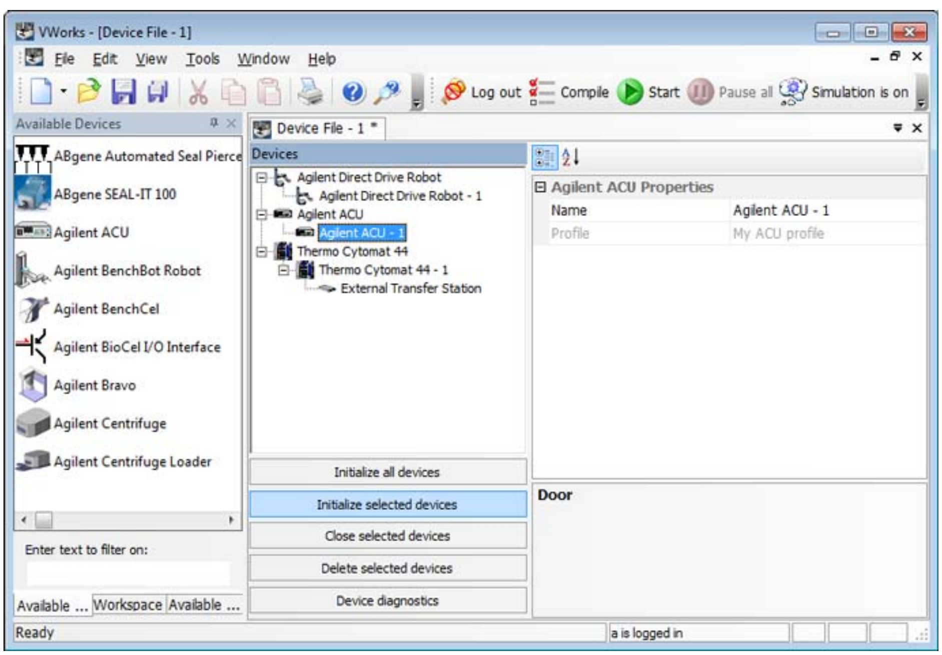
- Status lights and audible alarms
- Pass-through gates
- Spill detection or other sensors

Opening the IO Manager

IMPORTANT Digital signal channels that have been assigned a name will appear in the IO Manager dialog box. Channels that have the default port names will not appear in the IO Manager dialog box. For instructions on naming the digital signal channels, see the *Automation Control Unit User Guide* or the user documentation for the I/O device.

To open the IO Manager:

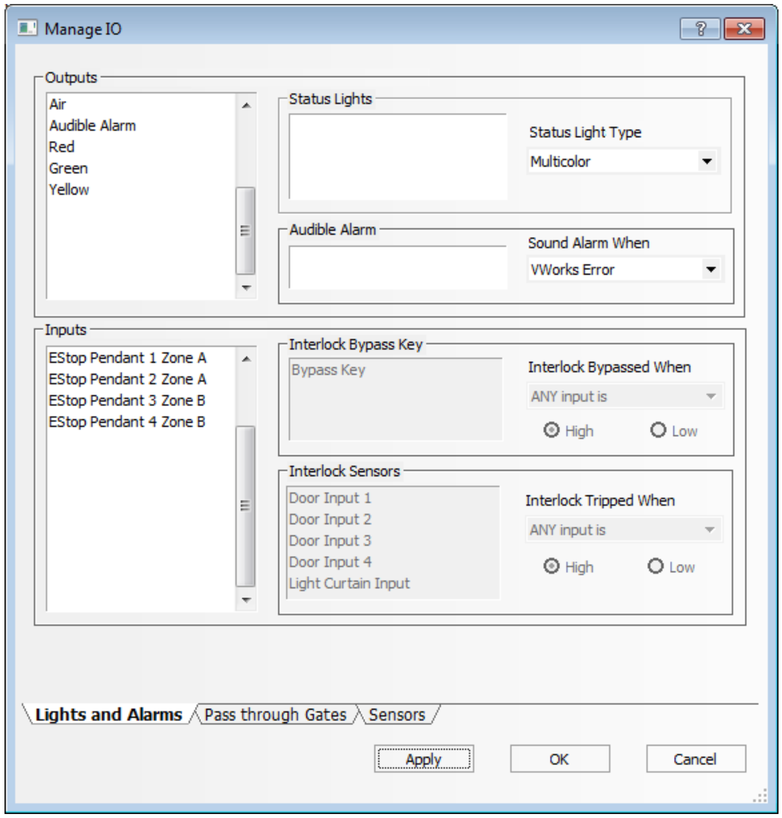
1 In the **VWorks** window, select the **ACU** or equivalent I/O device, and then click **Initialize selected devices.**



2 Select **Tools > Manage IO.** The Manage IO dialog box opens and displays the Lights and Alarms tab.

Notice the following:

- The **Outputs** list contains the renamed digital output channels.
- The **Inputs** list contains the renamed digital input channels. The fields in the Inputs area are automatically filled in and cannot be changed.



Related information

For information about...	See...
Location and instructions of the Automation Control Unit or I/O console	• *BioCel System User Guide* • Workstation user guide • I/O device user documentation
Setting up the Automation Control Unit	*Automation Control Unit User Guide*
Status light description	*BioCel System User Guide*
Writing protocols	"Creating a protocol: basic procedure" on page 13
Adding I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Setting up the BioCel I/O Interface	*BioCel System User Guide*, Revision 01, August 2010.

Assigning channels to lights and audible alarms

About this topic

In the Manage IO dialog box, you can assign digital output signals from the Automation Control Unit, or equivalent I/O device, to lights and audible alarms.

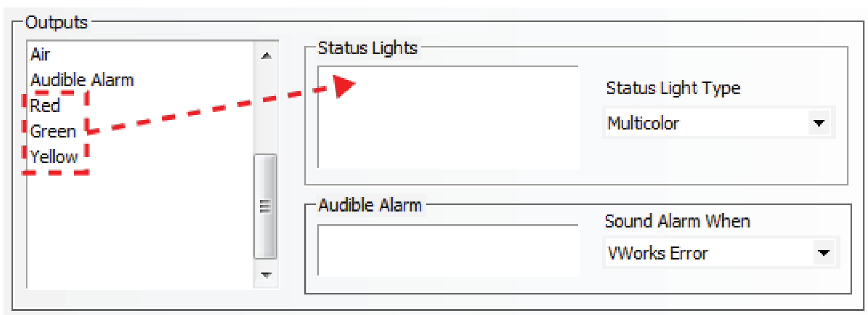
This topic explains how to do the following:

- "Assigning channels to lights" on page 674
- "Assigning channels to sounds" on page 675

Assigning channels to lights

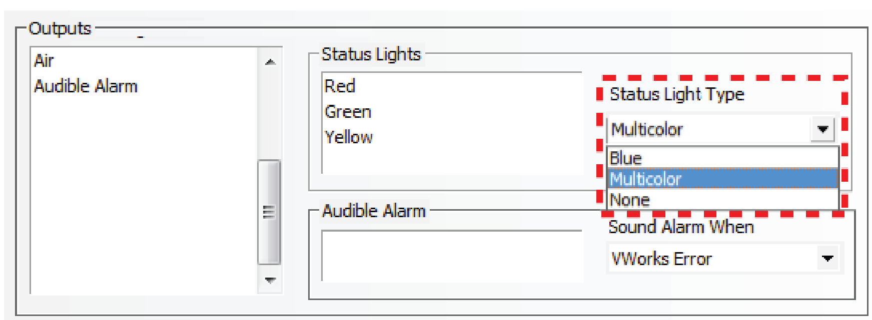
To assign digital output signals to lights:

1 In the **Lights and Alarms** tab, drag the names of light channels from the **Outputs** list into the **Status Lights** box. To use the multicolor lights, you need three channels, one for each of the colored lights, in the following sequence: red, green, and yellow.



2 In the **Status Light Type** list, select **Multicolor** (red, green, and yellow).

If your system does not use lights to indicate output signals, select **None**.

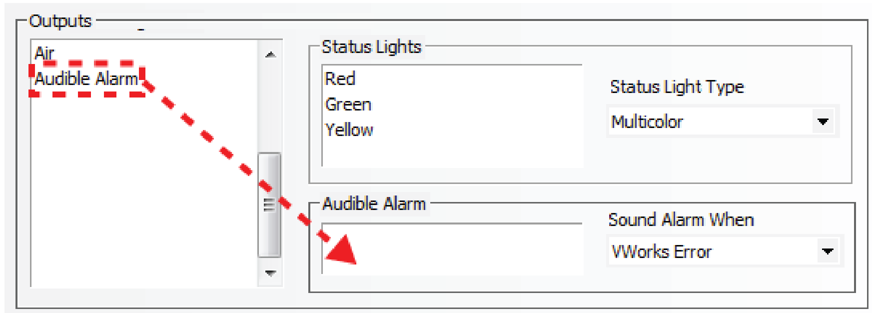


Note: The Blue status light type selection is provided to ensure backward compatibility with older systems only. To use the Blue light, you must have eight channels, one for each of the eight blue status lights at the top corners of the system.

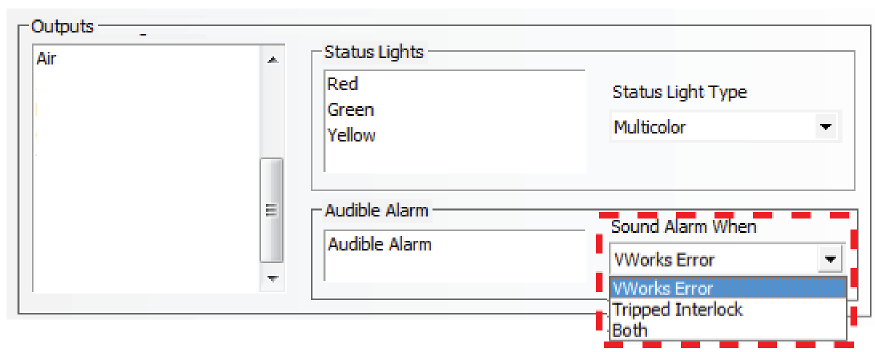
Assigning channels to sounds

To assign digital output signals to audible alarms:

1 In the **Lights and Alarms** tab, drag the names of the sound channels from the **Outputs** area to the **Audible Alarm** box.



2 In the **Sound Alarm When** area, select one of the following:



Selection	Description
VWorks error	Sound the alarm only when a run error occurs.
Tripped Interlock	Sound the alarm only when the interlock is tripped.
Both	Sound the alarm whenever a run error occurs or when the interlock is tripped.

Related information

For information about...	See...
Location and instructions of the Automation Control Unit or I/O console	• *BioCel System User Guide* • Workstation user guide • I/O device user documentation
Setting up the Automation Control Unit	*Automation Control Unit User Guide*
Status light description	*BioCel System User Guide*

For information about...	See...
Writing protocols	"Creating a protocol: basic procedure" on page 13
Adding I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Setting up the BioCel I/O Interface	*BioCel System User Guide*, Revision 01, August 2010.

Setting up channels for pass-through gates

About this topic

Some systems have an environmental-control option that creates fully contained environments within the system chamber. Automated pass-through gates can be used to permit labware to move between the system and external devices while maintaining the enclosed environment. External devices include an incubator below the system table or next to the system, the waste bin below the system table, and so on.

For each separated device that sits just beyond a pass-through gate, make sure you:

- Set up the signals for the gate in the I/O device. For each gate, two input signals and one output signal are required. One input signal is used to detect whether the gate is open. Another input signal is used to detect whether the gate is closed. The output signal is used to open or close the gate, depending on the control signal received.
- Select the signals that will be used to automate the gate actions.
- Associate the gate to a device.
- Specify the signals to use to automatically open or close gates during a run.

This topic explains how to set up I/O signals for pass-through gates.

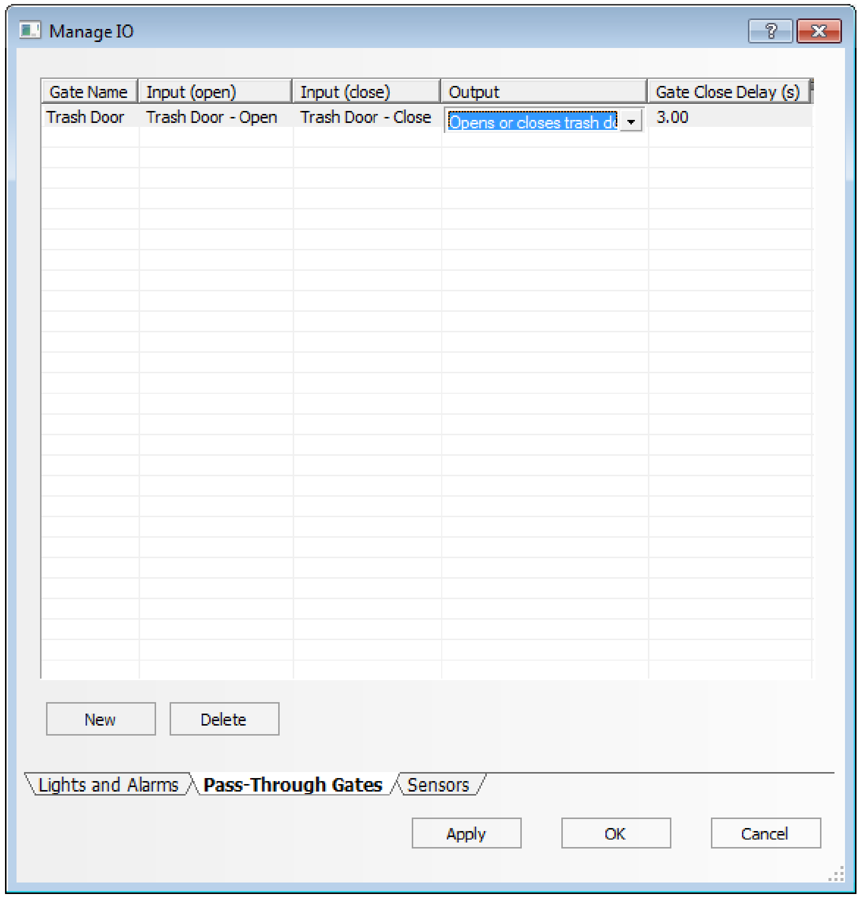
Before you begin

Pass-through gates require two 5 VDC digital input channels and one 24 VDC digital output channel. Make sure these channels are configured in the ACU Diagnostics I/O Setup tab before setting them up in the Manage IO dialog box. For instructions on configuring channels in the ACU Diagnostics I/O Setup tab, see the *Automation Control Unit User Guide*.

Procedure

To select signals to automate pass-through gate actions:

1. In the **Manage IO** dialog box, click the **Pass-Through Gates** tab.
2. Click **New** to add a gate.
3. Specify the following:



Parameter	Description
Gate Name	The name of the pass-through gate. Double-click in the field to type the name.
Input(open)	The I/O signal that detects whether the gate is open. Click the field to display the list of named signals, and then select one. If you do not see the signals in the list, initialize the Automation Control Unit.
Input(close)	The I/O signal that detects whether the gate is closed. Click the field to display the list of named signals, and then select one. If you do not see the signals in the list, initialize the Automation Control Unit.
Output	The I/O signal that changes to open or close the gate. Click the field to display the list of named signals, and then select one. If you do not see the signals in the list, initialize the Automation Control Unit.

Parameter	Description
Gate Close Delay (sec)	The length of time, in seconds, between when the robot moves labware through the gate and when the gate closes. Double-click in the field, and then type an integer.

4 When you are finished, click **OK** to save the changes and return to the VWorks window.

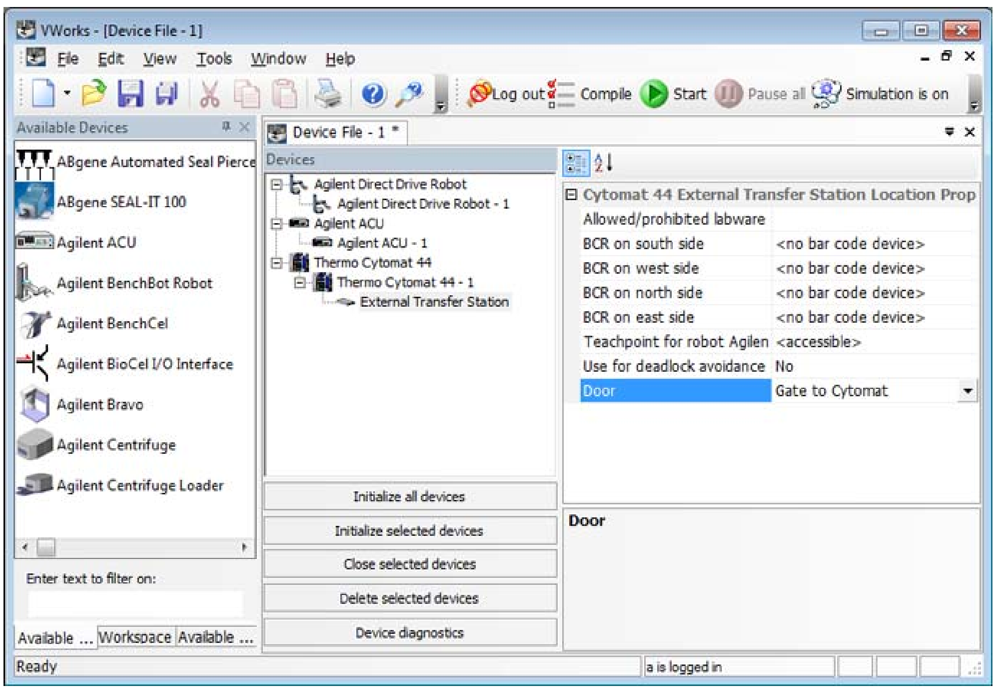
To associate the gate to a device:

1 In the device file, select the device.

2 In the device properties area, select the gate associated with the device from the **Door** list.

Note: You can open the IO Manager from the device properties area to edit or add gates. From the **Door** list, click **<Edit...>.**

In the following example, an automated gate was set up in ACU Diagnostics and the IO Manager for the Cytomat incubator device. The gate that was defined for the incubator in the IO Manager must be selected in the Cytomat device properties area.



Related information

For information about...	See...
Location and instructions of the Automation Control Unit or I/O console	• *BioCel System User Guide* • Workstation user guide • I/O device user documentation

For information about...	See...
Setting up the Automation Control Unit	*Automation Control Unit User Guide*
Status light description	*BioCel System User Guide*
Interlock bypass mode description	*Automation Control Unit User Guide*
Writing protocols	“Creating a protocol: basic procedure” on page 13
Adding I/O-handling tasks	“Setting parameters for I/O-handling tasks” on page 269
Setting up the Automation Control Unit	*Automation Control Unit User Guide*

Setting up channels for spill detection and other sensors

About this topic

Some systems with liquid management components use sensors to detect spills or other events. In the Manage IO dialog box, you can:

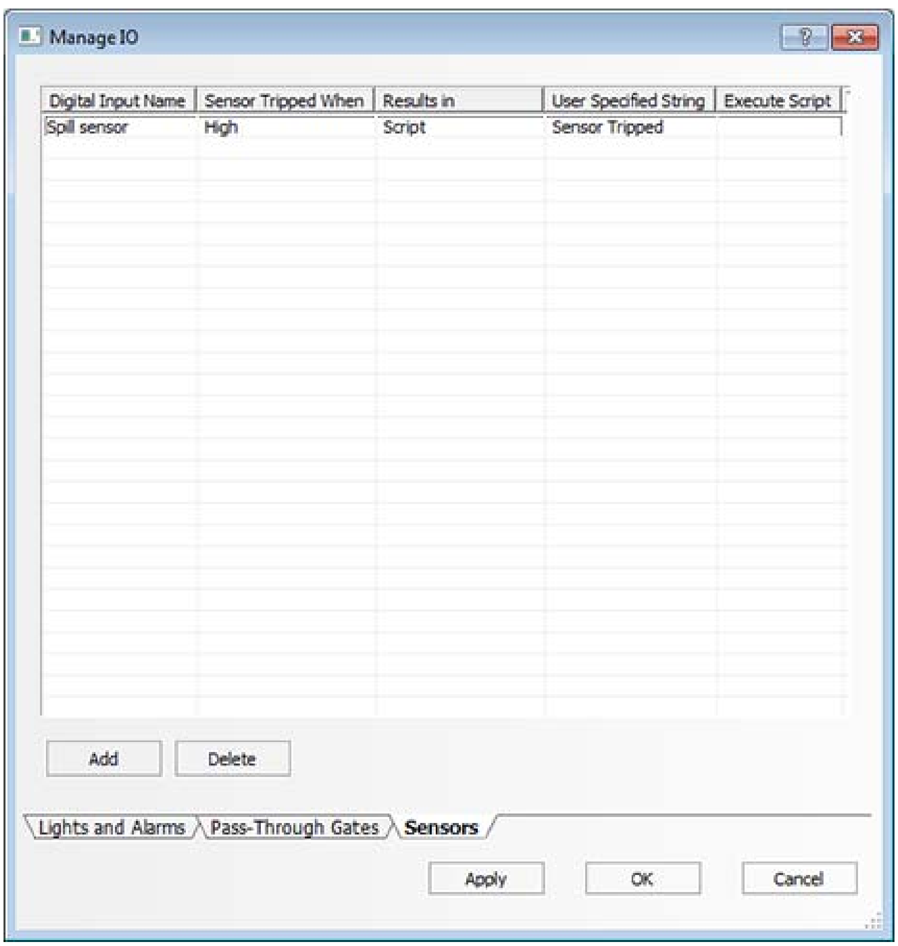
- Add new sensors.
- Specify when the sensor is tripped.
- Specify whether a detection event should be recorded in the log file, display an error message, or both.

This topic explains how to assign I/O channels to sensors.

Managing sensor signals

To set up channels for sensor signals:

1. In the **Manage IO** dialog box, click the **Sensors** tab.
2. Click **New** to add a sensor.
3. Specify the following:



Parameter	Description
Digital Input Name	The digital input signal used by the sensor. Click the field to display the list of channels, and then select the desired channel. If the sensor name does not appear in the list, make sure you have configured it in the I/O device diagnostics, such as ACU Diagnostics.
Sensor Tripped When	The state that indicates the sensor is tripped. Select one of the following: high-voltage signal (**High**) or low-voltage signal (**Low**). The system checks the sensor only during protocol runs. The sensor trips at the specified voltage signal, not the change in the voltage signal. For example, if the sensor should trip when the voltage signal is high, and signal is already high when you start the protocol, the sensor will automatically be tripped.
Results in	The software action when the sensor is tripped. Select one of the following: • **Log entry**. Records detection events in the log file. • **Message box**. Pauses the run and displays an error message when a detection event occurs. The error message explains which sensor is tripped. You can add a custom message by typing the desired text in the Custom Error String field. In the error message dialog box, you have the following options: – Fix the problem and resume the run. – Ignore the sensor (turn it off) and resume the run. The sensor will be turned on again after the protocol run is finished. • **Script**. Runs the specified JavaScript when the sensor is tripped. You can specify the script in the **Execute Script** box.
User-Specified String	The custom text to be displayed in the error message dialog box. Type the desired text in this field. This field is available only if you have selected **VWorks error message** from the **Results in** list.

Parameter	Description
Execute Script	The standard JavaScript or predefined JavaScript functions to run when the sensor is tripped. Type the script in the box. This box is only active if you selected **Script** in the **Results in** list.

4 When you are finished, click **OK** to save the changes and return to the VWorks window.

Related information

For information about...	See...
Location and instructions of the Automation Control Unit or I/O console	• *BioCel System User Guide* • Workstation user guide • I/O device user documentation
Setting up the Automation Control Unit	*Automation Control Unit User Guide*
Status light description	*BioCel System User Guide*
Interlock bypass mode description	*Automation Control Unit User Guide*
Writing protocols	"Creating a protocol: basic procedure" on page 13
Adding I/O-handling tasks	"Setting parameters for I/O-handling tasks" on page 269
Setting up the Automation Control Unit	*Automation Control Unit User Guide*

C
Setting up and using the Watcher tool

This appendix assumes that you know how to write programs in JavaScript or have basic programming knowledge. You also must have VWorks administrator or technician privileges.

The topics in this appendixsection are:

Watcher overview

Description

Watcher is a tool that automates the processing of designated files using JavaScript. Watcher monitors a designated folder for new file activity, and when a new file appears, Watcher runs a specified JavaScript to process the file. A couple of scenarios are described below, but your script might process files in a variety of other ways.

Scenario 1—Asynchronous post-processing of instrument-generated files

In this scenario, Watcher automates the post-processing of instrument-generated files as follows:

- A protocol includes a reader task that generates an output file and stores the file in a folder that is designated for monitoring by Watcher.
- When the new file appears in the folder, Watcher validates the file and then runs a script.
- The script parses out the relevant information from the instrument-generated file and performs additional tasks, such as aggregating the information across multiple files or saving the information in a database.

Scenario 2—Creating data-driven working protocols from a template protocol

In this scenario, the VWorks software is integrated with a LIMS. Watcher runs a script that automates the creation of a working protocol based on a template protocol and a LIMS-generated input file, where:

- *Input file*. A file that appears in the watched folder. In this scenario, the file specifies attributes of a single protocol run.
- *Template protocol*. A protocol that is used as the basis for creating a working protocol.
- *Working protocol*. The protocol that is created by JavaScript based on the template protocol and the input file.

When an input file appears in the Watcher monitored folder, Watcher runs a script. The script that has to parse the input file must be developed to recognize the file format. The script uses the information from the input file to modify the template protocol into a working protocol, and then schedules the newly created working protocol as part of a runset.

For an example script of this scenario, see "Creating the script that Watcher will run" on page 687.

Workflow for setting up and using Watcher

Step	For this task...	See...
1	Create the JavaScript that Watcher will run.	"Creating the script that Watcher will run" on page 687
2	Set up the Watcher configuration file.	• "Creating the configuration file" on page 691 • "Specifying the configuration file location" on page 693

Step	For this task...	See...
3	Turn on the Watcher.	"Turning on Watcher" on page 695

Creating the script that Watcher will run

About this topic

This topic assumes that you know how to write programs in JavaScript or have basic programming knowledge.

For a full description of the JavaScript language, see the Mozilla Developer Center at http://www.mozilla.org/js/.

Guidelines for creating your script

In addition to using good script-writing practices, follow these guidelines when creating a script for Watcher to run:

- *Define the JavaScript function correctly*. The function that Watcher will run, for example WatcherMain, must take one parameter:

Parameter	Type	Description
input_file	string	The file path to the input file.

The name of the function in the script must match the function name in the Watcher configuration file (.ini).

- *Identify or create any other scripts that your script will call.* Automation Solutions has JavaScript files with predefined objects and functions that you may use, including:

Script file name	Description
DeviceArbitration.js	Enables device pooling in JavaScript.

FileUtilities.js	Provides the following functions:
	`isFileExist(filename)`
	`DeleteFile(filename)`
	`StripPath(full_path) //for example, changes "c:/mydir/mysubdir/myfile.ext" to "myfile.ext"`
	`ForwardToBackSlashes(full_path) //for example, changes "c:/mydir/mysubdir/myfile.ext" to "c:\mydir\mysubdir\myfile.ext"`

ProtocolEditor.js	Provides various functions for editing a protocol.
StopGo.js	Provides a way to create stop and go tasks using JavaScript, which is useful when stop and go should be scripted based on instance number.
VIN_handling.js	Provides a way to assign virtual instance numbers (VIN) to plates. This is useful if instance numbers: • Must be passed from process to spawned process. • Require an out-of-order assignment, for example, if rejecting some plates, sequential [virtual] instance numbers can be assigned to the plates that remain.
XMLKeyValueLookup.js XMLKeyValueLookup SampleData.xml	Script and sample data for mapping one string to another string, where the definition is in XML format. For example, you might use this script to translate an alias into a final file name.
XML_files.js and Formatter.xml	Provides script for opening XML files, validation against schemas, and saving. Uses MSXMLDOM ActiveX, which requires Msxml2.DOMDocument.6.0 and Msxml2.XMLSchemaCache.6.0.
File_operations.js	Provides functions for various file operations, such as creating, reading, writing to, and debugging files.
plateDB_HowTOUse.js	Provides examples on how to use the VWorks plateDB object, which can be accessed by a script. For a description of the plateDB object, see "plateDB object" on page 112.

- *Define any required global variables, such as folder paths.* You should define the folder paths where processed files will be stored, such as the VWorks folder, the working folder, and an output folder, if applicable. For example, you might define the following:

```
var VWorksFolder     = "C:/VWorks Workspace/"
var WorkingFolder    = VWorksFolder + "WorkingFolder/"
```

```
var OutputFolder    = VWorksFolder + "Output/"
```

- *Include code at the end of the script to delete the input file from the monitored folder, if applicable.* To prevent repeated processing of files upon restarting Watcher, the script should delete the input file or move it into a processed folder or output folder after processing the file.

  Example script:

  ```
  print("deleting input file: [" + input_file_name + "]")
  DeleteFile(input_file_name);
  ```

Script example—Creating data-driven working protocols from a template protocol

This section provides example script for scenario 2 in "Description" on page 686. In this scenario, Watcher monitors a folder for new input files, each of which specifies the attributes of a single protocol run. When a new input file appears in the folder, Watcher runs a script to create a working protocol based on a template protocol and an input file, and then schedules the protocol.

The following script example shows a hypothetical WatcherMain function, which does following:

1. Verifies that the input file exists.
2. Reads the input file.
3. Opens the template protocol.
4. Modifies the template protocol to create a working protocol.
5. Saves the input file and the modified protocol (working protocol) to the working folder.
6. Schedules the working protocol as part of a runset.
7. Deletes the input file from the monitored folder.

To accomplish some of these tasks, the WatcherMain function includes calls to other predefined JavaScript functions, such as inputParser and protocolEditor.

WatcherMain function example

```
function WatcherMain(input_file_name){
   print("Starting WatcherMain...with input file: [" + input_file_name + "]")
   print("OutputFolder : [" +OutputFolder+"]")

   if(!isFileExist(input_file_name))
       Print("input file does not exist");

   var inputParser = new InputParser();
   inputParser.Open(input_file_name);

   var protocol_file_name = inputParser.getOrderAttribute("protocol")
   print("\n protocol file: " + protocolKey);

   print("\n protocol_file_name: " + protocol_file_name);

   var time_string = getTimeString()

   var protocolEditor = new ProtocolEditor();
   protocolEditor.Open(protocol_file_name);

   var working_input_filename = WorkingFolder + time_string + "_" +
    StripPath(input_file_name)
   print("saving working input file to: ["+working_input_filename+"]")
   inputParser.Save(working_input_filename)
   print("success")

   print("modifying protocol")
   ModifyProtocol(inputParser, protocolEditor, working_input_filename);
   print("success")

   var working_protocol_filename= WorkingFolder + time_string + "_" +
    StripPath(protocol_file_name)
   print("saving working protocol to: ["+working_protocol_filename+"]")
   protocolEditor.Save(working_protocol_filename);
   print("success")

   runset.appendProtocolFileToRunset(working_protocol_filename, 1, "this is
    a note: blah", false)


   print("deleting input file: [" + input_file_name + "]")
   DeleteFile(input_file_name);

}
```

Related information

For information about...	See...
Getting a license for Watcher	Automation Solutions Customer Service
Configuring the Watcher feature	“Setting up the Watcher configuration file” on page 691
Creating the JavaScript	“Creating the script that Watcher will run” on page 687
Turning on or off the Watcher feature	“Turning on Watcher” on page 695

Setting up the Watcher configuration file

About this topic

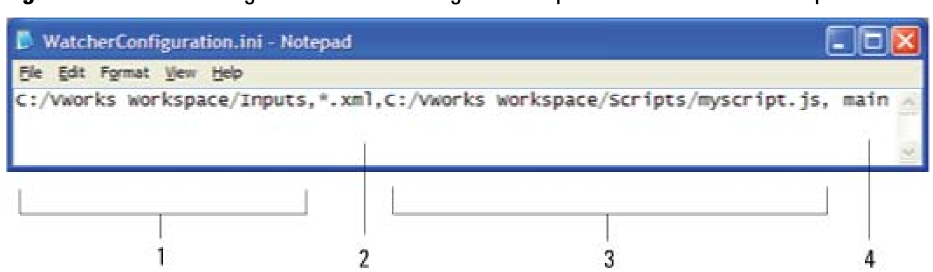
This topic assumes that you know how to write programs in JavaScript or have basic programming knowledge. You must also have VWorks administrator or technician privileges.

Creating the configuration file

Watcher requires a single configuration file (.ini) that contains one entry (monitoring condition) per line. Each monitoring condition entry must consist of the following four values separated by commas:

<folder to monitor>,<file filter>,<JavaScript file location>, <JavaScript function>

Figure Watcher configuration file showing an example of the four comma-separated values



Item	Value	Description
1	Folder to monitor	The full path to the file folder that Watcher will monitor for new files. Use forward slashes (/) as a path delimiter.

Item	Value	Description
2	File filter	The file name extension of the file type to be monitored. The filter applies to file name extensions only. You cannot filter by partial file names.
3	JavaScript file location	The full path to the JavaScript file. Use forward slashes (/) as a path delimiter.
4	JavaScript function	The function in the JavaScript file that you want to run when the desired files appear in the monitored folder.

Configuration example—Monitoring one folder for one file type

In the following example, Watcher will monitor the Inputs folder for the file type .xml. When an .xml file is added to the folder, Watcher will run myscript.js and call the `main` function. The function `main` must be defined to take in one parameter, the full path of the file that has appeared in the folder to be watched.

- Contents of .ini file:

  ```
  C:/VWorks Workspace/Inputs,.xml,C:/VWorks Workspace/
  Scripts/myscript.js, main
  ```

- Contents of myscript.js file:

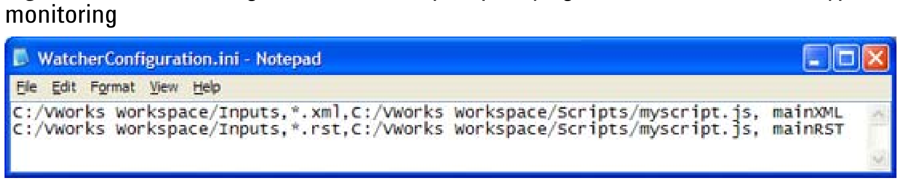
  ```
  function main(input_file){
      print("in main: " + input_file)
  }
  ```

Configuration example—Monitoring one folder for more than one file type

The Watcher configuration file (.ini) can specify multiple monitoring conditions, but each condition must be stated on a single line and consist of the four comma-separated values.

In the following example, Watcher will monitor the Inputs folder for the file types .xml and .rst (runset). When either file is added to the folder, Watcher runs myscript.js and calls the corresponding JavaScript function for the file type. The functions `mainXML` and `mainRST` must be defined to take in one parameter, the full path of the file that has appeared in the folder to be watched.

Figure Watcher configuration file example specifying one folder and two file types for monitoring


WatcherConfiguration.ini - Notepad

File Edit Format View Help

```
C:/VWorks Workspace/Inputs,*.xml,C:/VWorks Workspace/Scripts/myscript.js, mainXML
C:/VWorks Workspace/Inputs,*.rst,C:/VWorks Workspace/Scripts/myscript.js, mainRST
```

For this example, the myscript.js file would contain the following code:

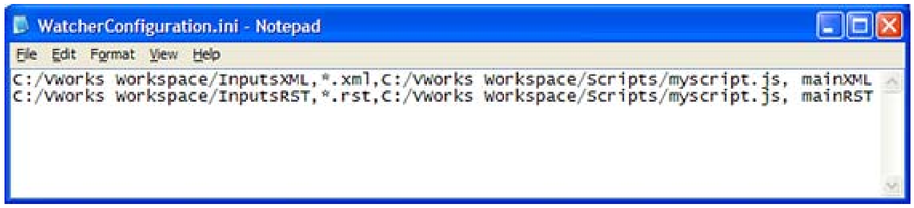
```
function mainXML(input_file){
    print("in mainXML: " + input_file)
}
function mainRST(input_file){
```

```
    print("in mainRST: " + input_file)
}
```

Configuration example—Monitoring more than one folder

You can specify that more than one folder be monitored and run different scripts for different file types that appear in each folder.

Figure Watcher configuration file example specifying two folders and two file types for monitoring



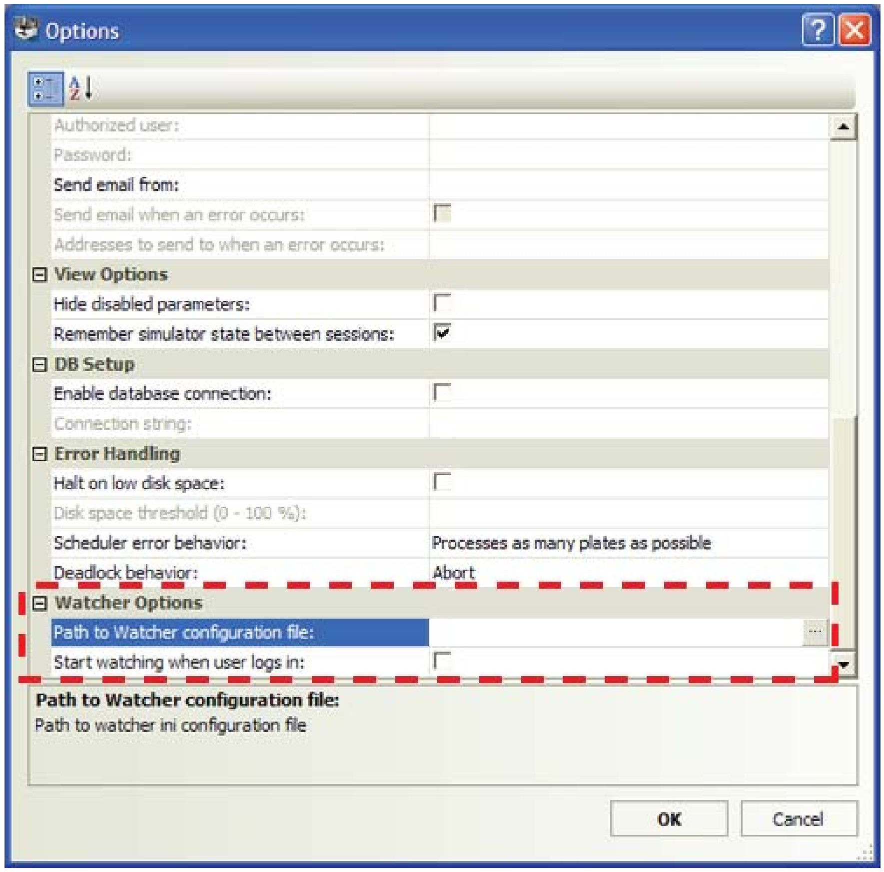
Using the same script from the previous example, this configuration file would do the following:

- Monitor the InputsXML folder for new .xml files, and then call the `mainXML` function. The `mainXML` function must be defined to take in one parameter, the full path of the file that has appeared in the folder to be watched.
- Monitor the InputsRST folder for new .rst files, and then call the `mainRST` function. The `mainRST` function must be defined to take in one parameter, the full path of the file that has appeared in the folder to be watched.

Specifying the configuration file location

To set the configuration file location:

1 Select **Tools > Options.** The Options dialog box opens.



2 In the **Watcher Options** area, click the field next to **Path to Watcher configuration file**, and then click the ... button that appears.

3 In the **Open** dialog box, select the desired location and click **Open.**

4 In the **Options** dialog box, verify the new path, and then click **OK**.

Related information

For information about...	See...
Getting a license for Watcher	Automation Solutions Customer Service
Creating the JavaScript	“Creating the script that Watcher will run” on page 687
Turning on or off the Watcher feature	“Turning on Watcher” on page 695
Reporting problems with the software	“Reporting problems” on page 661

Turning on Watcher

About this topic

This topic assumes that you have VWorks administrator or technician privileges.

About monitoring new files

The first time you turn on Watcher, the program runs the specified JavaScript for the existing files in the folder that is configured for monitoring. Subsequently, Watcher processes only the new files that appear in the folder.

Watcher will not process existing files whose contents were modified. For example, if a new file replaces an existing file with the same file name, Watcher does not reprocess the file. If you rename an existing file, Watcher processes the renamed file as a new file.

Turning on and turning off Watcher

You can turn on Watcher manually or automatically every time you log in to the VWorks software. You can turn off Watcher manually.

To turn on Watcher manually:

1 Select **Tools > Watcher is OFF**. The menu command changes to Watcher is On.

To turn on Watcher automatically every time you log in:

1 Select **Tools > Options.** The Options dialog box opens.
2 In the **Watcher Options** area, select **Start watching when user logs in.**

The next time a user logs into the VWorks software, Watcher will turn on automatically and remain on after the user logs out.

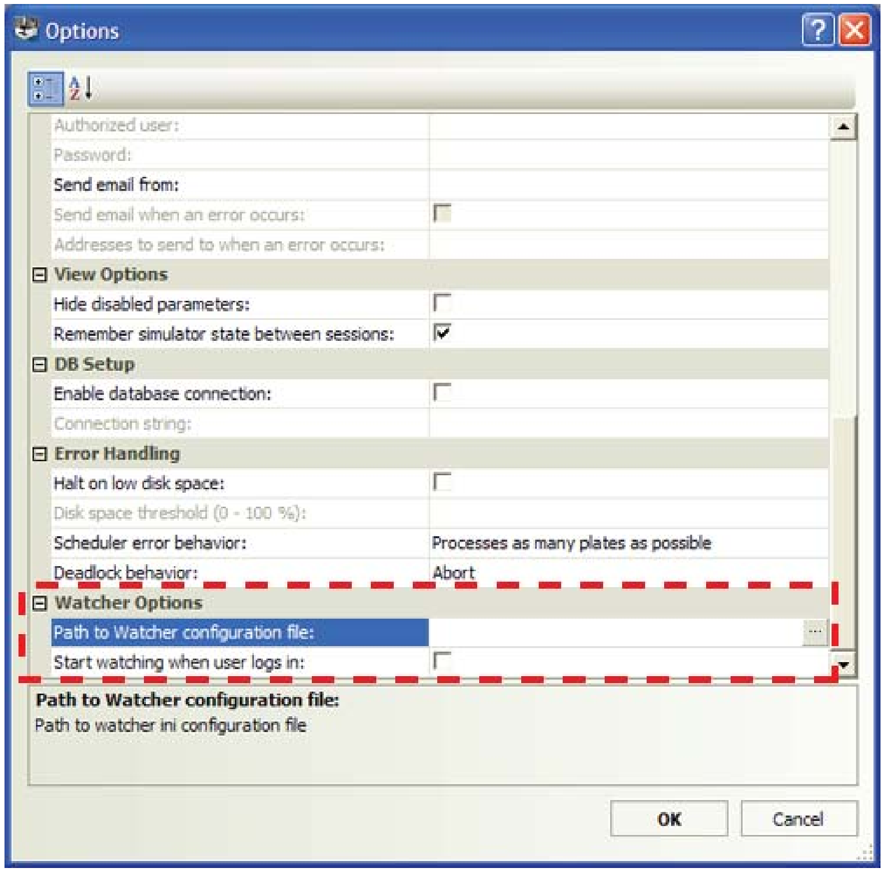
To turn off Watcher:

1 Select **Tools > Watcher is On**. The menu command changes to Watcher is Off.

To prevent Watcher from turning on automatically when a user logs in:

1 Select **Tools > Options.** The Options dialog box opens.
2 In the **Watcher Options** area, clear the **Start watching when user logs in** check box.

Figure Options dialog box showing the Watcher Options area



Related information

For information about...	See...
Reporting problems	"Reporting problems" on page 661
Log file directory settings	"Setting log file directories" on page 222
General and view options	"Setting general and view options" on page 224
Error-handling options	"Setting error-handling options" on page 229
Starting a protocol run	"Starting the protocol run" on page 237

Index

Agilent Technologies

User Guide

G5415-90063

Revision I, October 2014