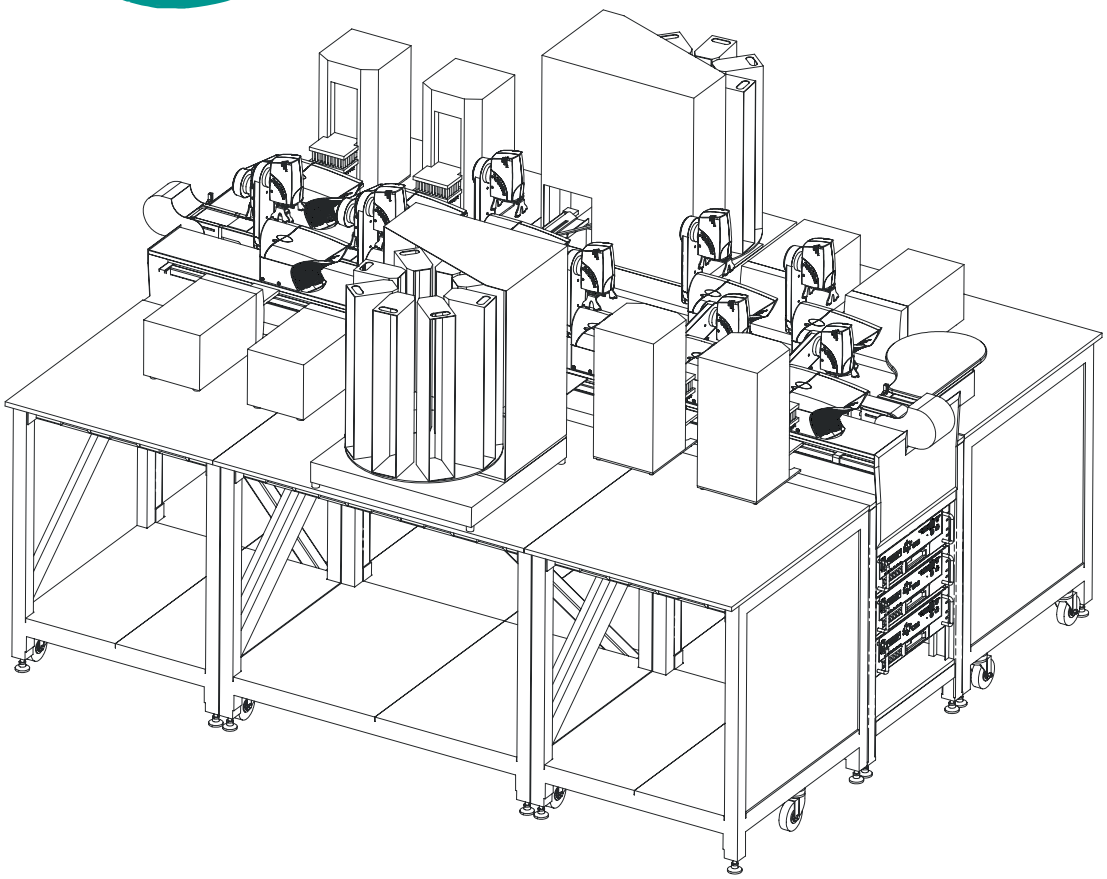
Thermo Fisher Scientific Inc.

POLARA 2.3 Researcher Guide

Version 2.3.5

POLARA



Thermo SCIENTIFIC

POLARA Researcher Guide 2.3

Version	Changes Made	Date
003	POLARA 2.3: No changes in content; version number change only.	08-2004
004	POLARA 2.3: Updated corporate names and trademark attributions.	11-2004
005	POLARA 2.3.5: Updated chapter 1 to introduce POLARA user interface; added description of new features in 2.3.5, including legend in scheduler	08-2007

About This Guide

This guide describes how to use Thermo Fisher Scientific's POLARA™ laboratory automation software.

Who Uses This Guide

This guide is intended for users of POLARA-based automated laboratory systems. These users perform the following tasks:

- Defining the sequence of operations a POLARA system uses to process a sample
- Defining the schedule a POLARA system follows when it processes one or more batches of samples
- Performing runs on the POLARA system

This guide assumes the reader knows how to use Microsoft® Windows 2000 or XP® and Thermo Scientific container transport systems.

How to Use This Guide

This guide is task-based and uses navigational aids to help you quickly find the topics and information you need. If a technical term is not familiar to you, refer to the **Glossary**.

Throughout this manual warnings are marked by a '!' symbol in the left margin. Failure to comply with these warnings can result in system errors, memory loss, or damage to the robot and its surroundings. ▲

Before attempting to follow instructions or examples in a topic, read the entire topic first.

This manual is not intended as a self-teaching guide. To use POLARA, you must have received training from Thermo LACI.

Note If you are viewing this Guide using Adobe® Acrobat® Reader®, you can click on any text highlighted in blue to jump to the referenced topic. ▲

Chapter Summary

This guide consists of the following chapters:

- **Introducing POLARA**, which provides an overview of the POLARA system architecture and the tasks you perform.
- **Safety**, which provides an overview of how to maintain safe operation of the POLARA system
- **Using Workspaces and Profiles**, which describes how to open workspaces and how to define sets, or *profiles*, of the lab system hardware.
- **Working with Methods**, which describes how to define the sequence of unit operations, or *method*, used to process a batch of samples
- **Creating and Optimizing Schedules**, which describes how to define schedules and optimize them for maximum throughput
- **Setting up and Performing a Run**, which describes how to process batches of samples on a POLARA lab system
- **Glossary**, which defines terms unique to POLARA and other products from Thermo LACI
- **Index**, which contains an index to subjects in this guide.

On-Line Help in POLARA

The POLARA users' guides, in on-line help format, are installed on the system computer with the POLARA software. Which version of the guide you access from POLARA depends on your user privileges: administrators get the *Administrator Guide*, researchers get the *Researcher Guide*, and operators get the *Operator Guide*.

To access on-line Help

- Select Contents from the Help menu. The on-line guide is opened with the first page displayed, along with Contents, Index, and Search tabs.
- On any screen or form, click the Help button. The on-line guide is opened at the page with instructions or information related to the tasks you perform on that screen or form.

For More Information

For information about the other components in your lab system, see their user guides.

You can obtain copies of these user guides or other Thermo LACI literature from the Customer Support Group.

Training

Thermo LACI offers training courses in POLARA and other Thermo LACI products at its facility in Burlington, Ontario, Canada, or onsite at your facility. For additional information, contact the Thermo LACI Training Department.

Surface Mail/Shipping

Thermo Fisher Scientific Inc.
Integrative Technologies Division
Laboratory Automation and Cellular Imaging
5344 John Lucas Drive
Burlington, Ontario L7L 6A6
Canada

Telephone

1-905-332-2000 (voice)
1-800-365-7587 (voice: toll free in Canada and United States)
1-905-332-1114 (facsimile)

E-Mail

Sales: sales.labautomation@thermofisher.com
Customer Support and Training: services.labautomation@thermofisher.com
General: info.labautomation@thermofisher.com

World Wide Web

www.thermofisher.com

Contents

Chapter 1 Introducing POLARA

This chapter provides an introduction to POLARA™ and the lab systems it controls. It covers the following topics:

- "What is POLARA?" on page 1-2, which provides an overview of the POLARA software and the lab systems it controls
- "How the POLARA Software Works" on page 1-8, which provides an overview of how POLARA functions
- "What You Need to Know to Use POLARA" on page 1-13, which describes what each type of POLARA user must know before using POLARA
- "The POLARA Main Window" on page 1-15, which describes the POLARA main window and explains the function of the toolbar buttons and the menu bar

What is POLARA?

POLARA is a suite of software components that enables you to integrate and control an automated laboratory system.

What are POLARA Laboratory Systems?

POLARA laboratory systems automate the processing samples through a protocol, such as an assay or method.

Figure 1-1 shows an example of a POLARA application: an automated screening system that runs simple ELISA assays.

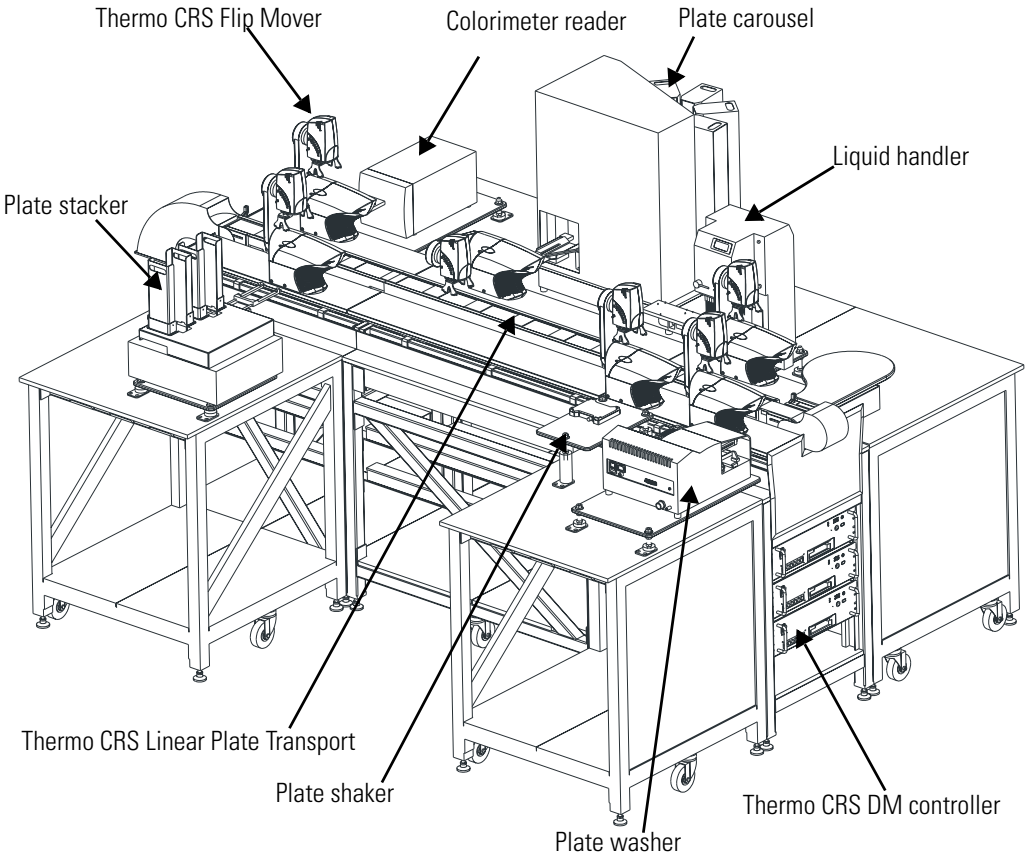


Figure 1-1. A POLARA Distributed Motion system: automating simple ELISA assays

The POLARA software controls all the hardware in the system, commanding the container transport to move microtitre plates from instrument to instrument, and commanding the instruments to process each microtitre plate as required by the assays. The POLARA software tracks the movement of each microtitre plate through the system and stores the data read from each microtitre plate by the colorimeter reader.

What Comprises a POLARA-based Lab System?

A POLARA automated laboratory system includes the following types of items:

- The laboratory instruments and peripherals needed to implement the required biological or chemical protocols. These typically include readers, liquid handlers, and incubators.
- A Microsoft Windows 2000 or XP lab system computer running the POLARA software
- A container transport. POLARA can control two types of container transports:
  - POLARA DM (distributed motion) systems that dedicate a container mover to each instrument. Figure 1-1 shows an example of a POLARA DM lab system.
  - POLARA AR (articulated robot) motion systems that use a single articulated robot or CRS Vertical Array Loader (VAL) to serve all instruments. Figure 1-2 shows an example of a POLARA AR lab system.
- One or more container storage devices, such as CRS Microplate Carousels and CRS Platefeeder Hotels.
- A table that supports the lab system hardware and provides electric and pneumatic power

A POLARA laboratory system can also include the following types of items:

- A beacon, which indicates the operational status of the lab system
- A barcode reader, which can be used to identify individual containers
- Programmable logic controllers (PLC), which provide additional control inputs and outputs (I/O) for the lab system computer
- An uninterruptible power supply (UPS), which enables the system to remain powered for a short period after a main power supply failure
- Barriers, which may include safety interlocks that restrict operator access to lab system components

Most of these components communicate with the lab system computer over serial ports and/or through digital I/O ports. The lab system computer uses these communication lines to receive data from and send commands to the other hardware components.

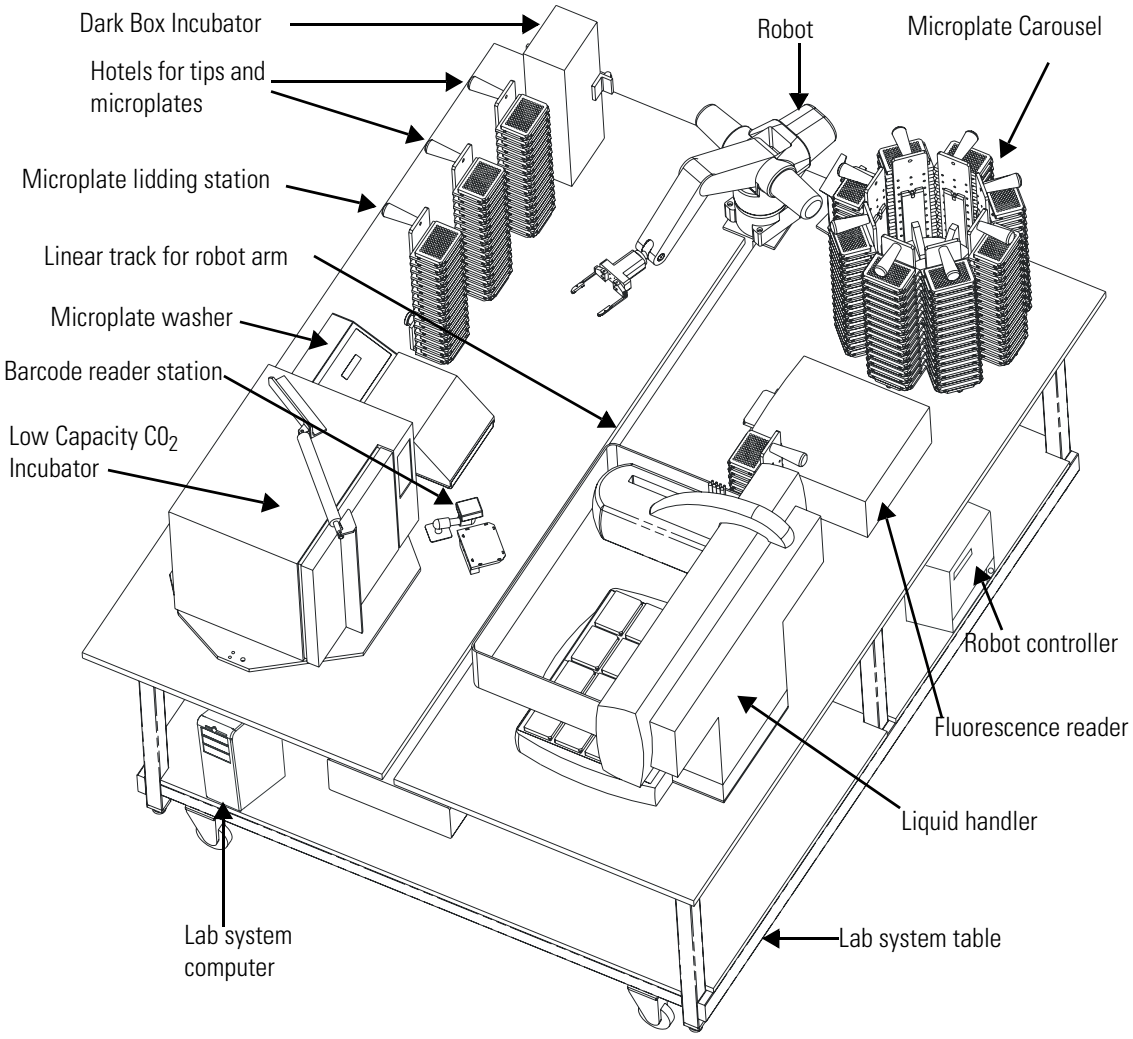


Figure 1-2. A POLARA AR lab system: automating cell-based screening assays

What Do You Do with a POLARA Lab System?

You use a POLARA lab system to automate laboratory protocols, such as methods and assays. You define the protocols in POLARA and then use POLARA's scheduler to generate the instructions the lab system follows to process the required number of samples through the protocols. The scheduler enables you to process samples through different protocols in

parallel on the lab system, which boosts throughput and makes more efficient use of lab system resources and personnel.

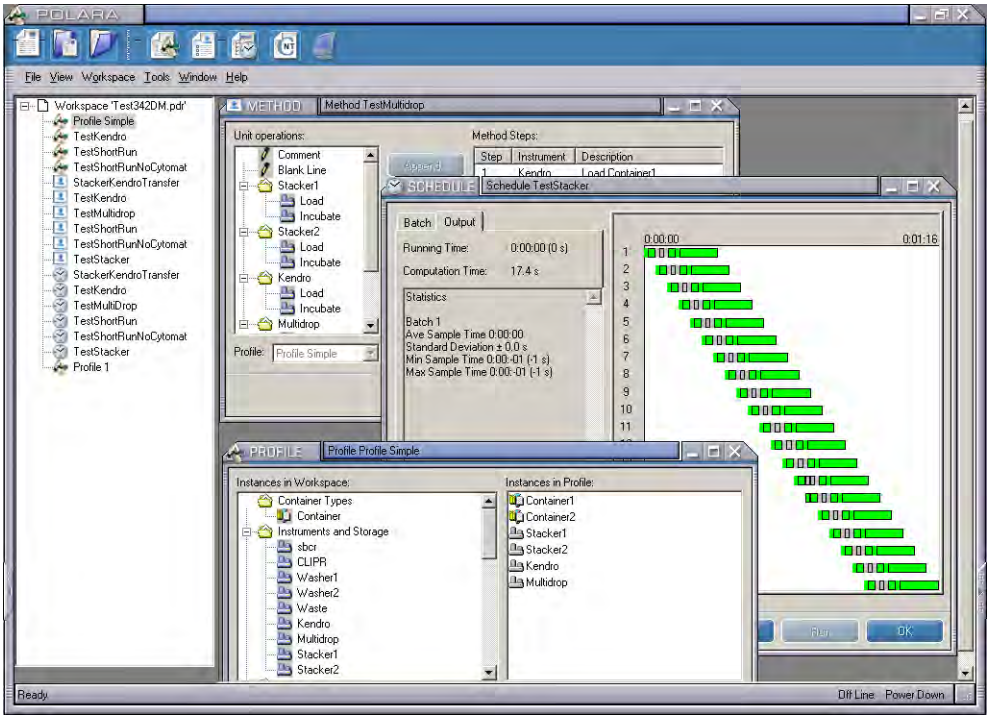


Figure 1-3. POLARA provides a rich set of editors with which to define protocols and generate schedules.

The POLARA software instructs the lab system operator on what instruments to prepare to run a schedule, and how many containers to load. As the run progresses, POLARA tracks each sample as it moves through the protocol. If POLARA encounters any problems, such as a missing container, it alerts the operator. (If it is running in unattended mode, it will first attempt to deal with the problem on its own.) The operator can suspend or abort a run at any time.

If there is an instrument failure, power outage, or some other reason for the operator to abort a run, the operator can save run recovery information. When the system is restarted, POLARA detects the saved information and asks whether the interrupted run should be recovered or abandoned.

When the run is complete, the operator collects the sample data generated during the run, and the POLARA software readies the system to perform another run.

How Do You Define a Protocol in POLARA?

To define a protocol in POLARA, you specify the following parameters:

- **The instruments and containers to use.** POLARA enables you to define the set of instruments and containers you need to implement the protocol. POLARA calls this set a *profile.*
- **The steps in the protocol.** POLARA enables you to define the individual steps that make up the protocol. POLARA calls the complete series of steps a *method*.

How Do You Create a Schedule in POLARA?

Once you have defined the steps that make up the protocol, you create a schedule that POLARA uses to generate the instructions the lab system follows to implement the protocol on the required number of samples.

To create a schedule in POLARA, you specify which method(s) to use to process the samples and how many samples to process through each method.

POLARA provides detailed schedule reporting and analysis functions that help you optimize the schedule for maximum throughput and sample processing uniformity.

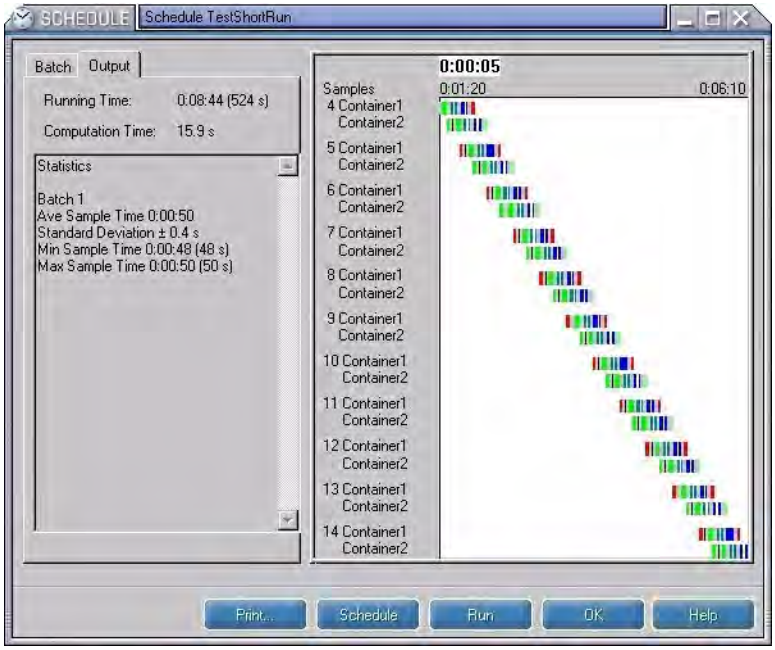


Figure 1-4. The POLARA Scheduler provides detailed timing reporting and analysis functions that help you optimize a schedule for maximum throughput.

Note You can install POLARA on an off-line workstation, enabling you to develop profiles, methods, and schedules without tying up the lab system. ▲

The scheduler generates a *sequence* file, which contains the instructions that control the run.

At any time during a run, an administrator or researcher can add samples to the run and generate a new schedule for the remaining samples and the added samples. The added samples can follow the methods followed by the other samples in the run, or a new method, provided that all methods use the same profile of instruments.

POLARA System Users

POLARA supports three different types of users: administrators, researchers, and operators. Each type of user is authorized to perform a different set of tasks:

- **Operators** can perform all tasks associated with performing a run, including selecting the schedule to use, specifying how many samples to process in the run, and controlling the run itself.

- **Researchers** can define and modify all aspects of a schedule, including which instruments to use (of the ones available) and what methods to perform with those instruments. Because they might need to perform test runs, researchers also have access to operator-level functions.

- **Administrators** can define which instruments are available for use in a lab system and what type of access each user has to the system's capabilities. They can also function as integrators, adding new instruments to the lab system, or even integrating a Thermo LACI container transport and several instruments into an entirely new system. Administrator users have full access to all POLARA features.

When using POLARA, you will only see options that are available for your user level.

How the POLARA Software Works

The POLARA software suite consists of the following main components:

- **The POLARA Windows application.** This component, which runs on the lab system computer, provides the user interface to the POLARA system.
- **CROS for Windows, or CROSnt.** The programs that directly control the lab system hardware run under CROSnt, an operating system that runs as a process within Windows.
- **Instrument interfaces.** These components provide the software interfaces between the lab system computer and the lab system hardware.

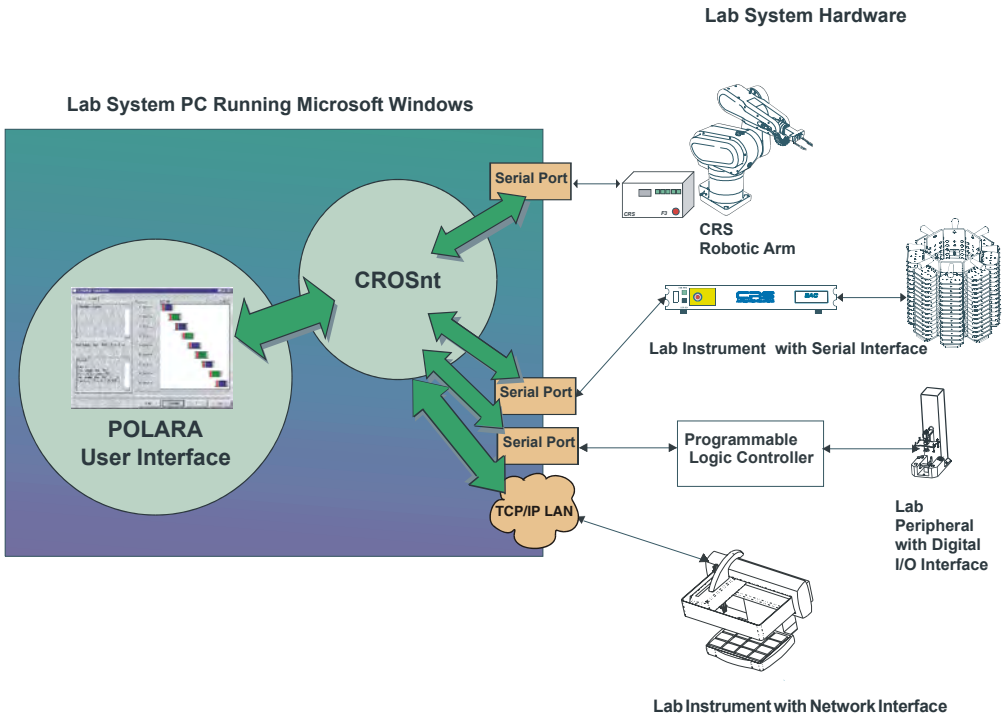


Figure 1-5. In a POLARA system, the POLARA Windows application uses CROSnt to interface to lab system hardware.

What Happens During a POLARA Run

When you click **Start Run** on the POLARA scheduler, the software performs the following operations:

1. POLARA compiles the .seq schedule file into a CROSnt program called the *dispatcher*.

2. POLARA opens the Monitor, the window you use to control the run.

3. POLARA configures CROSnt for the lab instruments specified in the current profile and launches it.

4. CROSnt launches the following programs:

    - **simsockd**, which provides communications with the robot controller (if a robot is used to move containers)

    - **utility servers**, such as the servers that control programmable logic controllers

    - **robot administration daemons (RAD)**, which manage requests from the instrument servers to move the container transport

    - **Run Time Manager (RTM)**, which supervises the execution of the instructions in the dispatcher

    - **the dispatcher**, which is the CROSnt application compiled in step 1.

5. The dispatcher starts the instrument servers, the programs that control each instance of instrument or peripheral in the current profile.

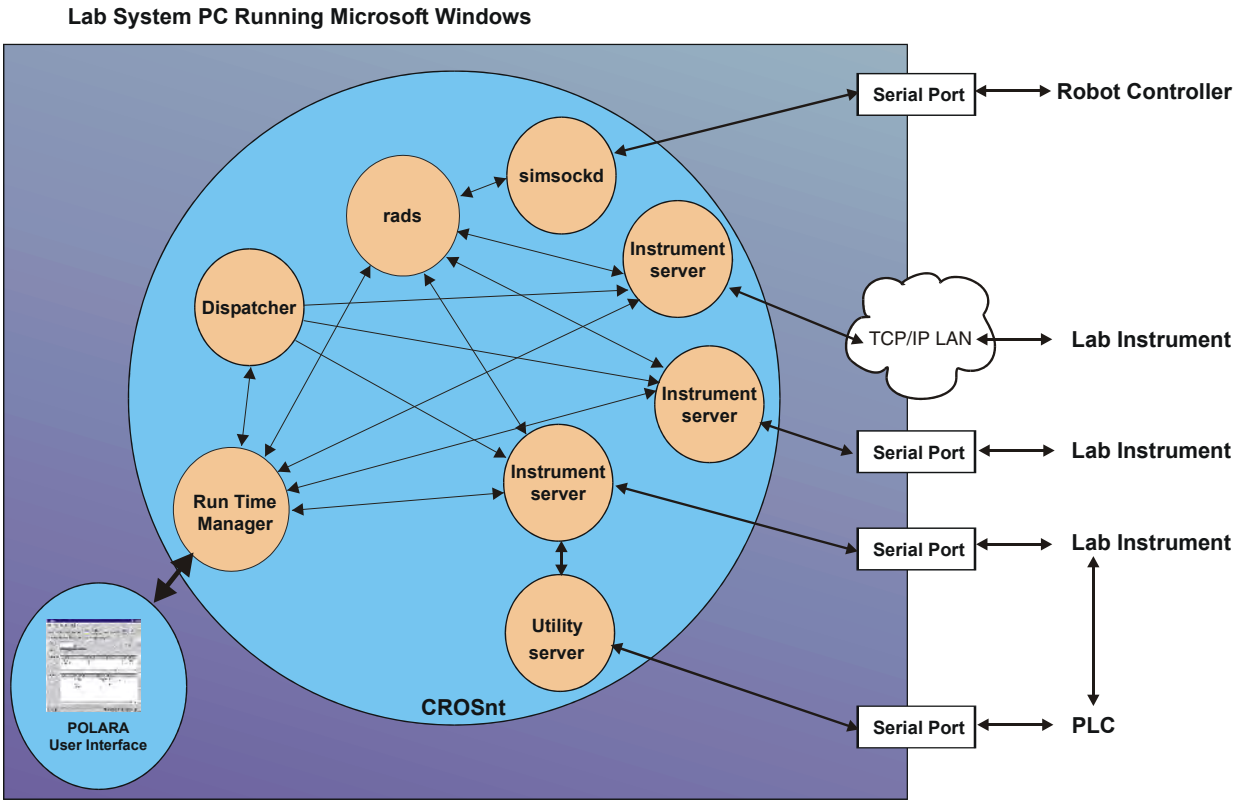


Figure 1-6. During a run, the dispatcher issues commands to each instrument server to move and process the containers.

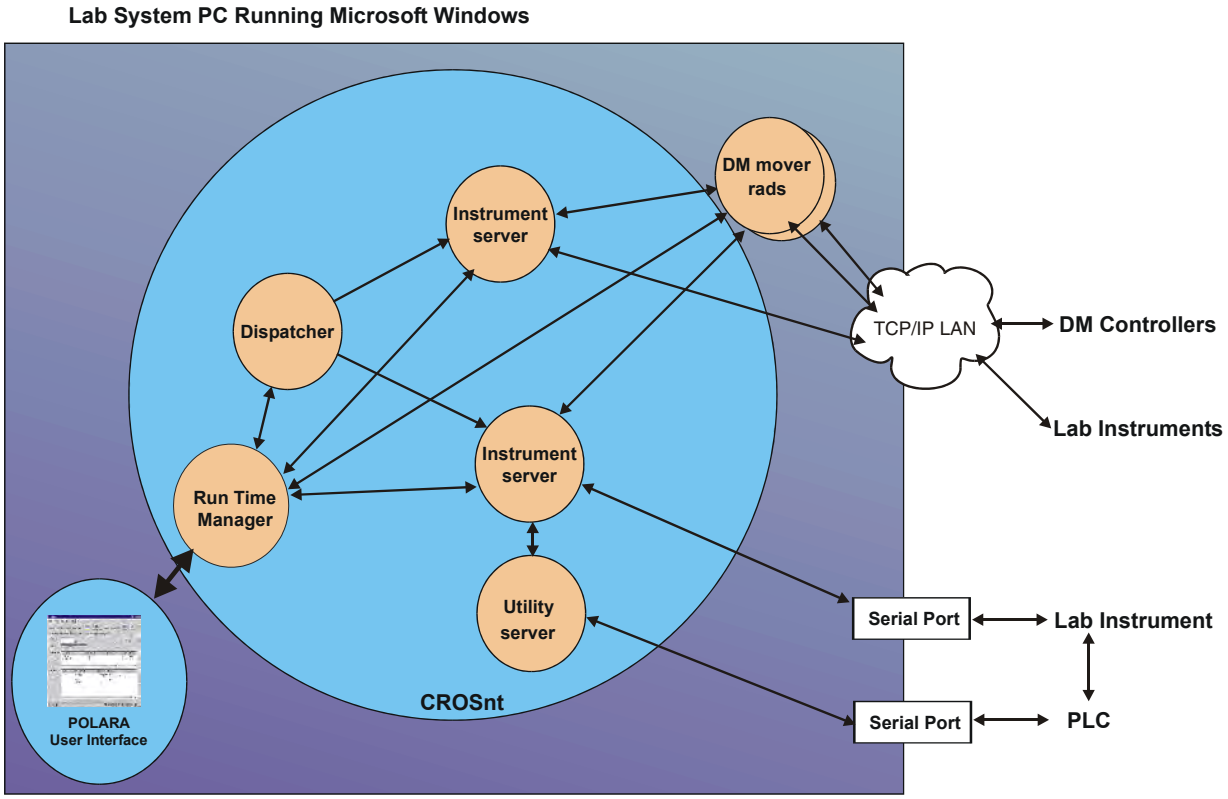


Figure 1-7. In a POLARA lab system that uses DM movers (including the CRS VAL), the instrument servers move containers by issuing commands to the DM controllers via the DM mover rads running in Windows.

As each server initializes, POLARA displays its status in the Monitor.

6. When you click **Start/Continue** in the Monitor, POLARA checks the status of the container transport. If the transport is ready, the RTM permits the instrument servers to start executing the instructions in the dispatcher.

7. If any server encounters a significant problem performing an operation, the RTM blocks the servers from executing instructions and displays a dialog box requesting intervention. After you resolve the problem, the RTM unblocks the servers, enabling them to resume executing the dispatcher's instructions until the run is complete.

8. When you shut down the system, POLARA terminates CROSnt and closes all log and data files.

What You Need to Know to Use POLARA

What you need to know depends on which type of POLARA user you are.

What All Users Need to Know

All users of a POLARA lab system must be familiar with the following subjects:

- Using a personal computer running Microsoft Windows 2000 or XP
- Basic Thermo LACI container transport system features and operations, including the following:
  - Safety features, including use of the emergency stop buttons
  - Powering up
  - Shutting down
  - Recovering from minor container transport error conditions, such as mover collisions
- POLARA operator features and operations, as described in the following chapters:
  - Chapter 2, Safety
  - Chapter 6, Setting up and Performing a Run
- Safe operation of each instrument or peripheral in the lab system.

Note All users of a POLARA system must take POLARA Operator training from Thermo LACI. See Training in the Preface for more information. ▲

What Researcher Users Need to Know

In addition to the subjects that *all* users of a POLARA lab system must know, researcher users must be familiar with the following topics:

- The protocols they wish to automate
- The unit operations provided by the instruments their processes require

Note The unit operations provided by each instrument are described in the instrument interface guides supplied with the lab system. ▲

- POLARA researcher features and operations, as described in the following chapters:
  - Chapter 3, Using Workspaces and Profiles
  - Chapter 4, Working with Methods
  - Chapter 5, Creating and Optimizing Schedules

Note Researcher users of a POLARA system must also take POLARA Researcher training from Thermo LACI. See Training for more information. ▲

The POLARA Main Window

The POLARA main window provides a menu bar and a toolbar that allow you to select workspaces, profiles, methods, and schedules.

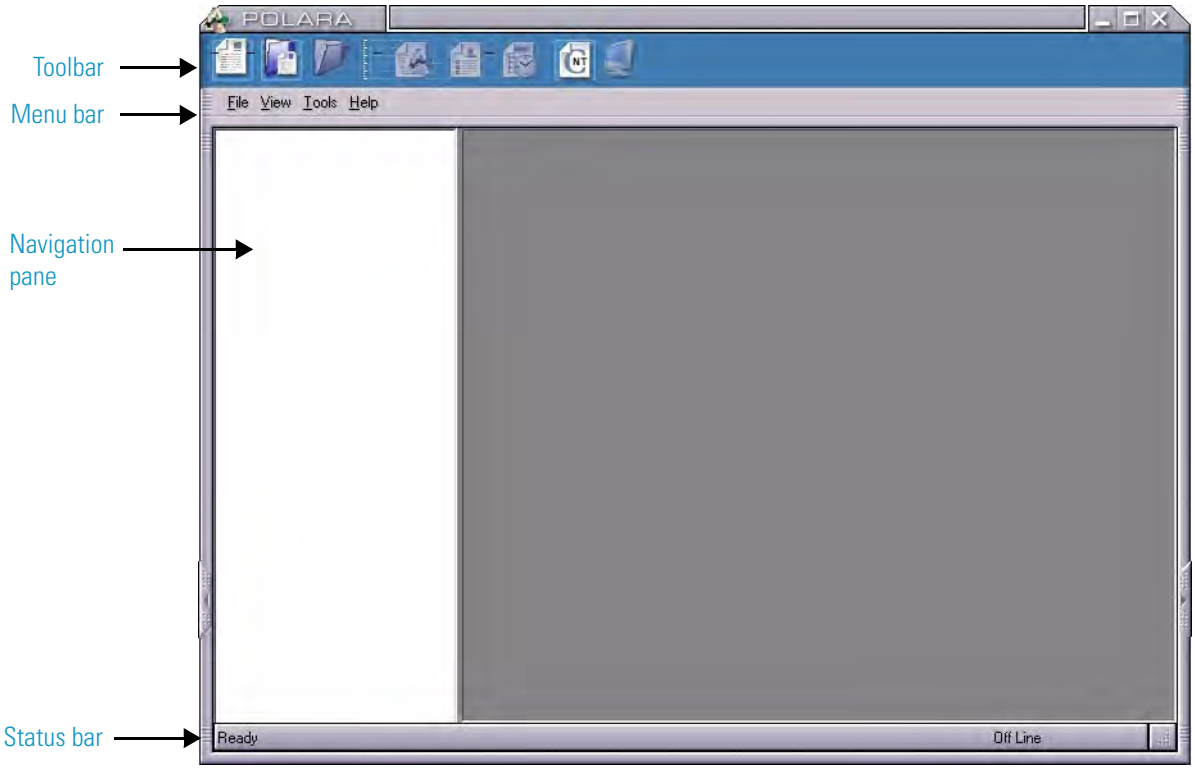


Figure 1-8. The POLARA main window

Using the Menu Bar

The menu bar allows you to perform the following tasks:

- **File**: Open a workspace, close a workspace, install instrument components, and exit POLARA.
- **View**: Adjust the appearance of the POLARA main screen, show/hide the toolbar, status bar, and module list, and open the **Robot Start Location** dialog box.
- **Tools**: Open various windows or wizards to enable options, monitor POLARA, and save run data for trouble shooting.
- **Help**: View information about this release of POLARA. and access on-line help.

Using the Toolbar

The toolbar buttons allow you to perform the following tasks:

Open Workspace: Open a workspace.

Close Workspace: Close a workspace.

Start CROSnt: Start CrosNT.

Bring Run Monitor to Front: Bring the Monitor window to the front. You use this button only when using the Monitor. For details, see "Using The Monitor Window" on page 6-5.

Using the Navigation Pane

When you open a workspace, the Main Window shows a tree view, or hierarchy, of the profiles, methods, and schedules.

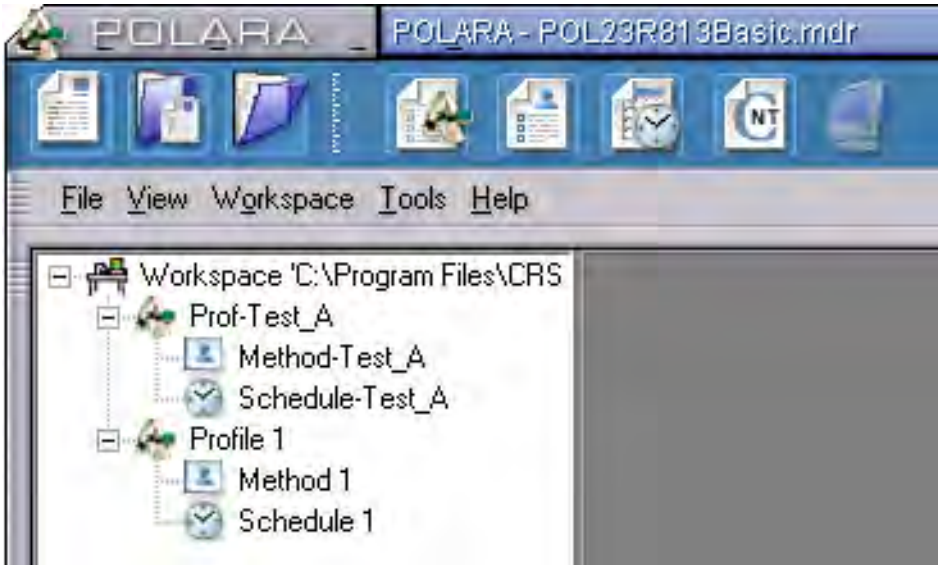


Figure 1-9. The POLARA navigation pane

To hide or show a profile in the navigation pane, you click the plus and minus signs beside the profile, just as you would in a Windows folder list.

To open a profile, method, or schedule, you click the name of the profile, method, or schedule.

Using the Status Bar

Read the text in the status bar to get information about the current operation.

Where to Go From Here

All users must proceed to Chapter 2, Safety to learn how to operate a POLARA lab system safely.

Chapter 2 Safety

Before performing any tasks with POLARA, you must be fully aware of any potential dangers presented by system components.

This chapter provides an overview of hazards that may be associated with your system and explains the safety measures that exist to protect you.

WARNING Because each POLARA system is different, this user guide cannot provide a complete description of all potential hazards. Make sure that you review and understand the safety concerns associated with each instrument and peripheral in your system. ▲

This chapter covers the following topics:

- "Built-in Safety Features" on page 2-2, which provides an overview of the safety features of POLARA lab systems
- "Before Using POLARA" on page 2-3, which describes what subjects you must know to safely use a POLARA system
- "Ensuring Safe Use of The POLARA System" on page 2-4, which provides an overview of the safe operation guidelines all users of POLARA must follow
- "Triggering an E-Stop" on page 2-5, which describes how to trigger an emergency stop on a POLARA system, and how to recover the system from it
- "Understanding POLARA Warnings and Errors" on page 2-7, which describes the two types of alerts POLARA issues
- "Understanding the Beacon" on page 2-8, which describes the beacon lights and how to interpret them
- "Recovering the System from a Flip Mover Collision" on page 2-9, which describes how to deal with a Flip Mover collision

Built-in Safety Features

The following safety features are standard for all POLARA Systems:

- Emergency Stop (E-Stop) buttons enable you to immediately halt container transport motion.
- The run-time Monitor window displays information during the progress of a run, including error messages and warnings.
- If POLARA detects that operator intervention is required during a run (for instance, if a non-standard container needs to be repositioned or removed), it suspends the run and alerts the operator.

Your system may also include some or all of the following optional safety features:

- A three color beacon provides a visual indication of system status.
- Universal E-Stop buttons halt multiple devices in the laboratory system.
- Safety guarding may either partially or completely enclose the system to prevent access during a run.
- An Uninterruptible Power Supply (UPS) can be configured to ensure that the system is halted in an orderly fashion if power is suddenly lost.
- Sealed environments protect operators from hazardous substances inside the laboratory system enclosure.
- Door interlocks are integrated with the system E-Stop chain to ensure that the robot and/or instruments are halted when a door is opened.
- An end-of-arm load limiter protects equipment from collision damage. The load limiter is a pneumatic safety-release device used with Thermo's more powerful CRS F3 robot. It is located between the gripper and the tool flange. If the load limiter detects a force greater than its pre-set threshold, it triggers an E-Stop and halts the arm.
- Automatic telephone paging can be used to remotely notify system personnel of error conditions that occur during a run.

Before Using POLARA

Before using a POLARA system, you must be trained in its safe use by Thermo LACI.

You must also have a thorough understanding of the following subjects:

- The operation of the POLARA software
- How to develop profiles, methods and schedules
- How to perform a run and monitor its progress
- How to safely operate the container transport
- How to safely operate each of the instruments on the table
- How to respond to warnings and error messages
- How to use E-Stop buttons
- How to recover the system after an E-Stop is triggered or a container mover error occurs

The user guides supplied with your system provide the information you need to understand these subjects. Read all of it before using a POLARA lab system.

Ensuring Safe Use of The POLARA System

When using the POLARA system, you must observe the following safety guidelines:

- Do not allow untrained persons to control the POLARA system.
- Any person who is authorized to operate the container transport must be fully trained in container transport safety and the use of container transport motion commands.
- Train any person who uses the laboratory system in the dangers and risks associated with each instrument and with the laboratory system as a whole.
- If the biological or chemical process that you are investigating involves hazardous materials, ensure that operators are aware of the dangers and take appropriate measures to minimize risks.
- Any person who can approach the container transport hardware must be made aware of the following rules:
    - Never place any body part within the reach of a mover when mover power is on. The mover can move unexpectedly at high speeds.
    - Do not touch automated components when the system is running.
    - Never operate the system if any substance may have penetrated the container transport hardware.
- If the system is being serviced or modified, or is acting abnormally, ensure that it is clearly labeled as “Out of Service” to prevent accidental use.
- Following any change in lab system hardware, thoroughly test the changes and perform a sample run before making the system available for use.

Triggering an E-Stop

In case of emergency, operators can quickly halt container transport motion by triggering an E-Stop.

WARNING Other instruments may not stop moving when you trigger an E-Stop. Before you operate your POLARA system, ensure that you know how each instrument will respond to an E-Stop. ▲

To trigger an E-Stop

- Strike any E-Stop button.

The container transport halts. The beacon light flashes red and yellow, indicating that the system has been paused and is waiting for manual intervention. POLARA displays a dialog box on screen that reads: "The operation could not be completed: Arm is not powered - please turn on arm power and move robot to a safe location."

You then have the option to abort the run or attempt recovery.

To recover from an E-Stop

1. Inspect the system components. If any are physically damaged, recovery of the run is not possible. Click Abort in the message box presented by POLARA.

2. Verify that it is safe for the run to continue:

   - Clean up any spills. If any containers have been misplaced or dropped, replace them as needed.

   - Make sure that the emergency stop has not compromised the integrity of the run. Some processes are extremely sensitive to temperature changes or contaminants. If necessary, take note of any samples which might have been affected.

   - Make sure that the container transport work space is free of obstructions.

3. Twist the E-Stop button to reset it, or close the E-Stop device that triggered the stop.

Note E-Stop devices can include interlocked doors or any other devices that are connected to the plate transport E-Stop circuit. Devices connected to an E-Stop chain must be manually reset after an E-Stop has been triggered. ▲

4. If the system uses a DM container transport, bring it back to a ready state:

    a. Press the system Reset button.

    b. Press and hold the master Motor Power ON button until it turns green.

5. If the system uses a CRS articulated robot, take the following steps to move to a safe position:

    a. Choose a safe position that is close to the instrument that the robot is moving towards. This does not have to be a taught location: you can recover from any position.

    b. Press the Arm Power button on the C500C controller to restore power to the arm.

    c. If arm power cannot be restored, check again to make sure that all E-Stop devices have been correctly reset. If you still cannot apply arm power, recovery may not be possible. Contact your system administrator for assistance.

    d. Using ash, move the robot to the nearest safe position. You should choose a safe position that is close to the instrument that the robot is moving towards. This does not have to be a taught *safeloc* location: using an appropriate stance, you can recover from any position which is known to be safe.

6. When the DM transport is back online or the robot is in a safe position, perform one of the following steps in POLARA to continue the run:

    - If an error message box is shown on screen, click Retry (or any other option that will allow you to proceed). The run continues.

    - If a Robot Administration Daemon message box is shown, click Continue in the message box to allow the run to resume.

Understanding POLARA Warnings and Errors

During runs, POLARA issues warnings and error messages when it encounters a condition that should be noted or corrected.

For details on how to respond to specific warnings and error messages, see Chapter 6, Setting up and Performing a Run.

Warnings

POLARA issues a warning when it encounters a condition that should be noted and reported but is not serious enough to warrant suspending the run. It alerts operators to a warning in the following ways:

- The beacon turns yellow.
- A warning appears in the Messages tab of the Monitor window and the Monitor warning status indicator turns yellow.

Error Messages

POLARA will suspend a run when it encounters an error that requires intervention. It alerts operators in the following ways:

- The beacon flashes yellow.
- Depending on the type of error, a message box may appear on the screen.
- An error message appears in the Messages tab of the Monitor window and the Monitor error status indicator turns red.

Understanding the Beacon

The beacon is a lab system component that displays the status of the run. As you perform a run, look at the beacon to check the run's status.

The beacon uses three colored lights that are lit individually or in combination as follows*:*

Green

The green beacon light indicates the system is running, as follows:

- Solid green: The run is proceeding normally and no warnings have been issued.
- Solid green, solid yellow: The system is running, but warnings have been issued. Depending on the type of warning, the run may become suspended if corrective action is not taken.
- Solid green, flashing yellow: The system is running, but an error occurred that has not been cleared.

Yellow

The yellow beacon light indicates something is wrong, as follows:

- Solid yellow: A warning has occurred.
- Flashing yellow: An error has occurred and manual intervention is required.

Red

The red beacon light indicates the system is not running, as follows:

- Solid red: The run has completed.
- Flashing red: The system is in a suspended state.
- Flashing red, flashing yellow: An error has occurred and the run was suspended. Once corrective action is performed, the run can be continued.

Recovering the System from a Flip Mover Collision

If a mover collides with an object, its controller removes power to the mover. As soon as power is cut, the mover's brakes engage.

When POLARA detects that a mover is no longer responding to commands, it suspends the run and displays a message to the operator.

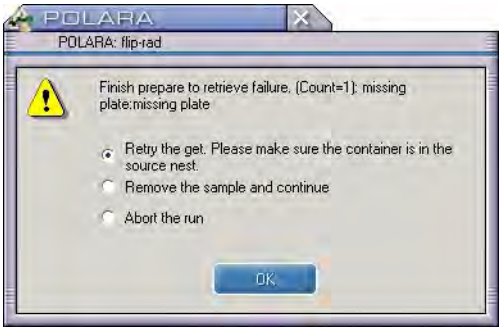


Figure 2-10. Error message POLARA displays after a mover collision

To recover from a mover collision

1. Inspect the mover and other system components for physical damage.

2. If any physical damage has occurred, click Abort the run in the POLARA message window to abort the run. Contact Thermo LACI Customer Support as described in For More Information in the Preface.

3. If a container has become misaligned in the mover gripper, take the following steps:

   a. Open the System utility on the system computer.

   b. In the System menu, choose View to open the System view window.

   c. With a trained user standing by to catch the released container, select the mover holding the misaligned container, click Clear Error, and click Empty Mover to release the container.

   d. Replace the container in the nest described in the retry option in the POLARA message box or select Remove sample and continue.

   e. Click OK to continue the run.

Where to Go from Here

To start defining protocols, see Chapter 3, Using Workspaces and Profiles.

To set up and perform a run, see Chapter 6, Setting up and Performing a Run.

Chapter 3 Using Workspaces and Profiles

This chapter provides an overview of how to use POLARA workspaces and profiles. It covers the following topics:

- "About POLARA Workspaces" on page 3-2, which introduces the POLARA workspace and describes how to open a workspace for use.
- "About POLARA Profiles" on page 3-4, which introduces POLARA profiles and describes how to add and configure instrument interfaces and containers
- "Creating a New Profile" on page 3-5, which describes how to create a new profile
- "Opening a Profile" on page 3-6, which describes how to open a currently existing profile
- "Adding and Modifying Instrument Instances" on page 3-7, which describes how to add and remove instrument instances to a profile and how to modify their properties
- "Copying a Profile" on page 3-9, which describes how to create a copy of an existing profile
- "Deleting a Profile" on page 3-10, which describes how to delete a profile
- "Printing a Profile" on page 3-11, which describes how to print a list of a profile's instrument instances and their properties
- "Closing a Profile" on page 3-12, which describes how to close a profile
- "Setting Container Instance Properties" on page 3-13, which describes how to modify the properties of container instances

About POLARA Workspaces

A workspace is POLARA's representation of a laboratory system. Created by the system administrator, the .mdr workspace file contains all of the instrument settings, profiles, methods, and schedules that you use to perform tasks with the system.

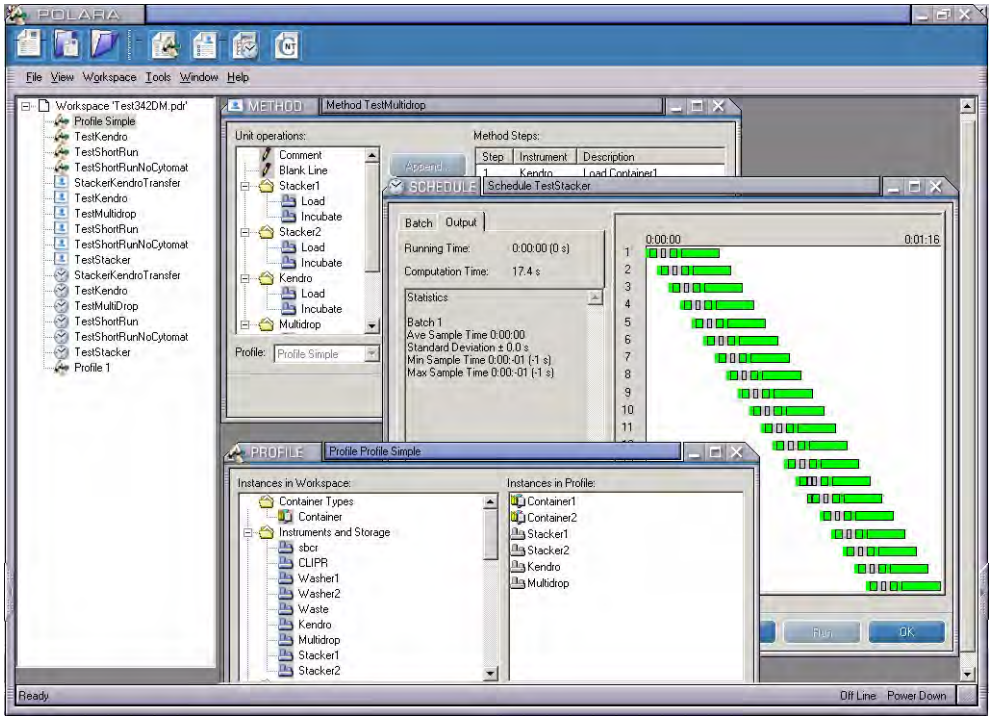


Figure 3-11. The workspace contains all instrument settings, profiles, methods, and schedules.

Opening a Workspace

In order to configure lab system settings or define profiles within POLARA, you must open a workspace.

To open a workspace

1. In the POLARA File menu, choose **Open Workspace**. POLARA opens a file browser.

2. Navigate to the workspace directory.

   Note By default, the workspace directory is C:\Program Files\CRS Robotics\Polara. Your system administrator may have created a different subdirectory for workspaces. ▲

3. Select a HSDM workspace (.pdr) or a robot workspace (.mdr) file and click **Open**.

Profiles, Methods, and Schedules

Within an open workspace, you define profiles, methods, and schedules to completely specify how a run is carried out by the laboratory system.

Figure 3-12. POLARA icons for profiles, methods, and schedules

- A **profile** is the POLARA representation of the group of instruments and containers required to carry out a biological or chemical protocol. A workspace can have more than one profile for different types of runs carried out on the system. For example, you might set up similar profiles with different container types for small and large sample runs.

  Tip Because only the instruments in the profile are active during a run, instruments that are not in the profile can be used for manual tasks while the system is running. ▲

- A **method** is a list of steps that defines the biological or chemical protocol that you want followed on an individual sample.

- A **schedule** is the instructions the lab system follows to process the required number of samples through the selected method(s)

About POLARA Profiles

The workspace defines all of the components that can be used with your laboratory system, and the physical settings required to enable POLARA to control those components. A profile defines which of these components are available to methods and schedules during a run and what their variable settings should be.

As a subset of the workspace, a profile performs the following functions:

- Defining which instruments and containers are available to a method
- Defining which nests are available to containers in a method
- Providing access to instrument parameters that can change from run to run, such as incubator temperature or the type of container that will be used

In general, profiles enable you to manage instrument and container settings on a run-by-run basis. For example, if you want to vary the temperature of an incubation step between runs, you create a separate profile for each temperature value you want to use. Then you create copies of the method, with each copy using a profile with a different temperature setting for the incubator. Finally, you create and run a separate schedule for each method.

Note The **Set for Teaching** button in each profile is provided only as a convenience for administrators for instrument integration testing and is therefore not described here. ▲

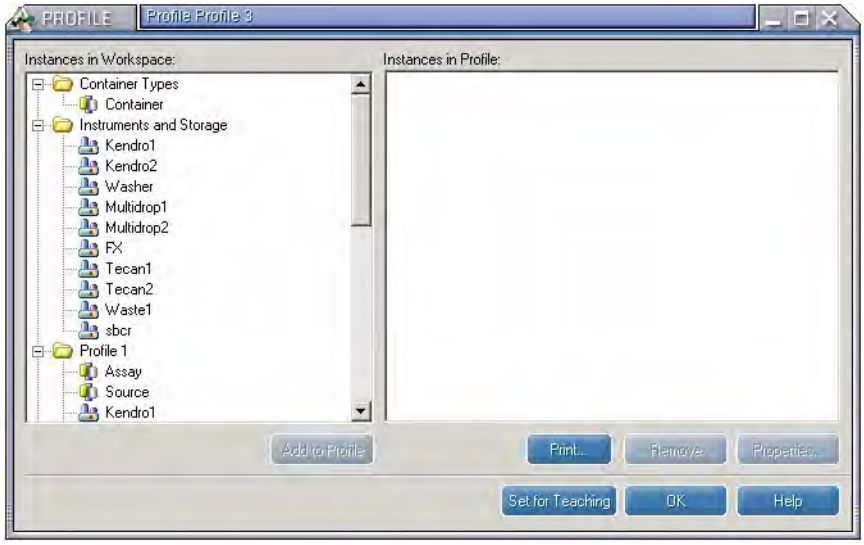
Creating a New Profile

To create a new profile, you must be in an open workspace.

To create a new profile

1. From the POLARA **Workspace** main menu, choose **New Profile**. POLARA opens the **Add new profile** dialog box.

2. Enter a name for the profile and click **OK**. POLARA creates a blank profile.



At this point, you can either proceed to add instances to the profile, as described in "Adding and Modifying Instrument Instances" on page 3-7, or you can click **OK** to close and save the profile.

Opening a Profile

You can only open a profile when a workspace is open.

To open a profile

1. Open the workspace containing the profile you want to open.

2. Click on the name of the profile or click the profile icon in the left pane. The profile window opens in the right pane of the POLARA window.

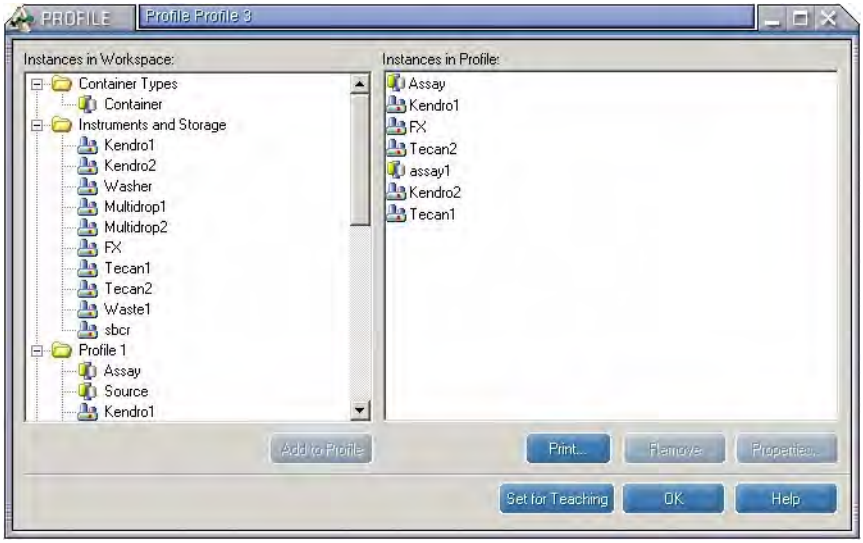


Figure 3-13. The profile editor

All the instances available for use in profiles are shown in the pane called **Instances in Workspace**. All of the container and instrument instances that have been added to the profile are shown under **Instances in Profile**. A container or instrument must be listed here for you to use it in a method.

Tip By clicking the maximize button, you can enlarge the profile to use the full space in the right pane. ▲

Adding and Modifying Instrument Instances

To define which instruments and containers are in a profile and to define their run-time settings, or *properties*, you use the **Profile** editor.

Adding an Instance to a Profile

To make an instrument's unit operations available for use in methods, you must add an instance of the instrument to a profile.

Note You cannot add an instance of an instrument or peripheral until you have added one instance of a container to the profile. For details, see "Setting Container Instance Properties" on page 3-13. ▲

To add an instance to a profile

1. In the **Instances in Workspace** pane of the **Profile** editor, select the desired instance from **Instruments and Storage** or another profile, and click **Add to Profile**.

   Tip If you add an instance of a instrument or container from another profile, POLARA retains the instances' properties. However, if you add an instrument instance from another profile, you must ensure that the container instances it uses in the other profile are also added. ▲

   POLARA displays the properties for the instance.

   Note For details about the properties of an instrument, see its interface guide. ▲

   Tip You can also add an instance by double-clicking its name in the **Instances in Workspace** pane. ▲

2. If you want to make future instances of this instrument use the current properties as defaults when they are added to a profile, click **Set as Default**.

3. When you have finishing specifying the properties for this instance, take one of the following steps:

   a. Click **OK**. POLARA adds the new instance to the **Instances in Profile** pane.

   b. Click **Cancel** to return to the profile without adding the instance.

Viewing and Editing Instance Properties

Each instance of an instrument or peripheral in a profile has a set of properties that define its run-time behavior.

To view or edit instance properties in a profile

1. In the **Instances in Profile** pane of the profile, select the instance and click **Properties**. POLARA displays the properties for the instance.

   Note For details about the properties of a container, see "Setting Container Instance Properties" on page 3-13. ▲

   Note For details about the properties of an instrument, see its interface guide. ▲

   Tip You can also view an instance's properties by double-clicking its name in the **Instances in Profile** pane. ▲

2. If you want to make future instances of this instrument use the current properties as defaults when they are added to a profile, click **Set as Default**.

3. When you have finishing viewing or editing the properties for the instance, take one of the following steps:

   a. Click **OK**. POLARA saves the instance properties to the profile.

   b. Click **Cancel** to return to the profile without changing the instance's properties.

Removing Instances From a Profile

Remove an instrument instance from a profile when it is no longer required.

To remove an instance from a profile

- In the **Instances in Profile** pane of the profile, select the instance and click **Remove**.

Copying a Profile

You can duplicate an existing profile using the **Copy** command.

To copy a profile

1. In the left pane of the workspace editor, right-click on the profile name and choose **Copy** from the context menu. POLARA opens a New Profile window.

2. Enter a name for the new profile and click **OK**. POLARA creates a duplicate of the original profile.

Deleting a Profile

You can delete an existing profile using the **Delete** command.

To delete a profile

1. In the left pane of the workspace editor, right-click on the profile name and choose **Delete** from the context menu. POLARA prompts you confirm the delete operation.

2. Click **OK** to delete the profile or **Cancel** to abort the delete operation.

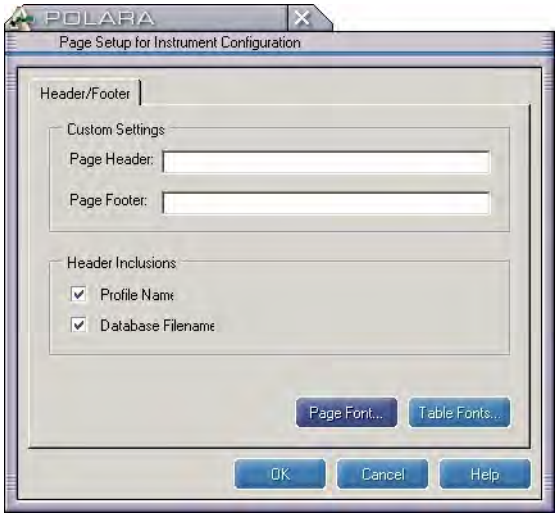
Printing a Profile

When printing a profile, you can choose which fields to print, enter text for a header or footer, and customize the fonts used in the printed copy.

To print a profile

1. Open the profile you want to print.

2. In the profile, click **Print**. A page setup dialog box opens:



3. Use the page setup dialog box to customize the printed output:

   - Enter text for the header or footer of the page and select whether the profile name and database filename are to be printed.

   - Click **Page Font** to select a different font for the header and footer information. Click **Table Fonts** to select a different font for the printed list of instruments.

4. Take one of the following steps:

   - To send the profile to the printer, click **OK**.

   - To cancel the printing operation, click **Cancel**.

Closing a Profile

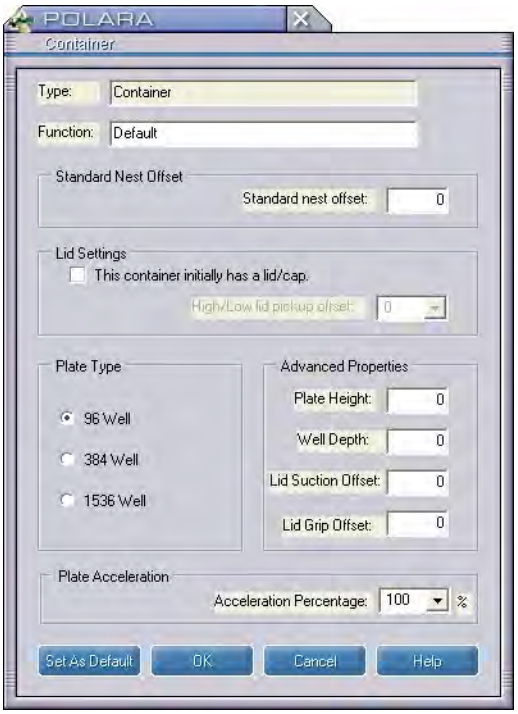
When you have finished viewing or editing a profile, you can close it.

To close a profile

- Click **Close** to return to the workspace editor.

Setting Container Instance Properties

A container instance in a profile has the following properties:



- **Function** is the unique name of this container in this profile.

- **Standard Nest Offset** lets you specify that the container should be gripped higher (or lower) than the standard taught position. To grip lower on the plate, specify a negative value in millimeters; to grip higher, specify a positive value in millimeters. If a standard nest offset applies, the system administrator can provide you with the exact value that should be used here.

- **Lid Settings** let you specify whether the container has a lid, and if so, whether it should be gripped at a different height when the lid is on. To grip lower on the plate, specify a negative value in millimeters; to grip higher, specify a positive value in millimeters.

  Note If you check the **Lid Settings** box, you must add a lidder to your profile to enable delidding. ▲

- **Plate Type** lets you specify the number of wells in the plate. POLARA does not use this value itself; it simply passes it through to the instrument servers.

- The settings under **Advanced Properties** are not supported for most instruments. Refer to your instrument interface guide for more detail.
- Plate Acceleration lets you specify the rate of acceleration or speed used to transport the plate, as a percentage of maximum possible transport acceleration or robot speed. Reducing the rate of acceleration reduces the risk of spilling contents when the plate is moved. The degree of risk also depends on the size of the wells (smaller wells offer less risk) and the viscosity of the contents (higher viscosity offers less risk).

To change and save the container instance properties

1. Change the property values as required.
2. If you want to make future instances of this instrument use the current properties as defaults when they are added to a profile, click **Set as Default**.
3. Take one of the following steps:
   - Choose **OK**. POLARA saves the instance properties to the profile.
   - Choose **Cancel** to return to the profile without changing the instance's properties.

Where To Go From Here

Once you have created profiles containing the instruments you need for your methods, you can begin creating methods. For details, see Chapter 4: "Working with Methods".

Chapter 4 Working with Methods

This chapter describes how to create and set up the methods used in a POLARA run. It covers the following topics:

- "About POLARA Methods" on page 4-2 describes how methods are used to automate biological or chemical procedures.
- "Creating a Method" on page 4-5 describes how to add a new method to a workspace.
- "Opening a Method" on page 4-6 explains how to open an existing method.
- "Adding and Modifying Steps" on page 4-7 describes how to add, remove, and change the order of steps within a method, and how to edit step properties.
- "Printing a Method" on page 4-10 describes how to print the steps in a method.

About POLARA Methods

In POLARA, a *method* is a series of *steps* that defines the biological or chemical protocol used to process one *sample*.

About POLARA Samples

A POLARA *sample* is the smallest set of containers needed for one iteration of a method. For example, if a method replicates a mother plate to four daughter plates, the sample consists of five containers.

On systems using CRS F3 robots, POLARA also supports stacks of microtitre plates, such as those used by instruments with Titertek stack feeder mechanisms. In this case, the container defined in the profiles that have instances of such instruments refers to the entire stack. The properties of method steps for such instruments contain settings for *sub-containers*, which refer to the individual microtitre plates within the stack. The POLARA scheduler also provides settings for the number of sub-containers per container, so that you can schedule a variable number of microtitre plates within each stack.

About POLARA Steps

A *step* is a line in a method that invokes a *unit operation*. Each instrument in a POLARA lab system provides one or more unit operations.

Note For details on the unit operations provided by an instrument, refer to the instrument's interface guide. ▲

Some commonly used steps in POLARA methods include the following:

- **Load**, which you usually use to tell POLARA where to pick up a container at the beginning of a method and where to place it at the end of the method
- **Incubate**, which tells POLARA how long a container should rest in the instrument. While a container is incubating, POLARA may be able to move other containers through the method, providing a degree of parallel processing.

  Note Incubate is used generally for hotels, carousels, and incubators. ▲

- **Transfer**, **which moves material from one container to another**
- **Read**, which tells POLARA to get data from an instrument such as a fluorescence or colorimeter reader.

About Step Properties

Each step in a method has *properties* you set to define its attributes and refine its behavior. For example, the hotel incubate step has the following properties:

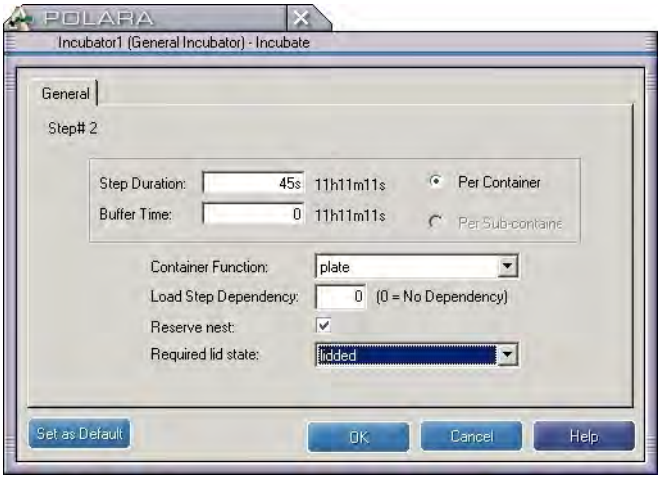


Figure 4-14. The properties of the incubate step for a hotel or other room-temperature incubator

- **Step Duration** estimates the time required for the instrument to perform the unit operation. This time does not have to exactly match the actual time taken by the instrument during a run, but the more accurate it is, the more accurate the schedule will be in its time estimates. To make step duration estimates accurate, see "Optimizing Step Time Estimates" on page 5-30.

- **Buffer Time** defines the maximum time the scheduler is permitted to leave the container in the instrument after the instrument has performed the unit operation. This enables you to limit the amount of time a container can spend in an instrument (often important for incubate operations), but increases the likelihood of creating a bottleneck that the scheduler cannot overcome.

- **Container Function** specifies which container type to use in the step.

- **Load Step Dependency** prevents the incubate step from executing until the specified previous step has completed.

- **Reserve Nest**, when set, prevents the container transport from putting another container into the nest used by this step's container. You set this property so that you can return the container to this nest later in the method.

Note If you reserve the nest in this step, you must reserve it in every subsequent step that places the same container in the same instrument; otherwise, the scheduler will report that it failed to find a solution. ▲

Note You must reserve the nest if you want to use this method for a batch added to a run in progress, even if your method does not require the container to return to a specific nest in the instrument. ▲

- **Required Lid State** sets whether the container must be lidded or not in the incubate step.

Note For details on the properties of a step, refer to the interface guide for the instrument that performs the step. ▲

Shadow Instruments and Implicit Steps

POLARA's scheduler automatically includes some steps in a schedule, freeing you from having to explicitly include these steps in your method.

For example, if the containers in your method are lidded at the beginning of the method, but must be unlidded when placed in a dispenser, the POLARA scheduler will automatically include a delidding step before the dispensing step in the schedule it generates for the method. You do not have to include in your method before the dispense step an explicit delidding step; you simply set the dispense step's **Required Lid State** property is set to "unlidded".

POLARA refers to these implied steps as *shadow operations* and to the instruments that perform such operations as *shadow instruments*. Shadow instruments include the following:

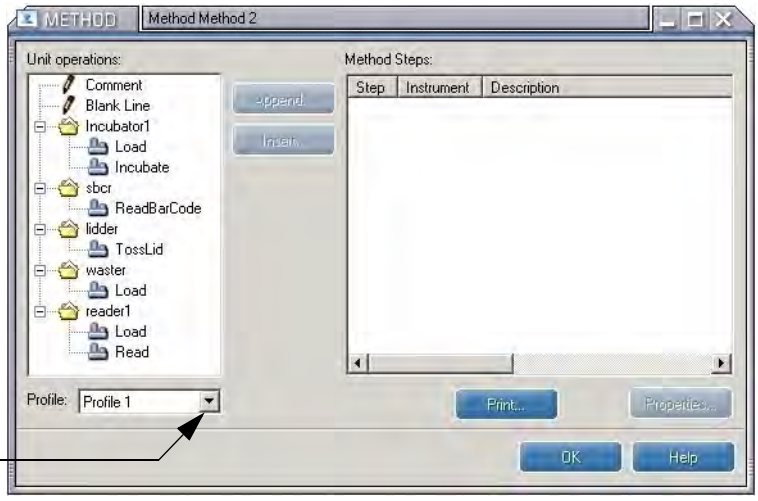
- The **Shadow Lidder with Shelf** removes and replaces lids on microtitre plates.
- The **Shadow Regrip Station** rotates a container by 90°, enabling an articulated robot to switch its grasp of a container: from the long sides of a microtitre plate, for example, to the short sides.

Creating a Method

To create a method, you must be in an open workspace and you must have defined at least one profile.

To create a method

1. In the POLARA **Workspace** menu, choose **New Method** and enter a name for the method. POLARA opens a blank **Method**:



Click in the Profile drop-down box to select a profile.

2. In the **Profile** selection box, choose the profile containing the instruments your method requires. The list of unit operations supplied by the instruments appears in the left pane of the **Method** window.

   You can now start adding steps. For details, see "Adding and Modifying Steps" on page 4-7.

Opening a Method

When you open a workspace, the methods associated with that workspace are shown in the left pane of the POLARA window.

To open a method

- Click the name of the method or click the icon for that method in the left pane of the workspace. POLARA opens the method in the right pane of the workspace.

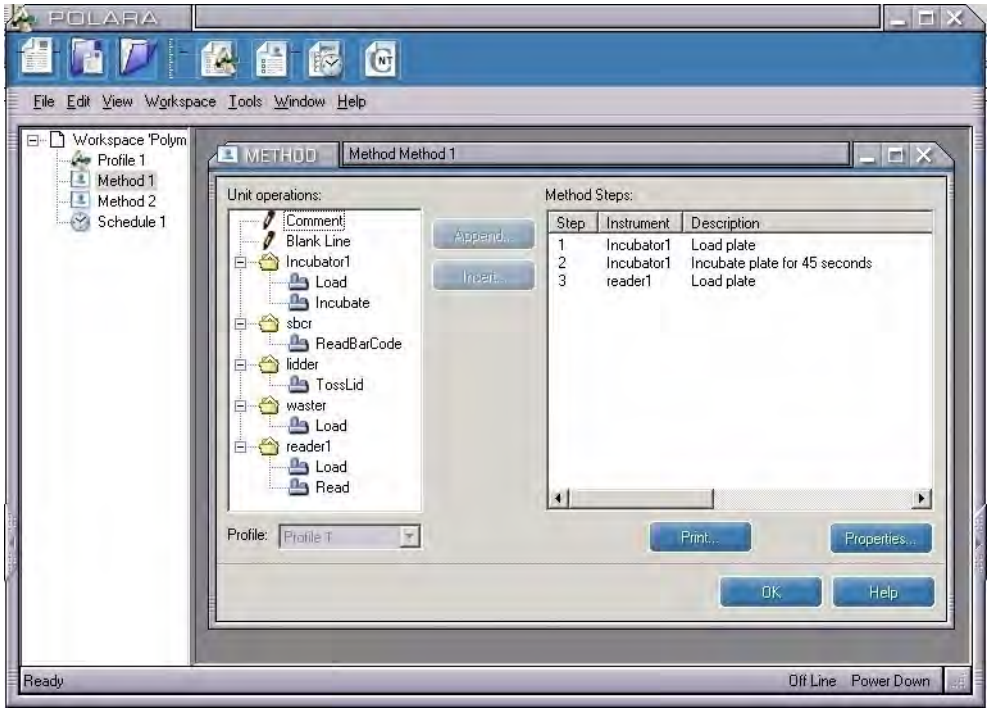


Figure 4-15. The Method window displays a list of unit operations.

To add or modify steps in the method, see "Adding and Modifying Steps" on page 4-7.

Adding and Modifying Steps

You add and modify steps in a method to define the protocol or assay.

Adding a New Step to a Method

When adding a new step to a method, you can use **Append**, which adds the step to the end of the list in the right pane, or you can use **Insert**, which inserts the step just before the current selection in the list.

Note Once a method contains a step, you cannot change the profile on which the method is based. ▲

Note For details on unit operations and their properties, consult the interface guide for the instrument. ▲

Tip The first step in a method should generally be a Load operation. ▲

To append or insert a new step

1. If you want to **Insert** a new step, select the step in the right pane of the **Method** window *before* which you want POLARA to insert the new step.

2. Select an instrument's unit operation from the profile in the left pane of the **Method** window and click **Append** to add a step to the end of the list or **Insert** to insert the step into the list. POLARA displays the properties for the selected unit operation.

3. Modify the properties as required. Refer to the instrument interface guide for details.

4. If you want to make future instances of this step use the current properties as defaults when they are added to a method, click **Set as Default**.

5. Take one of the following steps:

   a. Click **OK**. POLARA appends or inserts the step to the list in the right pane of the Method window.

   b. Click **Cancel** to return to the method without adding the new step.

Viewing and Modifying Step Properties

In order to optimize a schedule, you may need to go back and adjust the properties associated with a step in the method.

To view or modify step properties

1. Select the step in the right pane of the method window and click **Properties**. POLARA opens the **Properties** window.

2. Modify the properties as required. Refer to the instrument interface guide for details.

3. If you want to make future instances of this step use the current properties as defaults when they are added to a method, click **Set as Default**.

4. Take one of the following steps:

   a. Click **OK**. POLARA updates the properties and closes the **Properties** window.

   b. Click **Cancel** to return to the method without adding the step.

Editing Steps

To add or remove steps within a method, or change their order, you can cut, copy, and paste them. You can also cut or copy steps from one method and paste them into another, provided both methods are in the same workspace.

WARNING When pasting steps or changing their order, pooling and step dependencies may be lost or corrupted. Always un-pool instruments and remove step dependencies before pasting steps. ▲

Tip You can select and copy steps using any of the standard Windows mouse and keyboard techniques. For example, you can copy selected steps into the clipboard by pressing CTRL-C and paste them into the destination method by pressing CTRL-V. You can also edit steps inside a method by right-clicking on the step and selecting commands in the context menu. ▲

To cut or copy steps

1. Select the steps.

2. In the **Edit** menu, choose **Cut** to delete the steps or **Copy** to copy without deleting the steps. POLARA copies the steps to the Windows clipboard.

To paste steps to the end of the list

- In the **Edit** menu, choose **Paste Append**. POLARA pastes the steps at the end of the list.

To paste steps into the middle of the list

1. In the list of steps, select the step *before* which POLARA should paste the steps.

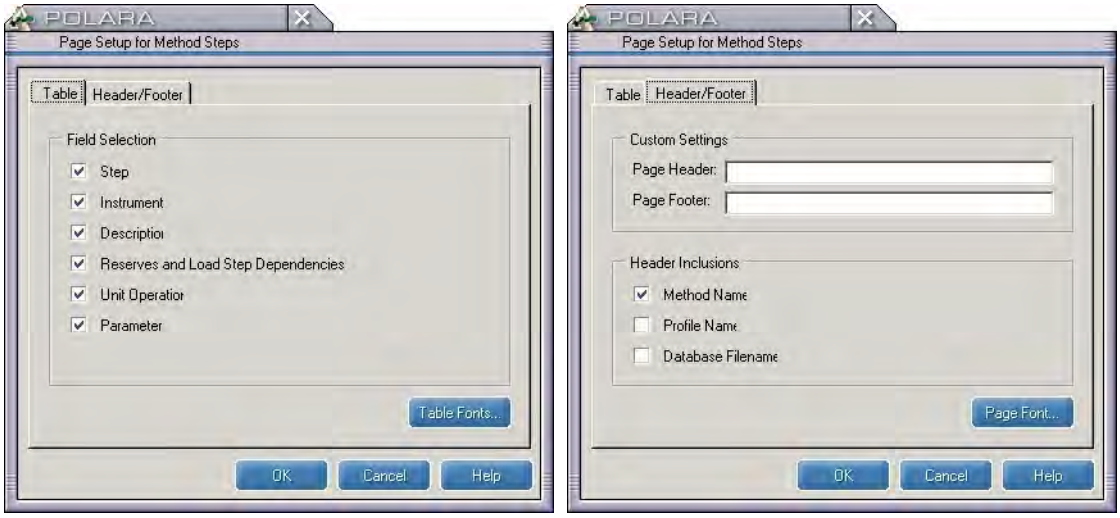
2. In the **Edit** menu, choose **Paste**. POLARA pastes the steps in the list.

Printing a Method

When printing a method, you can choose which fields will be printed, enter text for a header or footer, and customize the fonts used in the printed copy.

To print a method

1. In the **Method** window, click **Print**. A page setup dialog box opens:



2. Use the page setup dialog box to customize the printed output:
   - Select the **Table** tab to choose which fields to print.
   - Select the **Header/Footer** tab to customize the header and footer information in the output.

3. Take one of the following steps:
   - Click **OK** to send the method to the printer.
   - Click **Cancel** to abort the printing.

Where To Go From Here

Once you created your methods, you can create schedules to process samples with them. For details, see Chapter 5, Creating and Optimizing Schedules.

Chapter 5 Creating and Optimizing Schedules

The POLARA schedule controls how multiple samples are processed by the system resources. A well-designed schedule yields increased throughput while optimizing sample uniformity. This chapter provides an overview of how to build a schedule and basic optimization tips to help you create more efficient schedules for your system. It contains the following topics:

- "Developing a Schedule" on page 5-3 outlines the basic steps involved in creating a POLARA schedule.
- "Using The Schedule Window" on page 5-4 provides an overview of the features of the **Schedule** window, which you use to define, optimize, and generate schedules.
- "Creating a Schedule for One Sample" on page 5-11 describes how to begin a new schedule, starting with a single sample.
- "Scheduling Multiple Samples" on page 5-14 describes how to extend a schedule for a single sample to a batch of samples.
- "Scheduling Multiple Batches" on page 5-19 describes how to schedule multiple batches of samples.
- "Scheduling Mid-Run Sample Additions" on page 5-26 explains how to plan and test schedules that require samples to be added after a run has been started.
- "Understanding the Master Log File" on page 5-27 describes how to interpret the run time messages contained in the master log file.
- "Optimizing Step Time Estimates" on page 5-30 explains how to use data in the POLARA master log file to optimize step durations in the schedule. This topic also discusses the different types of log messages used by POLARA.

Note This chapter is only intended to provide a brief overview of how to develop and optimize schedules with a POLARA system. If you require scheduling assistance beyond the scope of this chapter, please contact Thermo LACI Customer Support for assistance. ▲

Developing a Schedule

Just as a method describes how a biological or chemical process is carried out on a single sample, a schedule defines how the process is carried out on multiple samples. By managing container movements and unit operations, the schedule makes the best possible use of available system resources.

To develop a new schedule

1. Develop a method that defines the unit operations that are used to process one sample. For details, see "Creating a Method" on page 4-5.

2. Schedule a single sample through the method. To learn how to use the POLARA scheduler, see "Using The Schedule Window" on page 5-4. To learn how to schedule a single sample, see "Creating a Schedule for One Sample" on page 5-11.

3. When the scheduler correctly schedules a single sample through the method, schedule a small batch (less than 10 samples) through the method. Adjust sample stagger values and method step parameters until the schedule is well optimized for throughput and sample uniformity. For details, see "Scheduling Multiple Samples" on page 5-14.

4. Schedule the number of samples through the method that you expect to process in a run and, if necessary, adjust schedule stagger values and method step parameters again for optimal throughput and sample processing uniformity.

5. Run the schedule on your lab system. If you need more optimization for throughput, use POLARA's **LUO Time Calculator** to adjust the average times to reflect actual system times as closely as possible. For details, see "Optimizing Step Time Estimates" on page 5-30.

6. When you have optimized the schedule for one batch, you can schedule additional batches. For details, see "Scheduling Multiple Batches" on page 5-19.

Using The Schedule Window

To define, optimize, and generate schedules, you use the **Schedule** window.

To create a new schedule

- Take one of the following steps:
  - Select New Schedule on the Workspace menu
  - Click the New Schedule button on the POLARA toolbar

To open a schedule

- Click on a schedule name in an open workspace in the left pane of the POLARA main window.

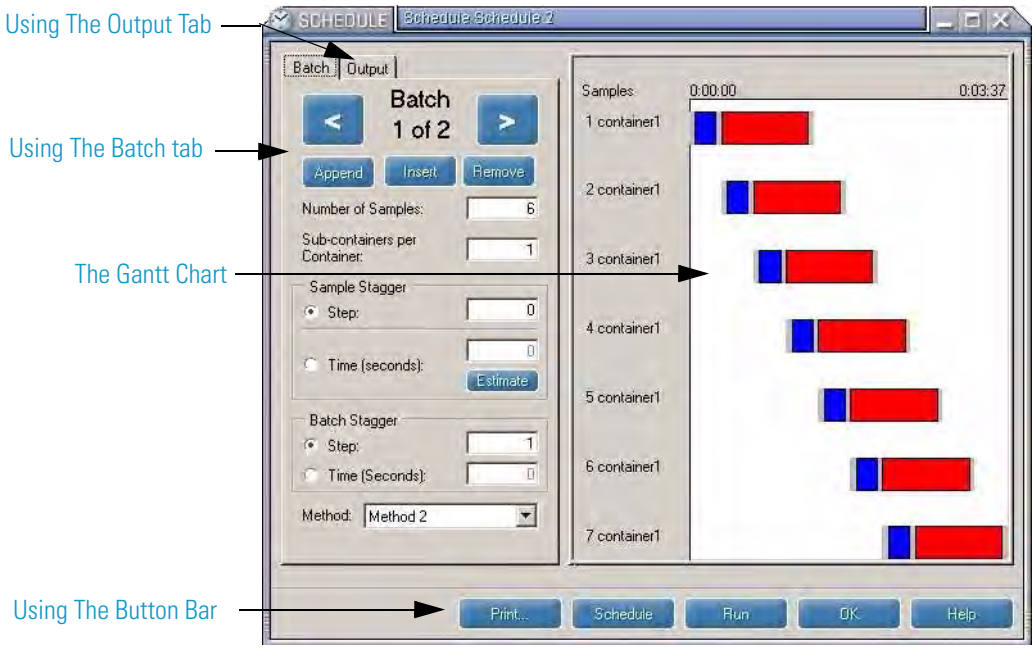


Figure 5-16. The Schedule window

The **Schedule** window has four parts:

- The **Batch** tab, where you define batches to be included in the schedule. For details, see "Using The Batch tab" on page 5-5.
- The button bar, which includes button for performing actions related to the schedule. For details, see "Using The Button Bar" on page 5-6.

- The **Output** tab, which displays information about the generated schedule. For details, see "Using The Output Tab" on page 5-6.

- The Gantt char, which displays the schedule graphically. For details, see "The Gantt Chart" on page 5-7.

Using The Batch tab

You use the **Batch** tab to set up the batches that are included in a schedule. It has the following controls and fields:

-  and  , which select the previous or next batch, respectively

- **Append**, which adds a new batch to the end of the schedule, or adds new samples to a currently running schedule

- **Insert**, which inserts a new batch before the currently displayed batch

- **Remove**, which deletes the currently displayed batch from the schedule

- **Number of Samples**, which sets the number of samples included in this batch.

- **Sub-containers per Container**, which sets the number of containers in a stacker magazine. You only set this value for methods in which multiple containers are moved together in a stacker magazine.

- The **Sample Stagger** section, which enables you to optimize sample processing uniformity by specifying a delay the scheduler must insert between samples. There are two types of sample staggers:

  - Select **Step** and enter the number of the method step that must complete before the scheduler can schedule the next sample to start.

  - Select **Time** and enter the number of seconds from the start of processing a sample that must elapse before the scheduler can schedule the next sample to start. Click the **Estimate** button to set the stagger to the minimum delay required to ensure that the instrument or mover most required to process a sample is free to process the next sample.

- The **Batch Stagger** section, which enables you to optimize batch processing uniformity by specifying a delay the scheduler must insert between batches. There are two types of batch staggers:

  - Select **Step** and enter the number of the method step in the last sample of the previous batch that must complete before the scheduler can schedule the next batch to start.

  - Select **Time** and enter the number of seconds from the start of processing the previous batch that must elapse before the scheduler can schedule the next batch to start.

- **Method**, which enables you to select the method used to process the samples in the batch.

For information on how to set up a batch and generate a schedule, see the following topics:

- "Creating a Schedule for One Sample" on page 5-11
- "Scheduling Multiple Samples" on page 5-14
- "Scheduling Multiple Batches" on page 5-19
- "Optimizing Step Time Estimates" on page 5-30

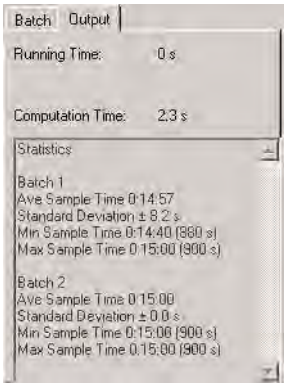
Using The Button Bar

The **Schedule** window features the following buttons at the bottom of the window:

- Print, which prints the currently displayed Gantt chart.
- Schedule, which generates a schedule for the currently-defined batch or batches.
- Run, which initiates the process of setting up and running the generated schedule on the physical lab system.
- Commit, which initiates the process of adding a newly-scheduled batch of samples to a run in progress
- OK, which saves the current schedule settings and closes the Schedule window
- Help, which displays information about the Schedule window

Using The Output Tab

The **Output** tab displays information about a generated schedule.

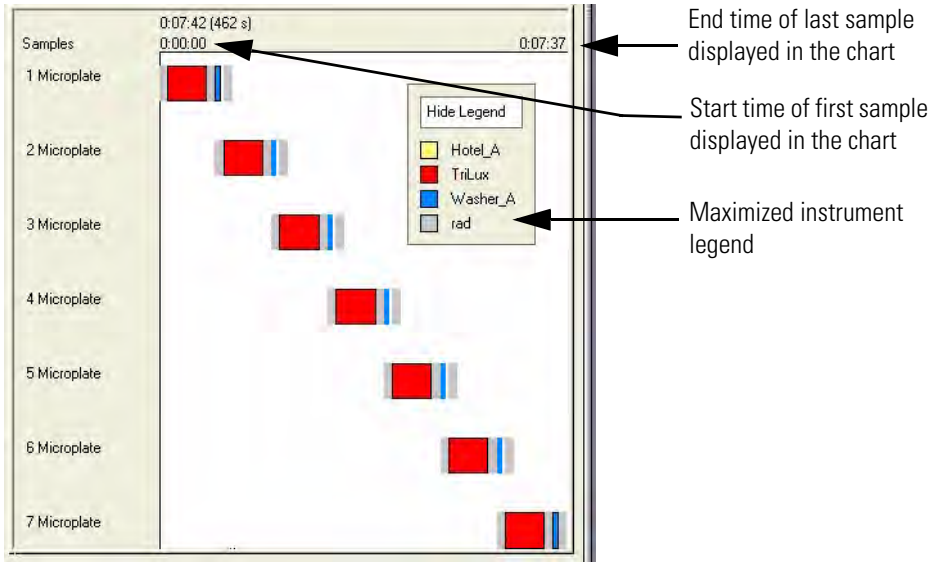


You use the **Output** information to determine whether the schedule suits the needs of the samples being run. For example, if there is too much variability in the sample times, you might wish to optimize the schedule, as described in the following topics:

- "Optimizing the Schedule for a Small Batch" on page 5-18
- "Scheduling Multiple Batches" on page 5-19.

The Gantt Chart

After you click **Schedule**, POLARA generates the schedule required to process the specified number of samples through the specified method(s), then displays the schedule graphically as a Gantt chart.



Each container in the run is represented as a horizontal bar on the chart. Each bar consists of smaller bars that represent the instrument or mover operations required to process the container, with each bar's length corresponding to the relative amount of time the operation takes to perform.

The chart legend provides a reminder of which instrument each color in the chart represents.

Table 5-1 lists the commands you can perform with the chart.

Table 5-1. Chart commands

To do this...	....take this action.
Minimize the legend	Choose **Hide Legend**.
Maximize the legend	Choose **Show Legend**.
Hide or show one instrument	Choose the instrument's color in the legend. A color with a white border indicates a hidden instrument.
Hide or show all instruments	Click the mouse cursor anywhere within the chart except the legend.

At the top of the chart, POLARA displays the start and end times for the samples shown in the chart.

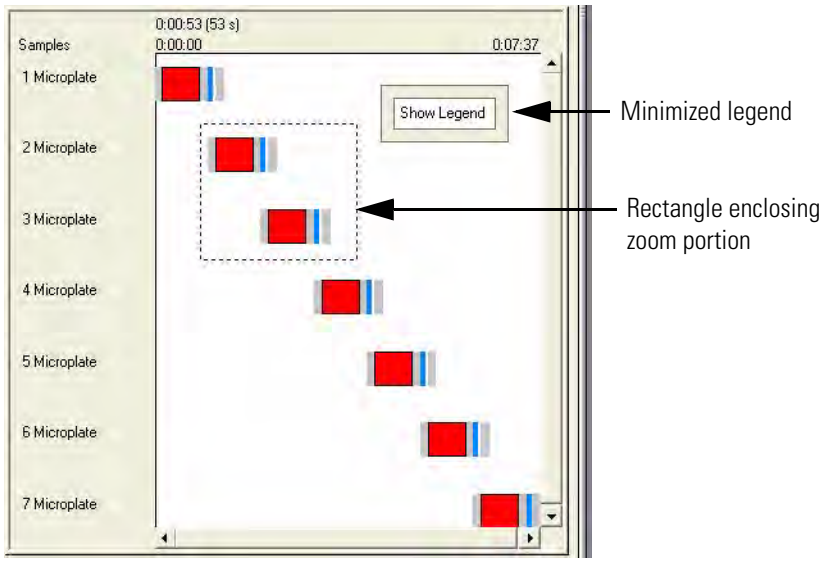
You use the Gantt chart to analyze the generated schedule. POLARA provides a number of features to make this analysis easier. For details, see the following procedures:

- "To zoom in on a portion of the chart" on page 5-8
- "To zoom out" on page 5-9
- "To view information about a container operation" on page 5-9
- "To measure the time required for a portion of the schedule" on page 5-10

To zoom in on a portion of the chart

1. Move the pointer to a corner of a rectangular area that you wish to view more closely and hold the mouse button down.

2. Drag the opposite corner of the rectangle so that the rectangle encloses the area you wish to view.



3. Release the mouse button. POLARA magnifies the selected area to fill the chart and updates the start and end times for the containers displayed. You can use the scroll bars to view other portions of the magnified chart.

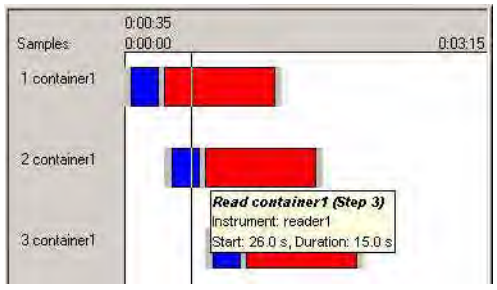
Tip Repeat steps 1 to 3 to zoom in on a smaller area to view. ▲

To zoom out

- Right-click on the Gantt chart display to zoom out. Zoom out as many times as you zoomed in to see the complete schedule.

To view information about a container operation

1. Move the mouse pointer over a container bar for several seconds. A pop-up message box displays information about the operation beneath the pointer. The information appears even when an instrument is hidden.



2. Move the mouse pointer to remove the message box.

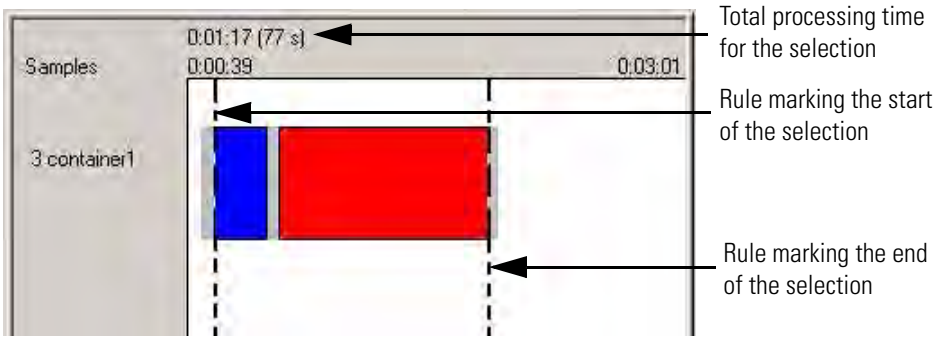
To measure the time required for a portion of the schedule

1. If necessary, zoom in on the portion of the schedule that you wish to measure, as described above.

2. Hold down the Shift key.

3. Click the mouse at the start of the portion that you wish to measure and hold down the mouse button.

4. Drag the mouse until the pointer is at the end of the portion you wish to measure. POLARA displays vertical dotted rules at the beginning and end of the portion, and shows the amount of time required to process the portion above the start time.



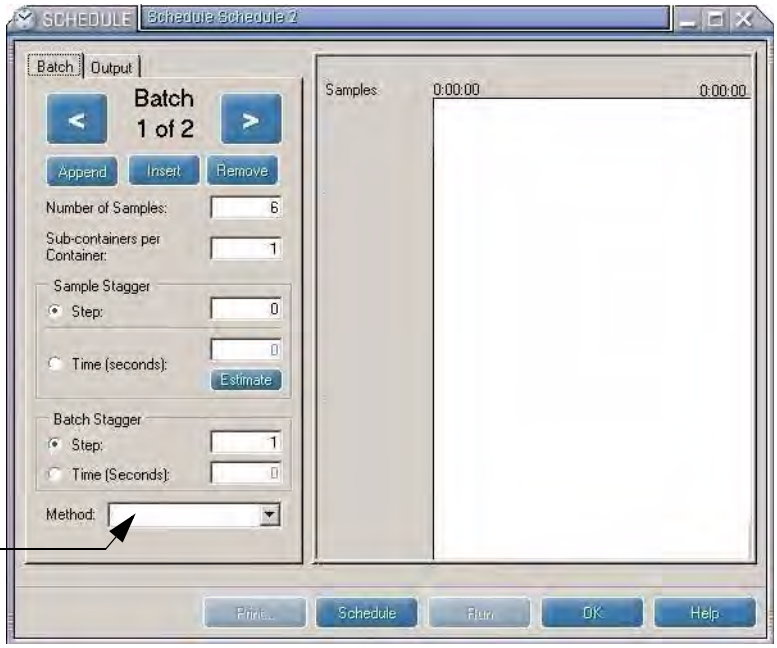
5. Release the mouse button and the Shift key.

Creating a Schedule for One Sample

To create a new schedule, you must have a workspace open.

To create a new schedule

1. In the **Workspace** menu, choose **New Schedule** and enter a name for the schedule. A blank Schedule window opens:



Click in the Method drop-down box to select a method.

2. Select a method for batch 1 using the **Method** drop-down list box.

3. Set **Number of Samples** to 1 and click **Schedule**. POLARA switches to the **Output** tab and generates a Gantt chart for the schedule in the right pane. Run statistics in the **Output** window give you an estimate of the sample times for the schedule.

   Note Sample times in the output window are partly determined by the step duration times entered in the profile. If your step duration times are not exact, the actual time taken to perform each step may be different when you perform a run. ▲

Fixing Single Sample Scheduling Failures

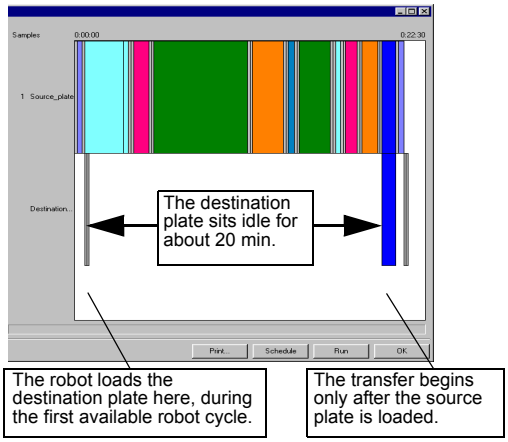
If the schedule cannot schedule a single sample through the method, the possible reasons include:

- **An unloaded container.** The method must include initial and final load operations to define where each container starts and finishes processing.
- **Missing lidder or regrip station.** The scheduler generates a "missing lidder" or "regrip required" message if your method requires delidding or a regrip station, but the interfaces are not present in the profile. Add an instance of the shadow delidder or shadow regrip station to the profile and generate the schedule again.
- **Step buffer times that cause bottlenecks.** A bottleneck can occur when the scheduler does not have an available nest into which to place a container whose buffer time has expired. If your method does not permit you to increase or eliminate buffer times, you must increase the number of available nests where the bottleneck occurs. Try adding to the method's profile more instances of the instrument where the bottleneck occurs. If you pool these instances, you may not have to change the method. For details, see Pooling in the instrument interface guide for the instrument.

Optimizing the Schedule for a Single Sample

When scheduling a single sample, follow these guidelines:

- **Look for orphaned load operations**. POLARA always tries to optimize resource use by scheduling unit operations as soon as it can. Sometimes this results in an "orphaned" load operation; a container loaded into an instrument long before it is needed there. In the example shown below, the schedule loads the destination plate into the transfer instrument as soon as the robot is available. The destination plate then sits idle for about 20 minutes, while other operations are performed on the source plate.



Tip You can view the duration of the idle time by holding the mouse pointer over the blank space between the load ▲

Orphaned load operations can be a problem if the idle container ties up an instrument, thus preventing subsequent samples from using it. Orphaned load operations can also slow down the scheduling process itself.

To correct an orphaned load operation, set a step dependency in the properties of the load operation, specifying that it not be performed until another step has been performed. In this example, you would make the destination plate's load step dependent on the step immediately preceding the point at which the destination plate is required.

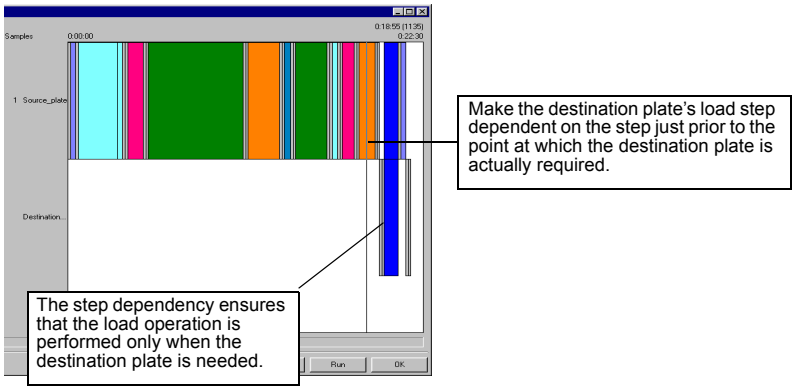


Figure 5-17. Adding step dependencies can eliminate orphaned load operations.

- **Look for large horizontal spaces between steps in the Gantt chart.** Horizontal whitespaces represent periods when a sample is inactive because it is waiting for another operation to complete or a resource to become available. You can minimize whitespace by adding load steps or step dependencies in your method.

  Eliminating gaps between steps helps to ensure sample uniformity and improves system efficiency. However, in a complex schedule with many resources, it may be difficult or impractical to eliminate all whitespace from the schedule.

  If the sample uniformity within your schedule remains acceptable, the amount of whitespace is not critical.

Scheduling Multiple Samples

Once you have determined that your method schedules well for a single sample, schedule a few samples to determine reasonable *stagger* values for the batch.

Sample staggers define when the lab system can start processing the next sample. You can use two kinds of staggers:

- A *step* stagger, which tells the scheduler which step in the processing of the current sample must execute before processing of the next sample can start

  Note Step staggers are *minimum* values; the scheduler can start processing of the next sample on any step beyond the specified step stagger. ▲

- A *time* stagger, which tells the scheduler how much time must elapse before processing of the next sample can start

To schedule multiple samples

1. Open the schedule window.
2. On the Batch tab, set **Number of Samples** to a small number (e.g. 5 or 10 samples).
3. Select a **Time** or **Step** stagger and enter an appropriate value. It is generally a good idea to start with step staggers.

   Tip If you are using time staggers, see "Determining a Minimum Time Stagger" on page 5-17. ▲
4. Click **Schedule**. POLARA generates a Gantt chart for the schedule in the right pane.

Fixing Batch Scheduling Failures

If the schedule cannot schedule multiple samples after successfully scheduling one sample, the possible reasons include:

- **Not enough nests to hold the containers.** Make sure that the batch size is realistic and that the storage devices used in the method can accommodate the total number of containers.

Check instrument instance properties in the method's **profile** to make sure that they allow containers in all required nests.

If your lab system has more than one storage instrument of the same type, you can pool the storage instruments in POLARA, enabling your methods to treat them as a single storage unit with nests totalling the sum of the two. For details, see Pooling in the instrument interface user guide for your storage instrument.

- **Step buffer times that create bottlenecks.** A bottleneck can occur when the scheduler does not have an available nest into which to place a container whose buffer time has expired. If your method does not permit you to increase or eliminate buffer times, try increasing staggers or adding more instances of the instrument where the bottleneck occurs.
- **Staggers that are too short.** Increase the staggers and schedule again.
- Too many reserved nests. Either remove all unnecessary nest reservations (such as those on the nests of shakers) or increase the total number of available nests by adding more instances of instruments.

Choosing Appropriate Staggers

This example uses a cell-based assay, whose method is specified as follows:

Method Steps:

Step	Instrument	Description	Reserves and Load step dependencies	Unitop	Paramet
1	Carousel	Load Tips	Tips(No Lid)	Load	0,Tips,0
2	Incubator	Load AssayPlate	AssayPlate(Lid)	Load	0,AssayF
3	Washer	Run Wash Assay Plate using AssayPlate	AssayPlate(No Lid)	RunP...	4500, 0,
4	Multidrop	Run Dispense media and dye using AssayPlate	AssayPlate(No Lid)	RunP...	2500, 0,
5	Incubator	Incubate AssayPlate for 1h	AssayPlate(Lid)	Incub...	360000,
6	Pipettor	Run Transfer compounds using AssayPlate	AssayPlate(No Lid)	RunP...	6000, 0,
7	Incubator	Incubate AssayPlate for 1h	AssayPlate(Lid)	Incub...	360000,
8	Imager	Run Dispense to and read assay plate using AssayPlate and Tips	AssayPlate(No Lid); Tips(No Lid)	RunP...	18000,
9	Waste	Load Tips		Load	0,Tips,0,
10	Incubator	Load AssayPlate	AssayPlate(Lid)	Load	0,AssayF

Figure 5-18. The method for a simple cell-based assay

The schedule for a single sample looks perfect:

However, when additional samples are scheduled with a step stagger of 1, there are increasing idle times at key points, resulting in large deviations between sample times. The idle times result from contention for robot resources and the Imager instrument:

Figure 5-19. Sample uniformity suffers from contention for resources

You could achieve consistent sample times by specifying a step stagger of 10, meaning that a new sample wouldn't start processing until the previous one was complete. However, then samples would not be processed in parallel, leaving resources under-utilized, and the processing time for the batch would be excessively long.

By implementing an appropriate stagger, you can achieve both a high level of parallel processing and consistent sample times. In some cases, you will be able to use a step stagger. In others, such as this example, you will need to use a time stagger:

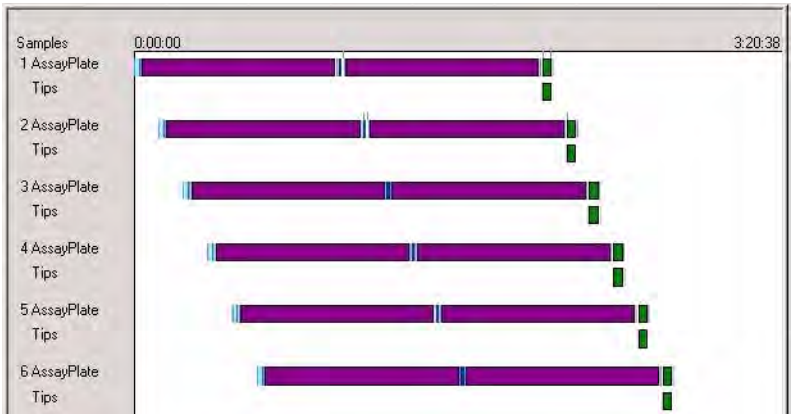


Figure 5-20. A sample stagger time resolves the resource conflicts

The goal is to achieve the greatest level of uniformity between samples, while using system resources most efficiently, thus ensuring that samples are processed consistently and total batch time is as short as possible.

Note In this case, the required sample stagger time is 437s (257s for robot time plus 180s for Imager processing time). ▲

Determining a Minimum Time Stagger

Once you have created a schedule for one sample, you can determine the minimum time stagger required between each sample in a batch.

To calculate minimum stagger time

1. Enter the number of samples in the batch.

2. In the Sample Stagger section of the **Batch** tab, click the **Time** radio button.

3. Enter the time it takes to process one sample minus thirty percent in the time stagger value field, or click **Estimate**. (The value generated by clicking **Estimate** is a minimum: to improve sample processing uniformity, you might need to increase the value generated by clicking **Estimate**, but decreasing the value will not improve uniformity.).

4. Click **Schedule** to generate a schedule for the required number of samples.

5. If the time stagger value does not yield a schedule of acceptable sample processing uniformity, increase the value and re-generate the schedule.

6. If the time stagger value yields a schedule of acceptable sample processing uniformity, but unacceptable throughput, decrease the value and re-generate the schedule.

If this trial-and-error approach does not yield an acceptable schedule, consider adding more instruments to the system.

Optimizing the Schedule for a Small Batch

When determining stagger values, start with a small batch. Generate a schedule and review each sample processing step in the Gantt chart carefully. Here are some of scheduling issues you may encounter, with some suggestions on how to resolve them:

- **Schedule does not reach a steady state.** As you increase batch size, a repeating pattern of step colors in each sample should form in the Gantt chart. If the order and length of the steps changes randomly from sample to sample, you have not reached a steady state. Increase the batch size and adjust sample staggers or method parameters until the pattern stabilizes.

- **Gantt chart shows large variations in sample times.** If your schedule works well for small batch sizes but exhibits large time variations in sample processing times when you increase the batch size, your schedule has not reached a steady state.

  Slightly increasing the incubation time for each sample, introducing step dependencies, or using multiple batches may also help to normalize sample processing times.

- **Samples are not processed in parallel.** If there is no vertical overlap of samples in the Gantt chart, the system is waiting until each sample is processed before starting the next one. This is a poor use of system resources. You can generally improve efficiency by decreasing your stagger size so that the samples overlap.

Scheduling Multiple Batches

You schedule multiple batches for reasons that include the following:

- You want to use a different methods in a single run. POLARA's scheduler enables you to overlap processing of two different methods. Using a timed batch stagger, you can start the second batch while the first batch is still processing.
- Your method encounters scheduling problems that cannot be resolved using sample staggers (because of resource limitations, for example).

Note You can also add a batch to a running schedule. For details, see "Scheduling Mid-Run Sample Additions" on page 5-26. ▲

To add a batch to a schedule

1. In the schedule window, click **Append** or **Insert** to create a new batch. The schedule window displays the pane for the new batch.

   Note Use the  and  buttons in the **Schedule** window to scroll forward and back through the batches in a schedule. ▲

2. In the batch window, click in the **Method** drop-down box and select the method to use. You do not have to use the same method for all batches, but each method must use the same profile.

3. Enter a number of samples for the batch and set appropriate sample staggers to be used within this batch.

4. Enter a value for the batch stagger. You can use a time stagger or a step stagger:

   - **Batch time staggers are calculated from the beginning of the run** (except when adding batches to a run in progress. See "Scheduling Mid-Run Sample Additions" on page 5-26). This can be useful if you want to overlap batch scheduling. By setting a time stagger that is shorter than the time required for your first batch, you ensure that batches 1 and 2 are performed in parallel. However, as with sample time staggers, you must take care to ensure that your unit operations are well timed or the schedule will accumulate slip time over a run.

- **Batch step staggers are calculated relative to the beginning of the last sample of the previous batch**. So if you have 15 steps in the method used for Batch 1 and you want POLARA to wait until Batch 1 has completed before beginning Batch 2, the batch step stagger for Batch 2 should be 15.

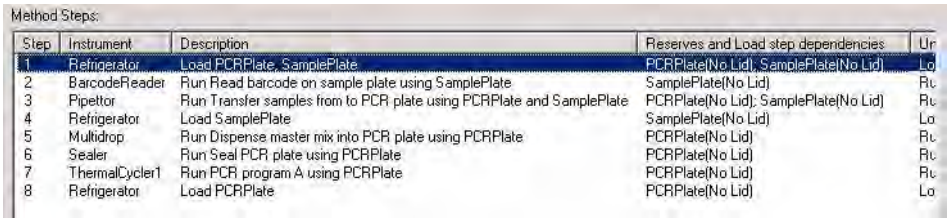
5. To view the schedule for the complete run, click **Schedule**.

See the following topics for practical examples of how to use multiple batches to resolve scheduling problems:

- "Resolving a Resource Conflict with Multiple Batches" on page 5-20
- "Resolving a Resource Conflict with Batch Staggers" on page 5-23

Resolving a Resource Conflict with Multiple Batches

This example uses a polymerase chain reaction (PCR) method in a workspace with six thermal cyclers.


Method Steps:

Step	Instrument	Description	Reserves and Load step dependencies	Un
1	Refrigerator	Load PCRPlate, SamplePlate	PCRPlate(No Lid), SamplePlate(No Lid)	Lo
2	BarcodeReader	Run Read barcode on sample plate using SamplePlate	SamplePlate(No Lid)	Ru
3	Pipettor	Run Transfer samples from to PCR plate using PCRPlate and SamplePlate	PCRPlate(No Lid); SamplePlate(No Lid)	Ru
4	Refrigerator	Load SamplePlate	SamplePlate(No Lid)	Lo
5	Multidrop	Run Dispense master mix into PCR plate using PCRPlate	PCRPlate(No Lid)	Ru
6	Sealer	Run Seal PCR plate using PCRPlate	PCRPlate(No Lid)	Ru
7	ThermalCycler1	Run PCR program A using PCRPlate	PCRPlate(No Lid)	Ru
8	Refrigerator	Load PCRPlate	PCRPlate(No Lid)	Lo

Figure 5-21. An example PCR method

A schedule created for a run of only six samples results in very little deviation in processing times between samples because there is always a thermal cycler available to process a sample.

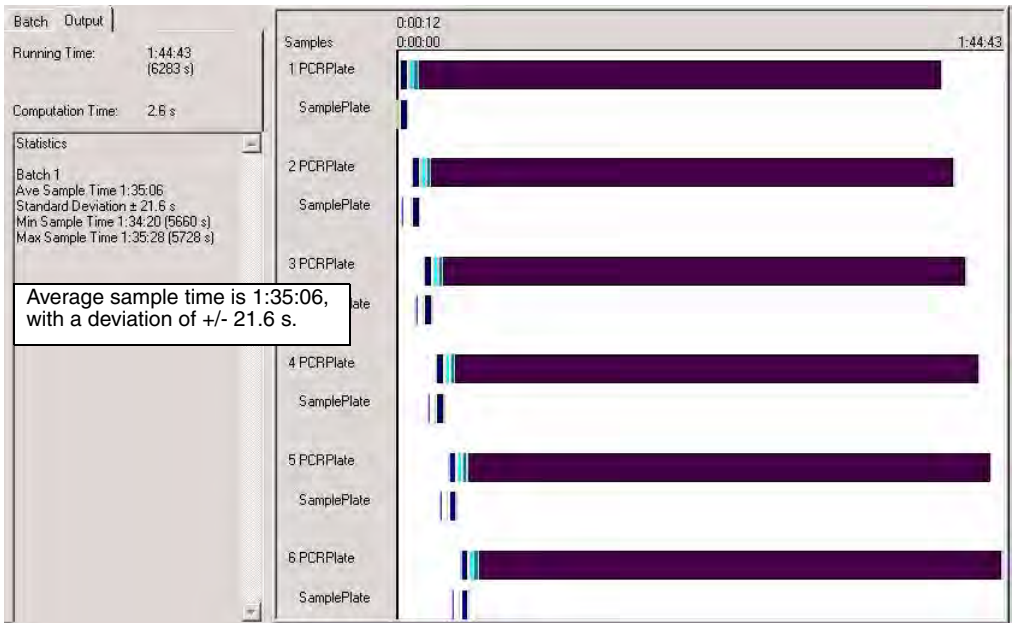


Figure 5-22. Six samples can be processed through the PCR method efficiently and consistently when there is one thermal cycler available for each sample.

However, a schedule created for *seven* samples results in a lengthy idle time before the seventh sample can be loaded into a thermal cycler; it has to wait until the first sample is finished. The seventh sample takes twice as long as the first six and spends over an hour sitting in the sealer:

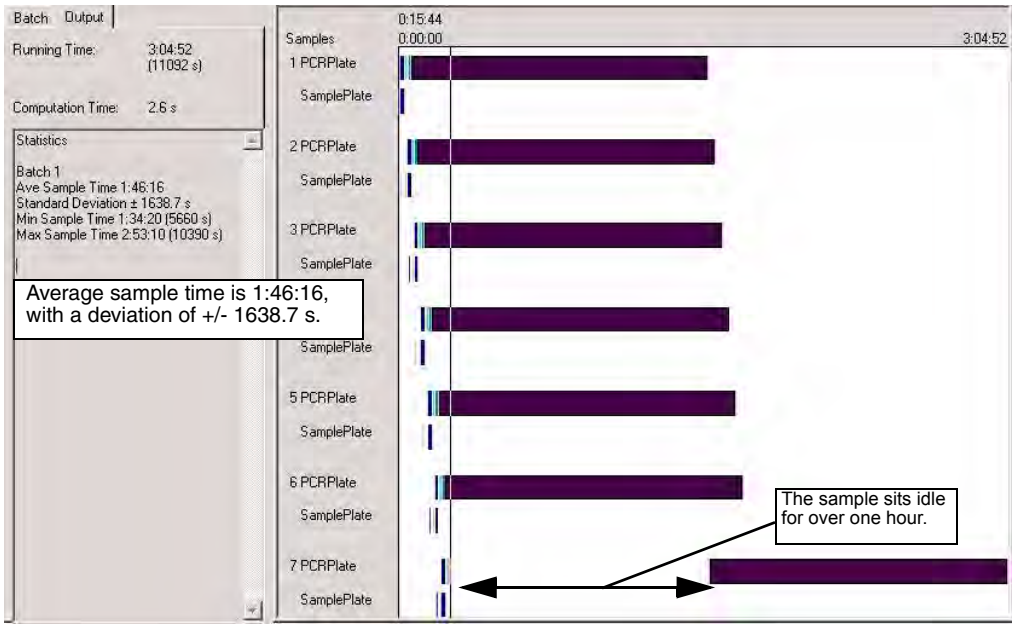


Figure 5-23. The seventh sample causes a serious scheduling problem

Trying to overcome this problem with time or step staggers may result in consistent sample times, but you would lose the very efficient processing within each group of six samples, because *every* sample would be staggered. This would result in an excessively long batch time.

The more efficient solution is to set up the run as a series of batches with six samples each. (This makes sense, since the process is limited by the six thermal cyclers.) You can set the Batch Stagger Time to slightly longer than the maximum cycle time recorded for the first six samples (about 5780 or 5800 s) to ensure that the first sample of one batch is finished before the next batch starts. Such a configuration results in consistent processing for each sample:

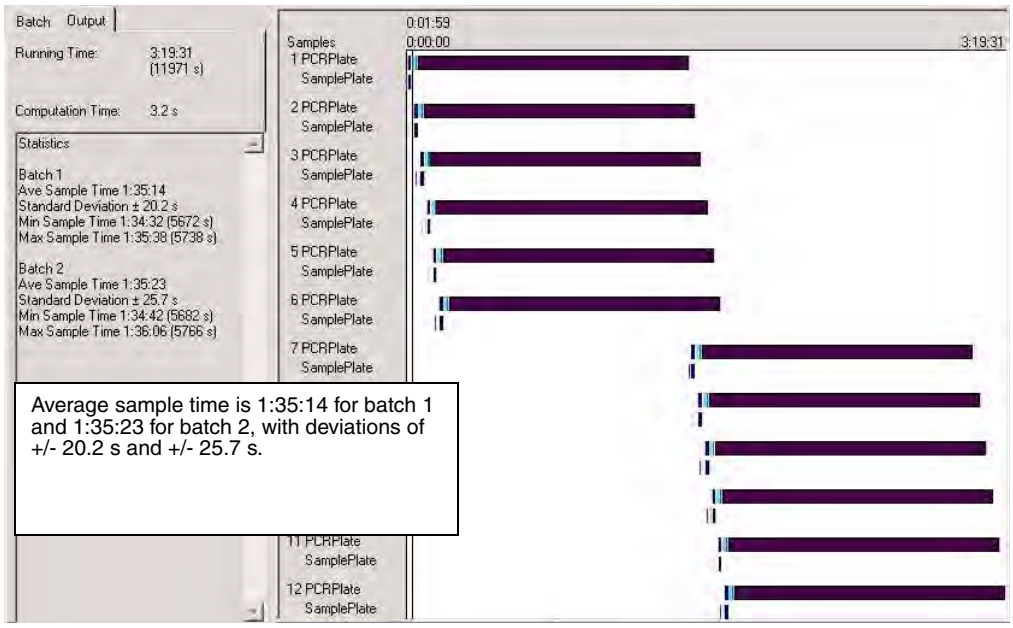


Figure 5-24. Multiple batches improve sample processing uniformity and make more efficient use of resources.

Resolving a Resource Conflict with Batch Staggers

Here is an example of a schedule where the method makes heavy use of the robot (grey) at the beginning and end of processing each sample:

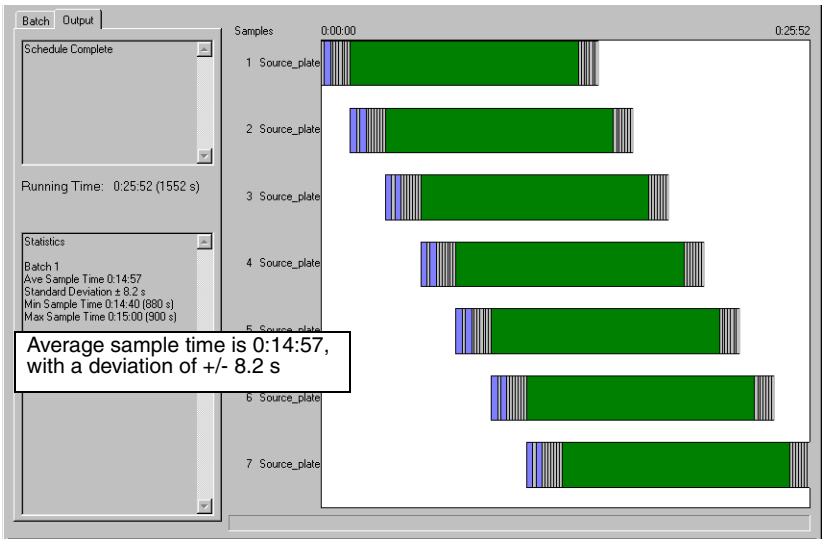


Figure 5-25. This schedules uses a method that provides enough time in the incubation step for the robot to load seven samples into the incubator before it must start removing them.

The long incubation step (green) in this method gives the robot time to load seven samples into the incubator before it must start removing samples that have finished incubating.

However, when we schedule more than seven samples, the robot operations begin to overlap: it has to load the additional samples as well as remove the initial samples. The robot becomes a bottleneck. White spaces start to appear in the Gantt chart before and after the incubation step, since the robot can't possibly move fast enough to move a sample when it is ready to be moved. Sample processing times become increasingly non-uniform, with the deviation from average sample processing time increasing from 8.2 seconds for seven samples to 51.0 seconds for 15 samples:

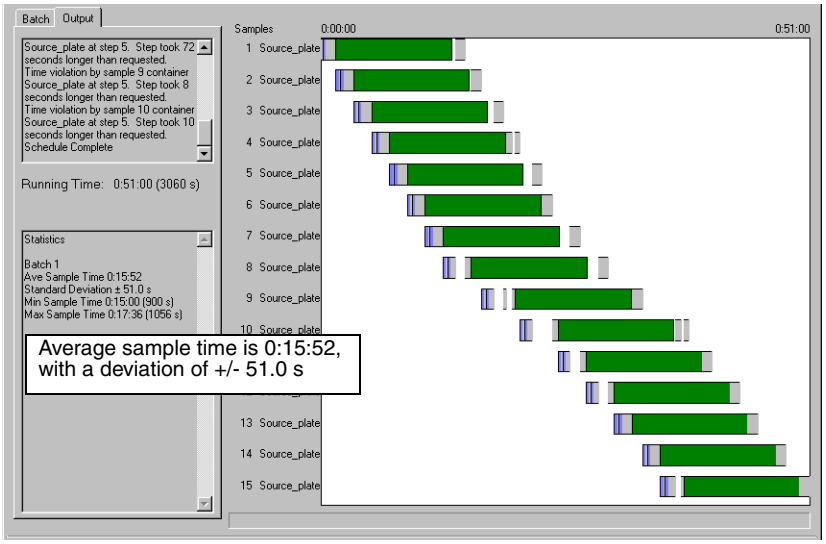


Figure 5-26. With a larger number of samples, the robot has to both load samples into and remove samples from the incubator, causing sample processing times to become increasingly non-uniform.

A possible solution is to increase the incubation time so that the robot can finish loading all samples into the incubator before it must start unloading samples that have completed incubation (assuming the incubator has the capacity to store all the samples). But lengthening the incubation period might well compromise the validity of the method.

To improve sample processing uniformity without lengthening incubation times, you can use a batch stagger.

Using Batch Time Staggers to Improve Uniformity

In this example, you can eliminate the robot as a bottleneck by separating samples into batches of seven and setting batch time staggers large enough to prevent the system from starting to process a new batch until the previous

batch is complete. In our example, this results in a longer run time for the total number of samples (52 minutes for 14 samples, instead of 51 minutes for 15 samples), but restores sample processing times to their original uniformity.

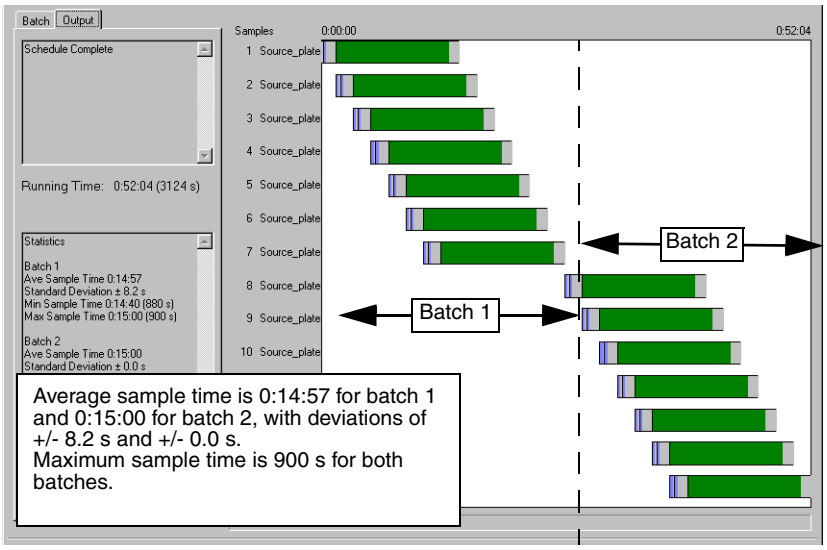


Figure 5-27. Breaking the schedule into batches and setting a batch stagger to separate batch processing restores sample processing uniformity.

Scheduling Mid-Run Sample Additions

POLARA version 2.1 and later enables you to add new samples to a run in progress, provided that the nests required to hold the samples' containers are available.

Note In order for the POLARA scheduler to know what nests are available for the added samples, you must reserve instrument nests to which containers return during processing. For details about reserving nests, see "About Step Properties" on page 4-3. ▲

To add samples to a run, you schedule them as a new batch in the schedule that controls the run. POLARA then replaces the unexecuted part of the run's schedule with the updated schedule.

Note When adding samples to a schedule for a run in progress, the batch stagger times you specify start from the point in the run where POLARA replaces the previous schedule with the updated one. ▲

Because samples added to a run are treated as a new batch, they can be processed with the same method as other samples in the run, or with a different method.

Before providing system operators with schedules they can use to add samples to a run, you must test the schedules to ensure they support sample addition.

To develop schedules that support mid-run sample additions

1. Using the schedule for the initial run, generate a schedule for the additions, as described in "Adding Samples to a Run" on page 6-22. Find the optimal settings for the added batch, including the following:

   - the number of samples to add
   - the Sample Stagger for the added samples
   - the Batch Stagger for the added batch

2. When you have generated an optimal schedule, run it, adding the samples as planned, to ensure that the settings work appropriately.

3. Document the appropriate settings and provide them to the operators who will be running the schedule and adding the samples.

Understanding the Master Log File

The master log file contains general system messages. Some common message types are described below

- INFO messages provide status information about operations performed by the system. Common types of INFO messages are described in more detail in "Common Types of INFO Messages" on page 5-28.

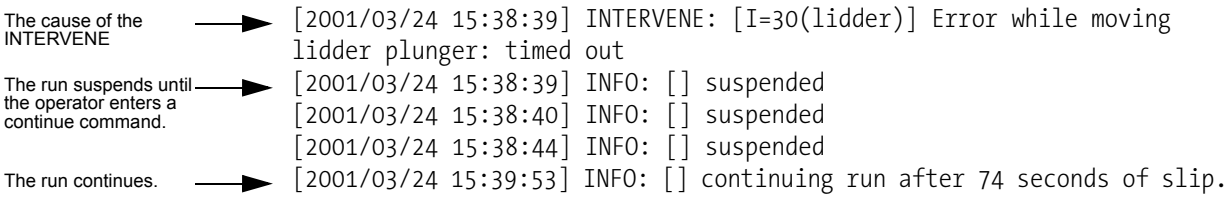
- WARNING messages are displayed when the system receives an unexpected result or cannot immediately complete a task.

  Note Warning messages do not necessarily indicate a problem. For example, if the system is waiting for a wash program to complete, it will query the washer at regular intervals to determine whether the wash has completed. While the wash is still in progress the query times out, producing a warning message. ▲

  Note The system generally retries an operation after a WARNING. After repeated failures, however, a WARNING condition may progress to INTERVENE. ▲

- INTERVENE messages indicate that POLARA has encountered a problem it cannot automatically resolve.

  During an INTERVENE condition, POLARA requests input from the operator (for example, the operator may be asked to remove a blocked plate). The run remains suspended until the operator provides the necessary input to indicate that the problem has been cleared.



Note Following an INTERVENE, any accumulated slip time is added to the schedule. This ensures that operations are still performed at the correct time. ▲

- An ERROR message indicates a minor problem that requires intervention. If the system is operating in attended mode, POLARA treats an ERROR condition as an INTERVENE: it displays a message and suspends the system until the operator resolves the problem. If the system is in unattended mode, POLARA treats the ERROR condition as a WARNING.

  For example, if a required microplate is not present during the initial Get operation for an incubator nest, POLARA issues an ERROR message and continues in one of the following ways:

  - In attended mode, POLARA suspends the system and asks the operator to load a microplate into the expected location.

  - In unattended mode, POLARA ignores the microplate and moves on to perform the next operation in the method.

- FATAL messages indicate that a run was terminated unexpectedly, generally due to an Abort command issued by the operator.

  Tip A FATAL message at the beginning of a run may indicate that an instrument failed to initialize properly. If you are encountering FATAL messages at the beginning of a run, check the settings in the instrument's configuration file, and make sure the configuration file for each instrument is saved in the correct location.

An Abort command causes these FATAL messages in the log. →
This INFO is also caused by the Abort. →

```
[2001/03/28 12:57:34] FATAL: [I=0(rad)] User aborted the run!
[2001/03/28 12:57:34] FATAL: [] Error during initialization -- run
aborted
[2001/03/28 12:57:34] INFO: [] could not suspend!
```

▲

Common Types of INFO Messages

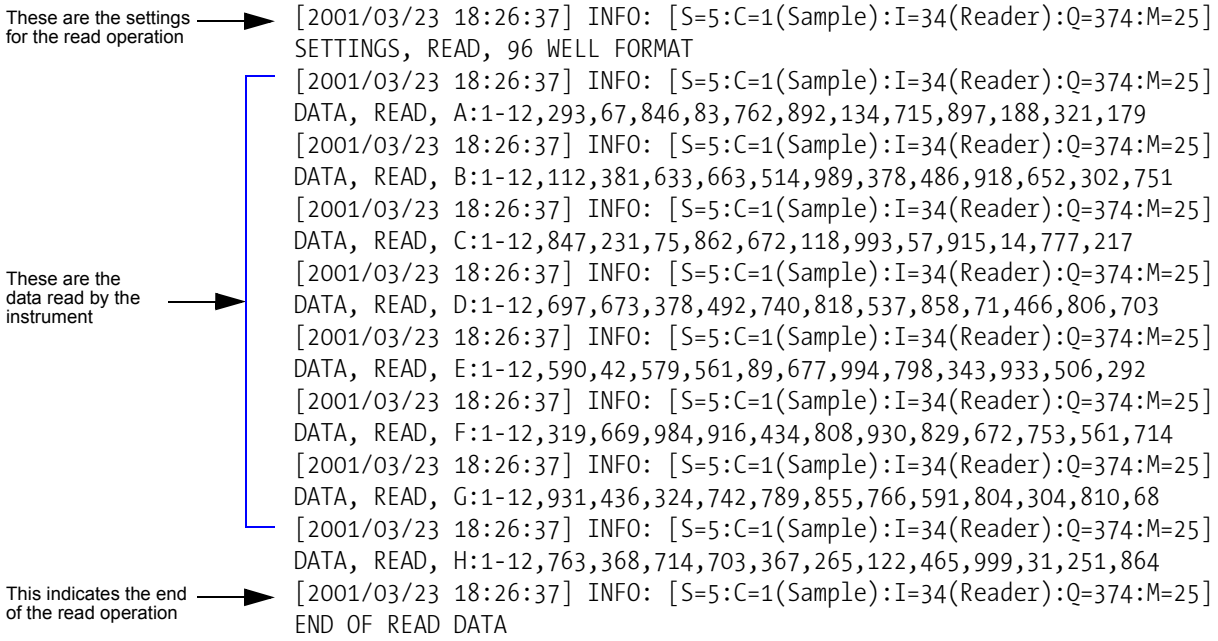
INFO messages can be useful for tracking a particular step or action, or for providing statistics about your run. Common types of INFO messages include:

- General informational messages from instrument servers. For example:

```
[2001/04/03 09:58:21] INFO: [I=33(Washer1)] Bio-Tek ELx405: Operation
completed succesfully.
[2001/04/03 10:00:46] INFO: [] UPS Online Mode
[2001/04/03 10:00:58] INFO: [I=29(sbcr)] Bar Code for S:2 C:1 was
"BC0125"
```

Tip For more detailed information on instrument functions, refer to the instrument log files located in the CROSnt /log directory. ▲

- SETTINGS, DATA, and END OF READ DATA messages, which contain the data obtained by a reader. For example:

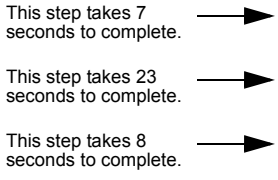



```
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
SETTINGS, READ, 96 WELL FORMAT
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, A:1-12,293,67,846,83,762,892,134,715,897,188,321,179
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, B:1-12,112,381,633,663,514,989,378,486,918,652,302,751
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, C:1-12,847,231,75,862,672,118,993,57,915,14,777,217
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, D:1-12,697,673,378,492,740,818,537,858,71,466,806,703
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, E:1-12,590,42,579,561,89,677,994,798,343,933,506,292
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, F:1-12,319,669,984,916,434,808,930,829,672,753,561,714
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, G:1-12,931,436,324,742,789,855,766,591,804,304,810,68
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
DATA, READ, H:1-12,763,368,714,703,367,265,122,465,999,31,251,864
[2001/03/23 18:26:37] INFO: [S=5:C=1(Sample):I=34(Reader):Q=374:M=25]
END OF READ DATA
```

- STARTING messages, which indicate the beginning of a step. For example:

```
[2001/04/03 10:00:30] INFO:
[S=1:C=1(AssayPlate):I=36(Carousel):Q=12:M=6] STARTING Put
[2001/04/03 10:00:39] INFO:
[S=1:C=1(AssayPlate):I=36(Carousel):Q=13:M=6] STARTING Incubate
[2001/04/03 10:00:39] INFO:
[S=2:C=1(AssayPlate):I=36(Carousel):Q=14:M=3] STARTING Get
```

- DONE messages, which are issued when a step completes. The start and end times included in the DONE message can be used to optimize step durations in your schedule.




```
[2001/03/23 16:56:02 - 2001/03/23 16:56:09] INFO:
[S=1:C=1(Sample):I=36(shelf):Q=1:M=3] DONE Get
[2001/03/23 16:56:10 - 2001/03/23 16:56:33] INFO:
[S=1:C=1(Sample):I=30(lidder):Q=2:M=3] DONE Delid
[2001/03/23 16:56:33 - 2001/03/23 16:56:41] INFO:
[S=1:C=1(Sample):I=35(Transfer):Q=3:M=3] DONE Put
```

Optimizing Step Time Estimates

When optimizing and planning schedules, it is important that the step durations in instrument instance properties match the actual times taken by the system when performing the step.

If your method *underestimates* step times, the schedule will encounter slip time errors as the system struggles to perform the run in the requested time. This can have a severe effect on sample uniformity, as later samples will take longer and longer to complete.

Tip You can check for slip time by comparing the actual run time with the estimated run time. If you have underestimated step times, the actual time for the run will be longer than the estimated time. ▲

If your method *overestimates* step times, the system will sit idle and wait for the estimated time to elapse before continuing with the next step in the method. This can prevent you from achieving optimal system throughput.

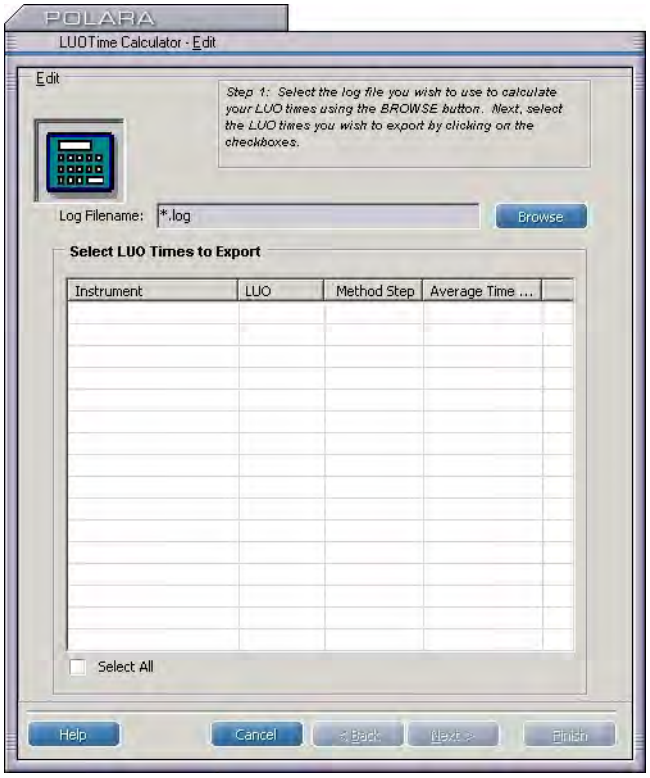
During a run, all system data and run times are appended to a master log file, which contains exact timing data from DONE operations.

Note To view the name of the log file, click  in the run monitor window while setting up your run. By default the log file is stored in the CROSnt /data directory as lastrun.log. If no name is specified, the master log file is stored in the CROSnt /log directory as master.log. ▲

To adjust the step time estimates in a profile to more accurate values, you use the **LUO Time Calculator**.

To use the LUO Time Calculator

1. In the POLARA **Tools** menu, choose **LUO Time Calculator**.



2. Follow the instructions at the top of each page.

Where To Go From Here

Once you have created a schedule, you can perform runs with it. For details, see Chapter 6, Setting up and Performing a Run.

Chapter 6 Setting up and Performing a Run

This chapter describes how to perform the steps required to setup, start, and monitor a run. It also describes how to recover a run, and finish up after a completed run. It covers the following topics:

- "Setting up the Run" on page 6-3, which describes how to prepare the lab system for a run, and initialize the instruments required
- "Using The Monitor Window" on page 6-5, which describes the Monitor window and explains the function of the toolbar buttons, status indicator lamps, tabs, and status bar
- "Starting a Run" on page 6-12, which describes how to start the run
- "Setting the AR Start Position" on page 6-13, which describes how to set up the safe start position for the robot
- "Monitoring a Run" on page 6-15, which provides an overview of monitoring a run
- "Suspending a Run" on page 6-18, which describes how to suspend, or pause, a run and how to continue the run
- "Inserting Pause Points into a Run" on page 6-19, which describes how to program pauses into a run to allow you to add reagents or set up new samples at a specific time.
- "Adjusting Containers or Reagents During a Run" on page 6-21, which describes how to safely suspend a run and work with containers or reagents that need attention
- "Adding Samples to a Run" on page 6-22, which describes how to add new samples to a run that is already in progress
- "Halting a Run" on page 6-24, which describes how to halt, or abort, a run and what happens when a run is halted

- "Recovering an Aborted Run" on page 6-25, which describes how to recover a run that was interrupted either because the operator halted it, or because of a system failure (such as a power outage)
- "Finishing Up After a Run" on page 6-32, which describes what happens when a run completes and explains how to finish up
- "Using the Remote Monitor" on page 6-33, which describes how to use the Remote Monitor to monitor a run from a remote location

Setting up the Run

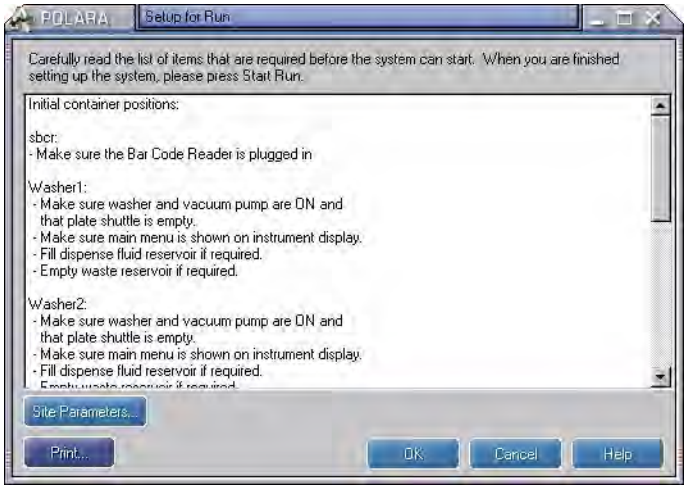
After opening and configuring a schedule, you are ready to set up the run.

To set up a run

1. In the **Schedule** window, click **Run**.

   Note On POLARA DM systems, POLARA launches the DM Startup Wizard, which prepares the DM movers for a run. If the DM movers require operator intervention, such as turning on master motor power, the DM Startup Wizard prompts you to take the appropriate action. ▲

   POLARA displays the **Setup for Run** window:



2. Carefully read and perform each step displayed in the **Setup for Run** window.

   Tip Click **Print** to print a checklist of steps. ▲

3. Click **Site Parameters**. Perform any steps displayed.

4. Click **Start Run**. The system initializes each instrument. During initialization the **Monitor** window appears and the following occurs (see "Using The Monitor Window" on page 6-5 for details):

   - The **Initializing Status Indicator** turns blue.
   - The CROSnt window opens briefly, showing the initialization process, then minimizes.

- The results of the initialization are listed in the **Summary** tab of the **Monitor** window.

5. Check the **Summary** tab of the **Monitor** Window, as described in "Summary Tab" on page 6-7. You should see a list of instruments that have initialized successfully.

See "Starting a Run" on page 6-12 for information on starting the run from the Monitor window.

Using The Monitor Window

The **Monitor** window enables you to control a run, monitor the progress of samples through the system, check the status of instruments, and observe warning and error messages. The **Monitor** window has a toolbar, status indicator lamps, tabs, and a status bar.

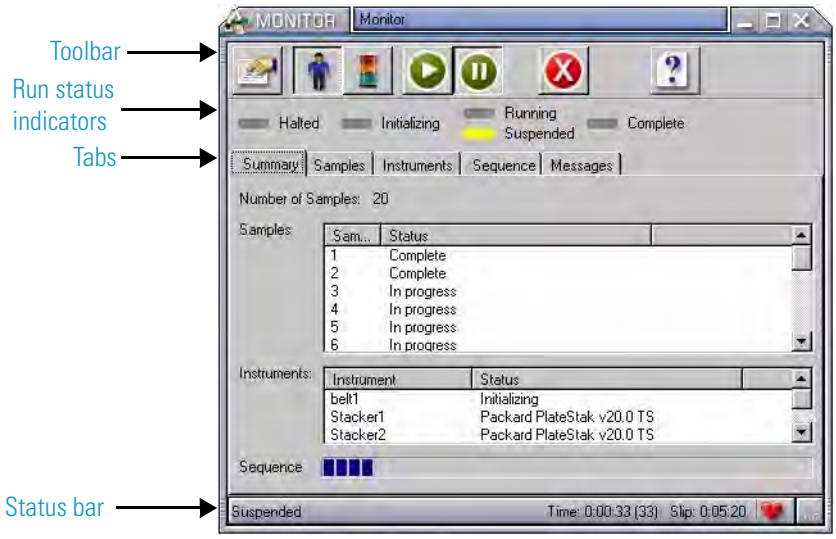


Figure 6-28. Monitor window key elements

Using the Monitor Toolbar

The toolbar buttons allow you to perform the following tasks:

Parameters: Opens the parameters dialog box, which allows you to view the location and name of the data log file, and the name of the sequence file.

WARNINGDo not change the following values without consulting your POLARA administrator. ▲

- data_dir: The location of the POLARA data directory
- log name: The name of the log file that POLARA creates in the data directory for each run
- sequence_name: The name of the sequence file that controls the run

Attended Mode: Set the mode of operation as one of the following:

- Attended: Use this mode if an operator is overseeing the run. When POLARA detects an error, it alerts operators who must clear the error before continuing the run.
- Unattended: Use this mode when an operator is not overseeing the run. POLARA will attempt to handle error conditions. If unable to do so in a safe manner, it suspends the system.

  You can change the mode of operation any time during the run.

Reset Beacon: Resets the beacon and the status indicator lamps.

Start/Continue: Starts a run, and continue a run that has been Suspended.

Suspend: Suspends a run. The current unit operation will be completed before the robot stops moving.

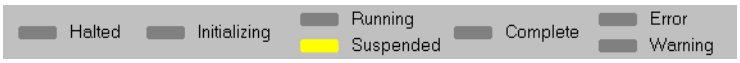
Halt: Ends the run. A dialog box appears that prompts you to confirm that the run should be halted. When confirmed, the run ends.

WARNING The Suspend and Halt buttons do not stop robot motion immediately! In an emergency, press an E-Stop button. ▲

Help: Displays online help for the current Monitor tab.

Understanding the Status Indicators

The status indicators display the state of the system:



Only one of the following status indicators will be lit at any time:

- **Halted**: Turns red when a run is halted.
- **Initializing**: Turns blue when the system is initializing.
- **Running**: Turns green when a run is underway.
- **Suspended**: Turns yellow when a run is paused.
- **Complete**: Turns blue when a run is complete.

The following status indicators may be lit individually or in combination:

- **Error**: Turns red when an error occurs.
- **Warning**: Turns yellow when a warning occurs.

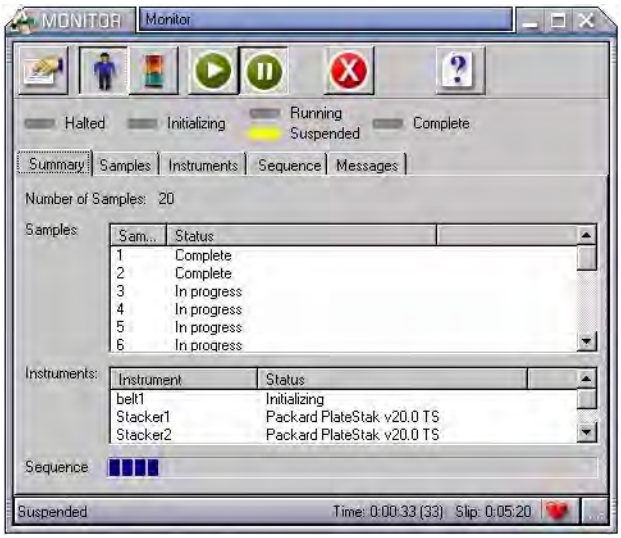
Using the Monitor Tabs

Each tab displays selected information that helps you to monitor the progress of a run.

Summary Tab

The **Summary** tab displays the status of each sample and the status of each instrument in the schedule. Use this tab to check whether each instrument in the system has initialized properly. While an instrument is initializing, its status reads “Initializing”. If initialization is successful, the status changes. If initialization is unsuccessful, the status remains “Initializing”. In addition, the warning status indicator flashes yellow, and a message box appears that

allows you to abort or retry initialization. The example below shows seven instruments that have successfully initialized.

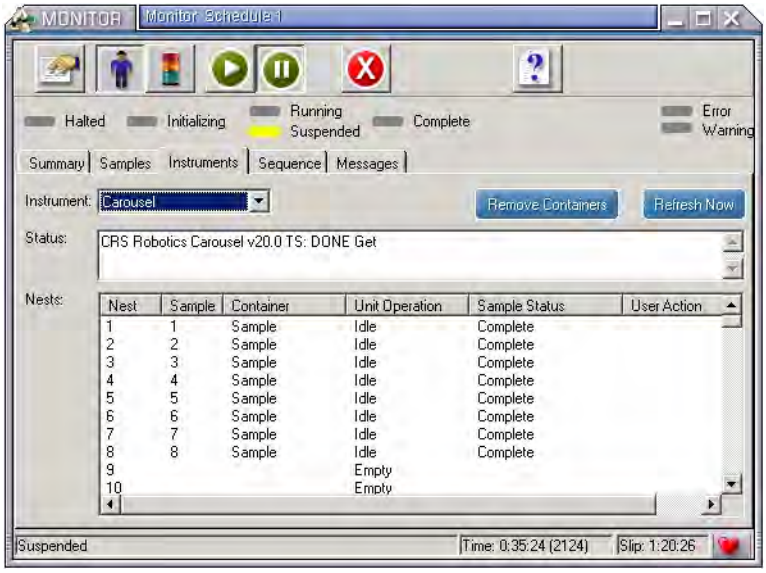


Samples Tab

The **Samples** tab displays the status of a specific sample. Use this tab to view the containers and method steps used by this sample.

Instruments Tab

The **Instruments** tab displays the status of a specific instrument. Use this tab to check whether the instrument is operational and to view details about each nest in the instrument.



Sequence Tab

The **Sequence** tab displays the progress of the unit operations in the schedule. Use this tab to monitor whether a unit operation is underway, or complete. Unit operations that are underway are indicated by a green arrow, while completed operations are indicated by a red checkmark.

The **Sequence** tab displays only *one* segment of the currently running schedule. To select the segment displayed, use the segment display control on the right edge of the form.

The segment display control represents all the segments of the schedule as a column of rectangles: the red rectangle indicates the currently displayed segment, while the green rectangle indicates the segment containing the currently executing unit operation.

To select the sequence segment to display

- Click the rectangle representing the segment.

  Tip If you hold the mouse pointer over a rectangle, the segment display control shows the range of method steps in that segment. ▲

To select the segment display mode

- Click the  button to switch between display modes: either scrolling, which keeps the currently executing unit operation displayed, or static.

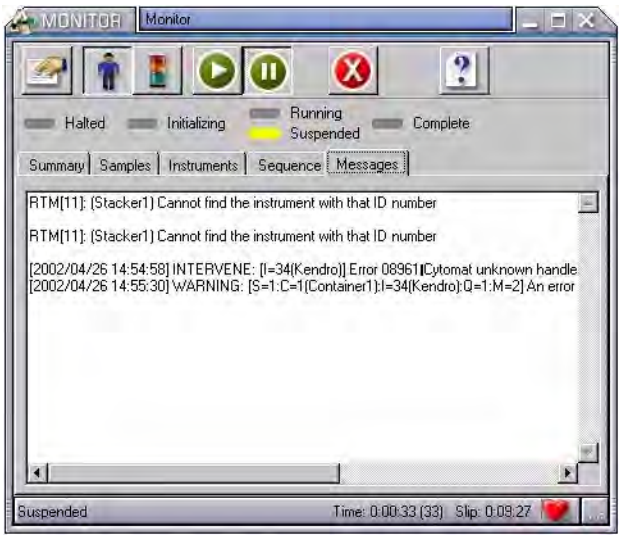
To view the unit operations in the segment

Use the scroll bar at the right of the list to scroll through the unit operations in the currently displayed segment of the schedule.

Note You can use the Sequence tab to set automatic pause points in the schedule, as described in "Inserting Pause Points into a Run" on page 6-19. ▲

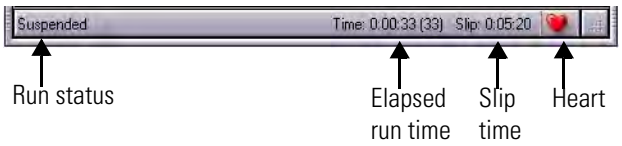
Messages Tab

The **Messages** tab displays all warning and error messages issued by the system. Keep this tab selected during a run to view the messages as they occur.



Status Bar

The Status bar displays the following information about the system:



- **Status**: the status of the system. This will be either Running, Suspended, or Halted.
- **Time**: the time that the system has been running since the run was started. This does not include time while the run was Suspended.
- **Slip:** the time the system was Suspended during a run
- **Heart**: A beating heart indicates communication between the POLARA system computer and the lab system.

Starting a Run

After setting up the run, you are ready to start the run.

Note The following procedure assumes you have powered up your system as described in the system manual. ▲

To start a run

1. Take the following steps:

   - If you are restarting a recovered run, ensure that the containers that were in process when the run was aborted are where POLARA expects them to be, based on the handling selected for the interrupted operations. For more information about handling options, see "Handling Interrupted Operations" on page 6-28.
   - Ensure that all laboratory personnel are clear of the system, and that there are no objects on the table that may obstruct mover motion.
   - If the lab system uses an articulated robot, ensure robot arm power is on.

2. In the **Monitor** window click **Start/Continue**.

   On DM systems, the run begins processing immediately.

   On AR systems, if the robot is not currently within the safe start zone, the Robot Start Position dialog box appears. Respond to the dialog as follows:

   a. Move the robot to a safe location using CROSnt.

   b. Click **Start Run**. The run begins processing.

   See "Setting the AR Start Position" on page 6-13 for details on setting and saving the start position.

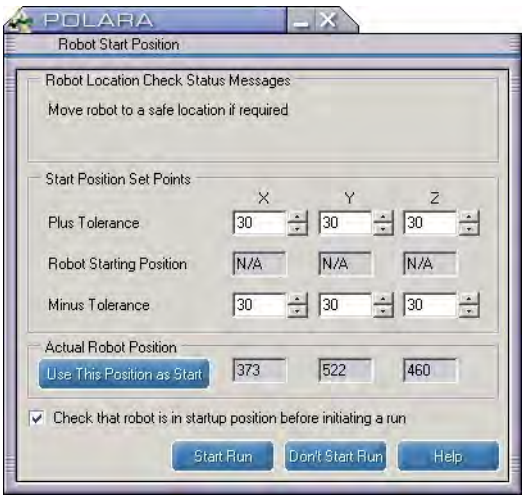
When a run completes, the Complete status indicator lamp on the toolbar turns blue, and the beacon turns red.

Setting the AR Start Position

On AR systems, POLARA needs to know that the robot is in a safe position before it starts a run. To do this, it checks the current robot position against against a defined safe start position at the beginning of each run.

The safe start position is a set of XYZ coordinates, with +/- tolerances, defining a three-dimensional zone within the robot's range of motion. Any position within this zone is considered a safe position from which to start a run.

You specify the safe start position using the Robot Start Position dialog box:



The Robot Start Position dialog box is opened in two ways:

- It opens automatically when you click Start/Continue on the Monitor window, if the robot is not currently within the safe start position zone.
- It opens when you select Robot Start Position from the View menu.

Note The buttons on the dialog box differ depending on how it was opened. The illustration above shows the buttons that are displayed when the dialog box is opened automatically upon starting a run. ▲

To set and save the robot start position

1. If the robot is not in a safe position to start a run, move it to a safe position using CROSnt.

   Tip The Actual Robot Position fields show the XYZ coordinates for the current position of the robot. ▲

2. When the robot is positioned, click Use This Position as Start. The current coordinates are copied into the Robot Starting Position fields.

3. If necessary, modify the Plus Tolerance and Minus Tolerance settings to define the zone around the start position, within which the robot is safe to start a run.

4. Make sure that the checkbox above the buttons is enabled so that the system always checks the position of the robot before starting a run.

5. Save the settings by doing one of the following:

    - If the dialog opened when you clicked Start/Continue on the Monitor window, click Start Run to save the settings and start the run.

    - If you opened the dialog by selecting Robot Start Position from the View menu, click OK to save the settings.

Monitoring a Run

As the run progresses, you monitor the run for warnings or errors. When POLARA encounters a condition that warrants a warning or error, it alerts the operator using the beacon, issues a message in the **Messages** tab of the **Monitor** window, and may display an error message box with several options.

Understanding the Beacon

The beacon is an external device that displays the status of the run. As you perform a run, you can look at the beacon to quickly check the status of the run. The beacon uses three colored lights that are lit individually or in combination as follows:

Green

The green beacon light indicates the system is running, as follows:

- **Solid green**: The run is proceeding normally and no warnings have been issued.
- **Solid green, solid yellow**: The system is running, but warnings have been issued. Depending on the type of warning, the run may become suspended if corrective action is not taken.
- **Solid green, flashing yellow**: The system is running, but an error occurred that has not been cleared.

Yellow

The yellow beacon light indicates something is wrong, as follows:

- **Solid yellow**: A warning has occurred.
- **Flashing yellow**: An error has occurred.

Red

The red beacon light indicates the system is not running, as follows:

- **Solid red**: The run has completed.
- **Flashing red**: Initialization has completed without fault. The system is in a suspended state.
- **Flashing red, flashing yellow**: An error has occurred and the run was suspended. Once corrective action is performed, the run can be continued.

Understanding Warnings and Error Messages

POLARA issues warnings and error messages when it encounters a condition that should be noted or corrected.

Warnings

POLARA issues a warning when it encounters a condition that should be noted and reported but is not serious enough to warrant suspending the run. It alerts operators to a warning in the following ways:

- A warning appears in the **Messages** tab of the **Monitor** window.
- The beacon turns yellow.
- The warning status indicator turns yellow.

Error Messages

POLARA suspends the run when it encounters an error that requires intervention. It alerts operators in the following ways:

- Depending on the type of error, a message box may appear on the screen.

Figure 6-29. An example POLARA run-time error message

The message box displays the specific error POLARA encountered and offers optional steps you can take to resolve the issue.

- An error message appears in the **Messages** tab of the **Monitor** window.

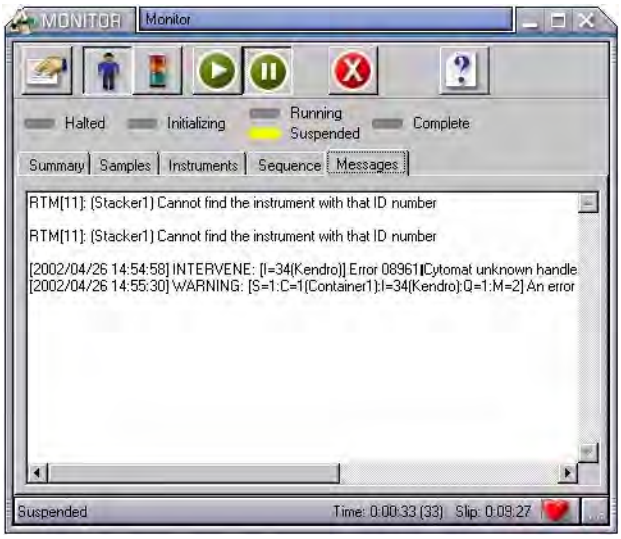


Figure 6-30. POLARA displays some error messages in the Message tab.

- The beacon flashes yellow.
- The error status indicator turns red.

Suspending a Run

POLARA suspends a run when it encounters an error that requires operator intervention. You may suspend a run to attend to a warning issued by POLARA, to adjust a container, or to refill reagents.

WARNING Suspending the run does not immediately stop the motion of the robot and other instruments. Nor does it remove power. Always suspend a run and press an E-Stop before reaching into the system. ▲

When you or POLARA suspends a run, the following occurs:

- The system completes any unit operations in progress but does not begin new unit operations.
- The suspended status indicator lamp turns yellow, and remains yellow until you click either **Start/Continue** or **Halt**.
- The run time stops accumulating and slip time (the amount of time that the run slips from the scheduled time) accumulates until the run is continued.
- The beacon turns yellow.

To suspend a run

- Click **Suspend** in the Monitor toolbar.

Continuing a Suspended Run

When a run has been suspended, you must ensure it can safely continue before resuming the run in POLARA.

To resume a suspended run

1. Ensure that all laboratory personnel are clear of the system, and that there are no objects on the table that may obstruct the motion of the robot or other instruments.
2. Ensure the robot is in a safe position.
3. Click **Start/Continue**. The run continues with the next unit operation in the schedule. When the run restarts, the running status indicator turns green.

Inserting Pause Points into a Run

Besides allowing you to pause a run manually, POLARA automatically suspends a run when it encounters a pause point.

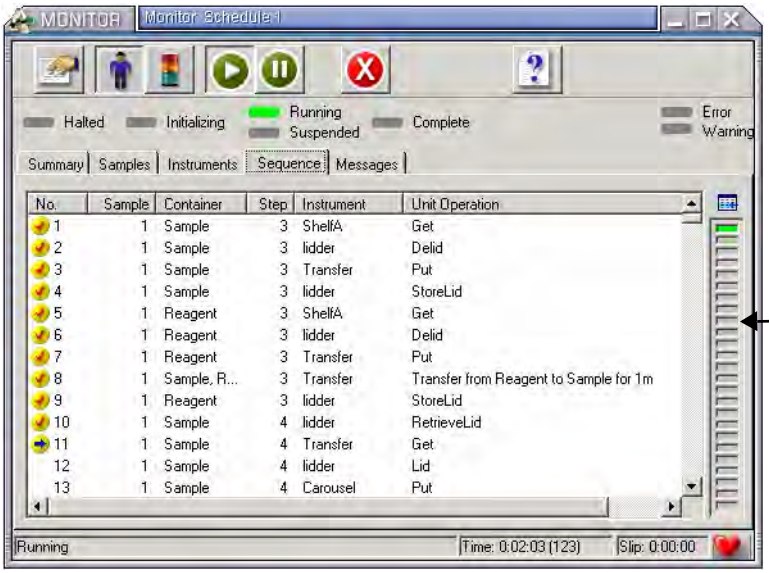
You use pause points to force the system to suspend in such cases as:

- When operators must refill reagents
- When operators must add containers to the system at the last possible moment (such as containers that must be kept refrigerated as long as possible before being used in a run)

Note You can insert pause points anywhere except on Convey operations. ▲

To insert a pause point

1. With the system running, select the Sequence tab in the Monitor window.



2. If necessary, use the segment display control to find the unit operation before which you want the system to pause.

3. Right-click on the line for that unit operation and choose **Insert pause** from the pop-up menu. POLARA displays the information for the operation in red.

To re-start the system after a pause point

1. Ensure that all laboratory personnel are clear of the system, and that there are no objects on the table that may obstruct the motion of the robot or other instruments.

2. Ensure the robot is in a safe position.

3. Click **Start/Continue**. The run continues with the unit operation at which you inserted the pause.

To search for pause points

1. Select the **Sequence** tab in the **Monitor** window.

2. Right-click on any unit operation and choose **Display next pause** or **Display previous pause**. POLARA finds the next or previous pause in the schedule and displays the subset of operations that contain the pause.

To remove a pause point

1. Select the **Sequence** tab in the **Monitor** window.

2. Use the Subset Selector or the **Display next/previous pause** feature to find the unit operation that is marked for the pause you wish to remove.

3. Right-click on the unit operation and choose **Remove pause**. POLARA displays the information for the operation in black.

Adjusting Containers or Reagents During a Run

During a run, you may need to adjust containers or refill reagents inside the system.

WARNING Always suspend the run and press an E-Stop button before working with the system. This removes arm power to the robot and prevents it from moving. ▲

WARNING Pressing the E-Stop only removes power from the robot. Be sure you understand what instruments are still operational before you reach into the system. ▲

To adjust containers or refill reagents

1. Click **Suspend** in the **Monitor** window.
2. Wait until the movers stop and the beacon light flashes yellow, indicating the run is suspended.
3. Press an E-Stop button. You may now enter the system workspace.
4. Adjust the containers or reagents on the table as necessary.
5. Recover the system.

Adding Samples to a Run

POLARA 2.1 and higher enables you to add samples to a run in progress.

WARNING Adding samples to a run controlled by a schedule that has not been specifically developed and tested to support the addition of samples may result in damage to equipment, or damaged or lost samples, or both! ▲

To add samples to a run, you schedule them as a new batch in the schedule that controls the run. POLARA then replaces the unexecuted part of the run's schedule with the updated schedule.

To add samples to a run

1. Open the schedule controlling the run by clicking on its name in the left pane of the POLARA main window.

2. Click **Append** to add a new batch to the schedule. The schedule editor displays a blank batch form.

3. Take the following steps:

   - Select a **Method** to be used with the new samples.
   - Enter the **Number of Samples** to be processed.
   - If necessary, enter sample and batch staggers required to optimize the run.

   Note The researcher or administrator who normally creates schedules should provide the operator with appropriate sample and batch staggers for the schedule. ▲

4. Click **Schedule**. The scheduler generates a new Gantt chart for the schedule.

5. Click **Commit** to replace the unexecuted portion of the running schedule with the new schedule. POLARA displays in the Monitor's **Message** tab the number of the sequence step at which the switch-over to the new schedule will occur.

POLARA then displays the **Setup for Run** window, which lists the containers needed for the added samples and the nests to place them in.

6. Wait until the mover serving the instrument to which the containers must be added has finished getting or putting a container.

7. Add the containers to the instrument, performing any instrument-specific steps that might be necessary. For example, you might have to power off the instrument before you add the containers, and bring it back online afterwards. For details, refer to thesystem manual .

8. Click **Start Run** on the **Setup for Run** window.

Note It is not necessary to suspend the run to add containers. In fact, suspending the run might add so much slip time that POLARA will not be able to switch over to the new schedule. If this occurs, you have to re-schedule the added samples. ▲

Halting a Run

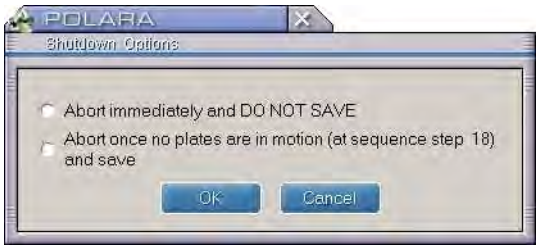
WARNING The Halt button does not immediately stop the motion of the robot and other instruments. Nor does it remove power. In an emergency use the E-Stop button. ▲

CAUTION Do not use the Halt button unless you want to end a run. Use the Suspend button to pause the system briefly. ▲

To halt a run

1. Choose **Halt**. A dialog box appears that prompts you to confirm that the run should be halted.

2. Choose **Yes** to halt the run. Container transport motion stops and the run ends. The halted status indicator lamp turns red, and the beacon turns red. The **Shutdown Options** dialog box appears:



3. Choose the appropriate Shutdown Option:

   - If you have no intention of attempting to recover the run, click Abort immediately and DO NOT SAVE.

   - If you wish to wait until all motion is completed, and intend to recover the run later, click Abort once no plates are in motion (at sequence step *n*) and save.

4. Choose **OK**. Container transport motion stops according to the specified Shutdown Option and the run ends. The halted status indicator turns red, and the beacon turns red.

Recovering an Aborted Run

If you saved run information when aborting a run, you can recover the run.

Note Until you recover an interrupted run, or explicitly command POLARA to abandon the saved run information, POLARA displays the run recovery options every time you start it or open the workspace containing the schedule that controlled the interrupted run. ▲

Whether you choose to recover or abandon the run depends on factors that include the following:

- **The duration of the interruption**. If the contents of your samples are sensitive to delays in processing, you must consider whether the duration of the interruption would invalidate the data from the samples that have not completed processing.
- **The number of samples remaining**. If only a small percentage of the total number of samples remains to be processed, it might be simpler to schedule them into a new run.

For more information about recovering a run, see the following topics:

- "Understanding Run Recovery" on page 6-25
- "Starting Run Recovery" on page 6-27
- "Handling Interrupted Operations" on page 6-28
- "Restarting the Run" on page 6-30

Understanding Run Recovery

When POLARA saves run information, it stores the operations that were still in progress when the run was aborted. Before you can resume a run, you must tell POLARA how to handle the interrupted operations, and then ensure that the containers affected by these operations are where POLARA expects them to be when the run resumes.

CAUTION If you resume a run without ensuring that the containers in process are where POLARA expects them to be, you may lose run data, damage equipment, or spoil samples. ▲

POLARA provides the following options for handling interrupted operations:

- **Resume** the operation. POLARA takes into account the time that has elapsed since the run was aborted, and adjusts the operation's times accordingly. For example, if a container had incubated for 10 minutes of a scheduled 30 minute period when the run was aborted, and if the run recovery was started 10 minutes later, POLARA sets the time remaining in the incubation operation to 10 minutes (30 - (10 +10) =10). **POLARA assumes the container is in the same place it was when the operation was interrupted.**

- **Restart** the operation. POLARA ignores the elapsed operation time and re-issues the command to perform the operation. **POLARA assumes the container is where it should be when the operation starts.**

- **Skip** the operation. POLARA assumes the operation is complete. When the run resumes, POLARA will issue the command to perform the next operation on the container as originally scheduled. **POLARA assumes the container is where it should be when the operation has completed.**

Note When a run is aborted and saved, POLARA may not have detailed knowledge of the status of each interrupted operation. For example, if an interrupted operation was a Run Program operation for a liquid handler, POLARA may not know what operations the program was performing on the container. Therefore, you must consider carefully which of the three options for handling interrupted operations will result in the most safe and effective way to resume the run. ▲

Starting Run Recovery

When POLARA detects saved run information, it displays a **Saved Run Detected** dialog box. The options available depend on whether POLARA detected the run information when it started or when you opened the workspace containing the schedule for the interrupted run:

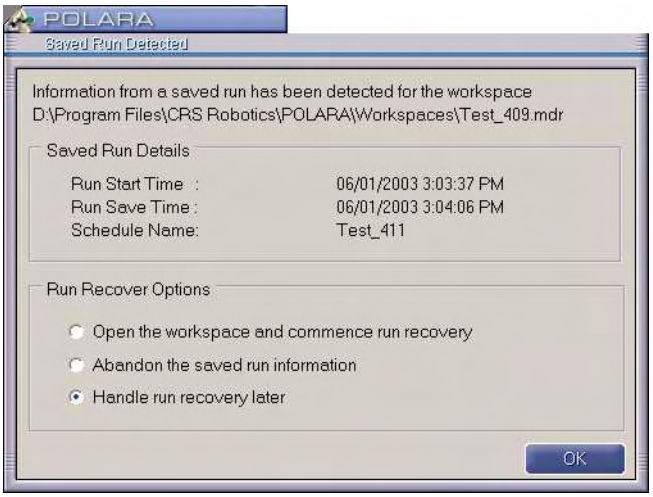


Figure 6-31. Saved run detected when starting POLARA

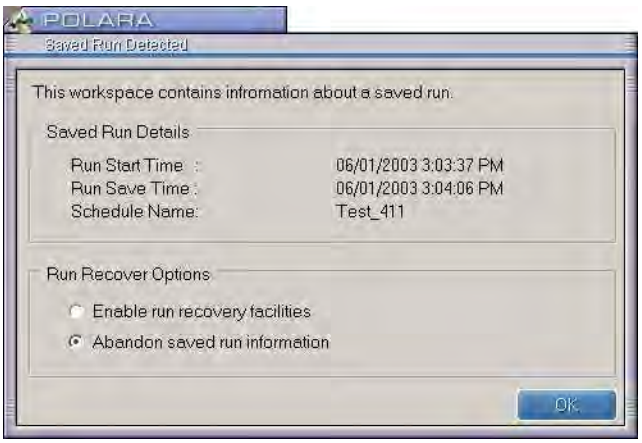


Figure 6-32. Saved run detected when opening a workspace

The **Saved Run Detected** forms enable you to start run recovery, postpone it, or abandon the saved run information.

To start run recovery

1. Select Open the workspace and commence run recovery or Enable run recovery facilities.

2. Click OK. POLARA displays the **Run Recovery - Containers In Progress** form. For details on using this form, see "Handling Interrupted Operations" on page 6-28.

To postpone run recovery

1. To postpone recovering the run, select **Handle run recovery later**. At the start of the next run, POLARA asks you whether to overwrite the information it saved on the previous aborted run. If you choose not to overwrite the information, you cannot save information on the new run.

2. Click OK.

To abandon saved run information

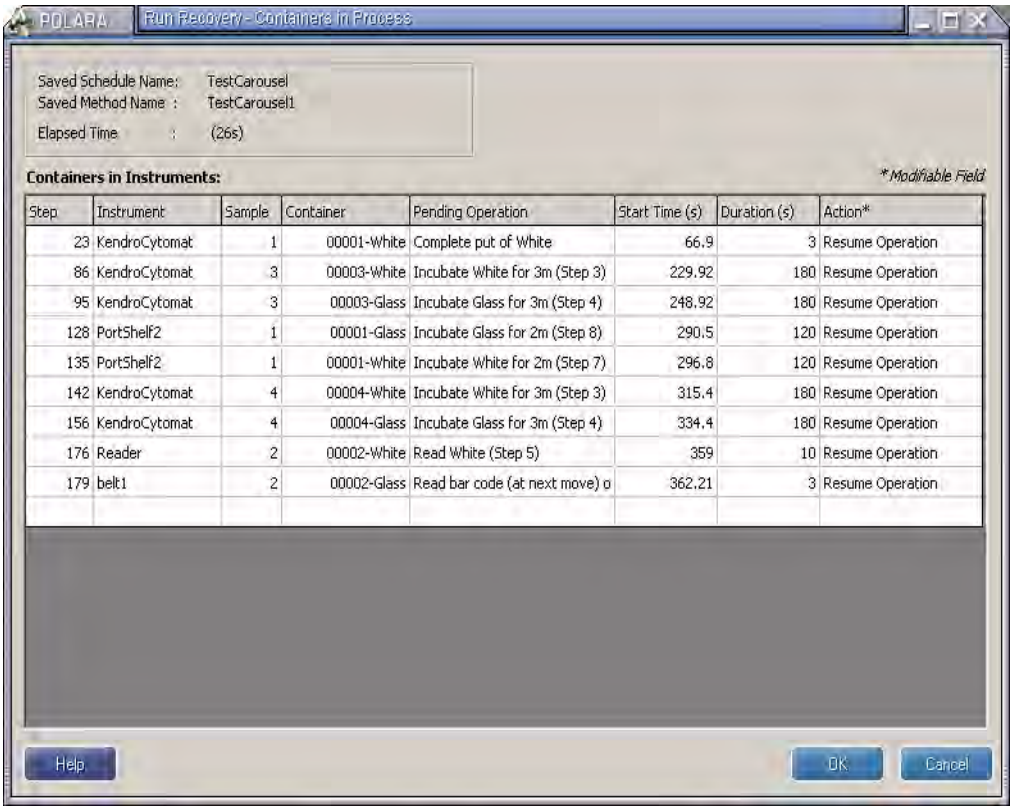
1. If you do not wish to recover the run at any time, select **Abandon saved run information**. POLARA will delete the saved run information.

2. Click OK.

Handling Interrupted Operations

When you choose to recover an interrupted run, POLARA displays the Run Recovery - Containers in Process form. This form displays a table showing the operations that were in progress when the run was aborted.

It consists of the following columns:



- **Step**, which is the step in the schedule where the interrupted operation was taking place
- **Instrument**, which is the instrument the container was in when the run was aborted
- **Sample**, which is the number of the sample to which the container belongs
- **Container**, which is the container on which the operation was being performed
- **Pending Operation**, which is the operation that was interrupted when the run was aborted
- **Start Time**, which is the time (in seconds) at which the interrupted operation started
- **Duration**, which is the duration (in seconds) defined for the operation
- **Action**, a drop-down list from which you select how POLARA should handle the interrupted operation

To handle interrupted operations

1. Click the **Action** cell in the **Containers In Instruments** table and select one of the following ways to reschedule the operation in the drop-down list:

   - **Restart Operation**. When you restart the run, **POLARA assumes the container is in the source nest for the operation**.
   - **Resume Operation**. When you restart the run, **POLARA assumes the container is where it was when the operation was interrupted**. It shortens the duration of the operation in the schedule's Gantt chart to account for the time the container has spent in the instrument before and during the interruption.
   - **Skip Operation**. POLARA marks the operation as completed. When you restart the run, **POLARA assumes the container is in the destination nest for the operation**.

2. Take one of the following steps:

   - Click **OK** to continue run recovery.
   - Click **Cancel** to close the **Run Recovery** form and postpone run recovery. To resume run recovery, choose **Run Recover** in the Workspace menu.

Restarting the Run

When you click **OK** in the **Run Recovery** form, POLARA prepares to restart the run. But before you click **Start/Continue** in the Monitor, you must make sure that each container that was in process when the run was aborted is where POLARA expects it to be, based on the options you selected in the **Run Recover** form.

CAUTION If you resume a run without ensuring that the containers in process are where POLARA expects them to be, you may lose data, damage equipment, or spoil samples. ▲

For information on how instrument servers handle interrupted operations, refer to the instruments' interface guides.

For more information about how to handle interrupted operations for the instruments in your system, refer to the system manual.

To restart the run

1. Ensure that the containers that were in process when the run was aborted are where POLARA expects them to be, based on the handling you selected for the interrupted operations. For more information about handling options, see "Handling Interrupted Operations" on page 6-28.

2. Click **Start/Continue** to resume the run.

Finishing Up After a Run

After a run completes, the red light on the beacon will be lit, and the complete status indicator turns blue.

Once you have confirmed the run has ended, remove containers and reagents from the system as necessary.

Using the Remote Monitor

The Remote Monitor allows you to view the progress of a run from a computer on the same network as the lab system computer.

The Remote Monitor provides the same real-time run information that the **Monitor** window on the lab system computer provides. It does not, however, allow you to control the operation of the run.

Note To use the Remote Monitor, your POLARA license must include the Remote Monitoring capability. ▲

To start the Remote Monitor

1. Ensure that both POLARA and CROSnt are running on the lab system computer.

2. On the remote computer, in POLARA's **Tools** menu, choose **Remote Monitor**. The **Get Remote Computer Name** dialog box appears:

3. Enter the name of the lab system computer in the **Enter a New Remote Computer Name** text box, or select the name from the **Choose a Known Remote Computer Name** list box. Click **Connect**. POLARA attempts to connect with the lab system computer.

   If log-in is unsuccessful, the **Remote Login Information** dialog box appears. Enter your **User Name** and **Password** and click **OK** to gain access to the lab system computer.

4. After a successful log-in to the lab system computer, the Remote Monitor Window appears.

To close the Remote Monitor

- Click **Close**.

Glossary

action message An intervene message that does not activate the beacon.

See also intervene message.

administrator A POLARA user with unrestricted access to the POLARA Windows application. A POLARA administrator must be a member of the Windows Administrator group on the same computer. A POLARA administrator typically performs system maintenance tasks, such as managing user accounts, performing backups, managing workspaces, managing system configuration, and performing basic troubleshooting. A POLARA administrator may also act as an integrator, adding and removing hardware from the lab system.

application shell (ash) A utility available at the CROS command prompt that you can use to move the robot arm, teach locations, and set values to other variables. You start it by entering ash at the CROS command prompt.

AR (articulated robot) See articulated robot (AR).

arm See articulated robot (AR).

articulated robot (AR) A mechanical device modeled on a human arm that, in a lab system, carries containers from place to another. It can be mounted on a track to move containers from one instrument to another.

See also POLARA AR system.

ash See application shell (ash).

assigning dispositions The process of defining how POLARA should re-schedule containers that were in process when a run was interrupted.

attended/unattended mode You use attended mode when the system is attended by an operator. If an error occurs, POLARA logs it and reports it to the Message box of the Run window, it suspends the run, and it starts the beacon's red light flashing.

You use unattended mode when the system is not attended by an operator: overnight, for example. If an error occurs, POLARA logs it, reports it to the Message box of the Run window, and repeatedly attempts to recover from the problem. If recovery is successful, the run continues.

barcode reader An optical sensing device that reads the unique bar code attached to the side of a container. POLARA includes a barcode reader interface.

batch A set of samples processed through one method during a run and having the same scheduling parameters. A run can process more than one batch, each using a different method and/or different scheduling parameters.

See also method and sample.

batch stagger A batch stagger delays the start of an entire batch.

A batch stagger by step delays the start of the batch until the previous batch has started the specified step (unit operation).

A batch stagger by time delays the start of the batch until the specified time has passed. The time period runs from the start of the previous batch.

See also sample stagger.

beacon A set of green, yellow, and red lights on a pole used to indicate system status to the operator. The lights are independent of each other. Light

activity resulting from messages can be cleared by the operator using the traffic-light icon of POLARA User Interface.

buffer time The length of time, allowed by the researcher, that a step can take beyond the regular step duration, without causing a problem to the sample. For example, an incubation with a regular step duration of 6000 seconds and a buffer time of 120 seconds allows a step to take from 6000 to 6120 seconds.

See also step duration.

calcloc A location variable used to store the locations immediately adjacent to nestloc locations. If the robot arm is moving stacker magazines, you teach two of these locations: calcloc[n,0] and calcloc[n,1]; otherwise, just one: calcloc[n,0].

cap A cover for a test tube or vial. A cap is not a sub-container.

See container.

carousel A rotating shelving unit with racks of shelves facing radially outward from an axis of rotation. A typical microplate carousel carries eight racks or columns, each with 15 shelves.

A carousel can be incorporated in an instrument such as an incubator. This provides a high number of nests for the instrument's unit operation.

class The general identification of a component by its type or model. In contrast, an instance identifies an individual robot, instrument, or utility.

For example, a workspace can have two instances (e.g. "incubator1" and "incubator2") of the same class (e.g. "incubator").

See also instance.

client A process that issues a request to other processes for a particular service. For example, instrument servers are clients of the robot administration server.

See also server.

configuring, configuration The action of entering parameters or specific settings, to set up a component for use. After installing an interface component in POLARA, you configure the component by setting parameters under Change Physical Settings.

congestion Condition where samples must wait for the availability of a station, resulting in accumulation of traffic and non-uniform processing. The condition can be alleviated by adding (pooling) instruments or adjusting staggers.

See also pooling, batch stagger, and sample stagger.

container A physical object that is transported by the robot and acted on by an instrument. Examples: microplate, test tube, vial, stack.

A container can have sub-containers. For example, a stack can have 20 microplates. Therefore, a container is not always a vessel that directly holds the substance used in the method.

A sample can have a number of containers.

controller See robot controller.

CROS (CRS Robotics robot operating system) An operating environment that RAPL-3 processes run on.

CROS-500C The version of CROS that runs on a C500 controller (used in a POLARA AR system). Command line access to it is through the terminal window of ActiveRobot Terminal.

When you power up the controller, CROS-500C starts up automatically.

CROS for Windows NT (CROSnt) The version of CROS that runs on Windows on a personal

computer. Command line access to it is through the CROS Command Prompt window.

When you perform a run, POLARA starts up CROSnt on the PC. On POLARA AR systems, you can start CROSnt from your PC's desktop to teach locations.

daemon A process that runs in the background and performs a task when necessary. In POLARA, examples of daemons include: SimSockD (simple socket daemon for computer-controller communication) and fastacid (fast advanced communication interface daemon for terminal communication).

See also server.

de-lidder See lidder

dependency See step dependency.

de-stacker See stacker magazine.

digital I/O card A PLC provided on a card that fits into a slot in the lab system computer. The card has a 50 pin connector to an expansion box or terminal block to which the lab system hardware is wired.

dispatcher The POLARA scheduler generates a dispatcher file from the method and scheduling information that you entered. When you start a run, the dispatcher file (with many other POLARA components) is executed. The dispatcher then issues the series of commands that implement the many unit operations on the samples, according to the schedule, until the run is finished.

distributed motion (DM) A container transport architecture designed by Thermo LACI that dedicates a container mover to each instrument in a lab system and uses a conveyor to move containers between each container mover. Distributed motion is the converse of the centralized motion provided by an articulated robot that serves every instrument in the lab system.

See also articulated robot (AR), DM controller, linear plate transport (LPT), and local mover (LM).

DM (distributed motion) See distributed motion (DM)

DM controller A specialized computer that controls the DM movers. A master DM controller controls the LPT and up to three local movers; a slave DM controller controls up to four local movers.

See also linear plate transport (LPT), and local mover (LM).

epilogue file A file that contains commands used when finishing up after a run. The file is automatically called by the dispatcher, after all unit operations are performed. The file is written, compiled, and placed in a directory of CROSnt by the integrator.

See also prologue file.

error message Signals occurrence of an event that is serious enough to require the operator's attention, if the run is in attended mode. Suppose, for example, that an instrument is not responding to communication. If the run is in attended mode, POLARA suspends the run and displays an error message. If the run is in unattended mode, however, POLARA attempts to deal with the error without suspending the run. In both cases, the beacon's red light starts flashing.

error notification service A utility installed in the workspace by the integrator and configured in Change Physical Settings. Depending on configuration, when a warning, error, intervene/action, or fatal event occurs, the system can activate an autodialer that places a telephone call or similar communication to off-site personnel. The error notification service can be configured to send an off-site message even when the system is in attended mode. See also messages

fatal message Signals an occurrence of an event from which recovery is not possible.

Gantt chart The chart generated by POLARA's scheduler showing the sequence and duration of unit operations performed on each sample processed by a method.

See also unit operation.

general purpose I/O (GPIO) The 32 digital I/O points provided by the C500C robot system controller. The devices are wired, with a termination block, to the GPIO connector on the back panel of the controller. Communication between the GPIO on the controller and the GPIO PLC server on the computer is routed through the SimSockD server that handles communications between the controller and the lab system computer.

See also general purpose I/O (GPIO), PLC server, digital I/O card, and PLC space.

get An action of the robot in which the robot gets, or retrieves, a container from the nest of an instrument.

A get action and a put action are always paired. The scheduler adds a get action and then a put action before each unit operation.

The system integrator specifies the duration of a get for an instrument when changing the physical settings of the instrument instance.

The system can check whether the robot has a container in its gripper before performing the get and put. This checking is configured by the system integrator when changing the physical settings.

See also put.

GPIO See general purpose I/O (GPIO).

gripper See robot gripper and nest.

high throughput screening A process used in pharmaceutical and other industries where one substance is screened against another substance (tested for an outcome such as a chemical reaction or biological growth or atrophy), and where this screening involves high throughput (a large number of substances processed in a time period).

hotel A static shelving unit containing a vertical array of nests, usually used for holding microtiter plates or lids. It can be permanently fastened to the lab bench, temporarily attached to a stand on the lab bench, or temporarily attached to one of eight hotel holders of a carousel.

I/O input/output

incubator An instrument used to provide controlled environmental conditions, such as temperature, humidity, or C02. An incubator resembles a small refrigerator, and usually contains a carousel to house samples. It may have one large door, or many small sliding doors for the robot to access each nest.

In POLARA User Interface, the class "incubator" is often used to form an instance of any instrument that has an array of nests, such as a carousel for storage, a hotel, or a true incubator.

info message Signals occurrence of a regular event. Example: completion of a unit operation. Not reported to the operator.

An attention message is an info message with a prefix, attn.

installation screen The main screen from the POLARA CD where you choose to install POLARA User Interface, CROS for Windows NT, CROS-500, or Robcomm3. Sometimes called a splash screen.

instance One incorporation of a lab system component class. Each instance is identified with a unique name.

See also class.

instrument server An instrument server is a program that controls an instrument, as commanded by the dispatcher. To carry out unit operations, the server sends messages to the

instrument, either serially or digitally through a PLC, and sends requests to RAD for the robot.

See also dispatcher.

instrument A physical device that performs at least one unit operation on a container, affecting the container's contents in some biologically or chemically significant way or generating data from the contents. Examples of instruments include incubators, dispensers, and readers.

For an instrument to be used in POLARA, it must be installed as a component. The component can then be used to define a specific instrument instance in a workspace and a profile.

See also peripheral and nest.

intermloc Type of location variables, one or more of which is used to move between safeloc locations and calcloc locations.

See also safeloc and calcloc.

inter-process communication In POLARA, clients and servers communicate with each other using sockets. A socket is a file-system device that allows two-way communication between client and server. A socket allows one server to connect at one end of the socket and many clients to connect at the other end.

intervene message Signals occurrence of an event that requires the operator's attention, whether the run is in attended mode or not. Suppose, for example, that an instrument needs to be restarted. POLARA suspends the run and displays the intervene message. The Run beacon's red light flashes. See also action message.

lab system computer The computer that controls the POLARA lab system.

See also off-line workstation.

lid A cover for a microplate. A lid is not a sub-container.

See container.

lidder A device or instrument used to lid and de-lid microplates. Some lidders have shelves to store lids while the plates are in instrument nests. Some lidders can lid/de-lid several plates at once.

See also shadow lidder with shelf.

linear plate transport (LPT) The central conveyor that moves containers (usually microtitre plates) between one local mover and another in a distributed motion lab system.

See also DM controller and local mover (LM).

LM (local mover) See local mover (LM).

local mover (LM) A container transport device that moves containers (usually microtitre plates) between an instrument and the central conveyor.

See also DM controller and linear plate transport (LPT).

location (general) A point in space with a specific orientation known to the robot.

With a Thermo CRS robot, a location can be one of two types of locations. A cartesian location (cloc) stores information according to a cartesian (straight-line axes meeting at right angles) coordinate system. A precision location (ploc) stores information according to pulse counts of the joints of the robot arm.

The information about the point is placed in a location through the process of teaching.

See also teaching and location (instrument).

location (instrument) In a POLARA system, each instrument has a set of locations for robot motion towards and away from its nest. If the instrument

has more than one nest, there is a set of locations for each nest.

See also teaching and location (general).

log file See master log.

LPT (linear plate transport) See linear plate transport (LPT).

master log A file containing time-stamped log entries that record the unit operations completed, warnings, and errors during a run.

See also messages.

messages POLARA has five kinds of messages: info, warning, error, intervene, and fatal. POLARA displays warning and error messages in the Messages tab of the Monitor Window.

Within a POLARA system, all five kinds of messages are logged in the RTM log file. The location of the log file is displayed in Options available from POLARA's View menu. You can read the log file with any text editor, such as Notepad.

See also action message, error notification service, fatal message, info message, intervene message, and Run-Time Manager (RTM).

method The chemical or biological process to be performed on a sample. In POLARA, a method is constructed of unit operations listed as steps to be performed in sequence.

A method is stored in a workspace and modified from within a workspace. The instrument instances in a profile determine which unit operations can be included in a method.

POLARA permits batches in a schedule to be processed with different methods, provided that the methods are based on the same profile.

See also batch and unit operation.

Mid-run addition A batch of samples added to the schedule controlling a run.

nest A position in an instrument where a container is placed. The get action retrieves a container from a nest and put action places a container in a nest.

nestloc A location variable that stores the position and orientation of the robot arm at the nest of an instrument or peripheral.

See also region.

off-line workstation A computer running the POLARA Windows application that is independent of (not connected to) the lab system. Used for developing methods and schedules. Can run under Windows NT, 98, 95 or 2000.

See also lab system computer.

operator A POLARA user restricted from all functions of POLARA, except the scheduler and the monitor. A POLARA operator must be a member of the Windows POLARA_OPERATOR group on the same computer as the POLARA Windows application. A POLARA operator typically sets up and performs runs.

parameter, instrument See property.

parameters, scheduling See scheduling parameters.

pause A temporary interruption in a running schedule. The POLARA operator can pause lab system operations (to refill reagents, for example) by clicking Suspend on the Monitor window, then resume operations by clicking Start/Continue. The operator can also insert any number of automatic (or programmed) pauses into the schedule, causing the system to suspend operations automatically at each specified pause point.

peripheral A physical device that performs at least one unit operation on a container, but which neither alters the contents of the container in any biologically or chemically significant way nor produces data from the contents. Examples of

peripherals include delidders, re-grip turntables and barcode readers.

See also instrument.

PLC See programmable logic controller (PLC).

PLC server The CROSnt program that controls a PLC, as directed by client instrument servers.

See also programmable logic controller (PLC), general purpose I/O (GPIO), digital I/O card, and PLC space.

PLC space An array in POLARA's memory to which all digital I/O points available in the lab system are mapped, so that POLARA can uniquely identify each point.

POLARA (programming architecture for laboratory automation) The suite of Thermo LACI software components that automate a lab system. It includes a Windows-based user interface and interfaces to lab system hardware.

POLARA AR system A POLARA system that uses a Thermo CRS articulated robot (AR) to move containers

POLARA DM system A POLARA system that uses Thermo CRS distributed motion movers to move containers

See also distributed motion (DM).

POLARA lab system, POLARA-based lab system Any lab system built using the POLARA software suite.

POLARA Windows application A component of, and the standard user interface to, a POLARA system. You use it to configure lab systems, develop methods and schedules, and perform runs.

pooling Configuring two or more instances of an instrument to appear as one instrument in POLARA. When pooled, the scheduler uses whichever instrument is empty when it needs to schedule a unit operation. This can increase throughput.

profile A set of instrument instances and containers. A profile can include all or only some of the instruments of a workspace.

A profile is stored in a workspace and modified from within a workspace. Only instances that are in the workspace can be added to a profile.

The instrument interfaces in a profile determine which unit operations are available to be used in methods.

programmable logic controller (PLC) An electronic device that provides digital input and output lines. Each line has two states: ON or OFF.

A PLC is used to manage digital devices that are parts of instruments such as valves for air cylinders opening doors or sensors indicating the presence of a plate. It is also used to manage non-instrument digital devices such as the beacon light or an autodialler for message notification.

POLARA supports several kinds of PLC: external industrial PLCs, such as those from Direct and Entertron; digital I/O that is part of the container transport system, such as general purpose I/O and end-of-arm I/O provided by Thermo CRS robot systems; and digital I/O cards that plug inside the lab system computer. Each PLC has its own PLC server.

See also general purpose I/O (GPIO), digital I/O card, PLC server, and PLC space.

prologue file A file that contains commands used when starting up a run, such as unparking the robot. The file is automatically called by the dispatcher during initialization, before any unit operations are performed. The prologue file is written, compiled, and placed in a directory of CROSnt by the integrator.

See also epilogue file.

property A characteristic of a component instance in the workspace and in a profile.

You usually specify the value of an instance's properties when creating the instance in the workspace or in a profile.

Unit operations in methods also have properties, such as load step dependency and required lid state.

See also unit operation.

put An action of the robot in which the robot puts, or places, a container in the nest of an instrument.

A get action and a put action are always paired. The scheduler adds a get action and then a put action before each unit operation. The get action gets the container from the instrument of the previous unit operation, and the put action puts the container in the required instrument.

See also get.

RAD See robot administration server (RAD)

RAPL-3 A high-level, block-structured, procedural language used for programming robot and other automation applications. Components of POLARA that run under CROSnt (such as instrument servers) are written in RAPL-3.

rc file A file automatically executed when CROS and CROSnt start up, similar to an autoexec.bat file in DOS. The rc file starts a number of processes.

When you run a schedule, POLARA inserts lines into the rc file of CROSnt to start processes, such as the RTM and RAD server, that are required by all instrument servers.

See also CROS (CRS Robotics robot operating system) and CROS for Windows NT (CROSnt).

reader An instrument that analyzes the contents of a container. For example, a reader might examine wells in a microplate for fluorescence, luminescence, or pH levels.

reagent A substance dispensed into containers as part of the method.

region A group of instruments on the lab bench.

In a simple bench layout, without regions, the robot moves freely between the safe location of one instrument and the safe location of any other instrument. Each safe location is positioned so that the robot avoids obstacles when moving from one safe location to another.

In a complicated bench layout, instruments and their locations are grouped into regions. Within a region, the robot moves freely between one safe location and another. To move from one region to another region, the robot moves through a specific path to avoid obstacles.

Motion between one region and another is managed by the Region Server, sometimes called the Region Manager.

See also location (instrument).

region server, region service See region.

re-grip station A device or instrument used to change the orientation of a plate held by the robot from portrait to landscape or from landscape to portrait. Many re-grip stations use a turntable to rotate the plate 90 degrees. If no turntable is used, the wrist of the robot must rotate before picking up the plate.

A re-grip station is usually a shadow instrument, which means you do not have to explicitly include a re-grip unit operation in your method. POLARA will add the step for you, based on the container orientation properties of the instruments your method uses.

See also shadow instrument.

remote monitoring A function of POLARA that enables the user of an off-line workstation on the same network as the lab system computer to monitor the progress of a run. The user cannot control the run from the off-line workstation.

researcher A POLARA user restricted from changing physical settings in a workspace. A POLARA researcher must be a member of the Windows POLARA_RESEARCHER group on the same computer as the POLARA Windows application. A POLARA researcher develops methods and schedules, and might also set up and perform runs.

reserve option A property that sets whether only the sample that last occupied a particular nest may occupy it later. It is used to prevent deadlock in a method where an instrument is used multiple times. It is also used to preserve the input order of a carousel, hotel, or shelf, so that the containers always return to the same nests.

robot administration server (RAD) A POLARA component, common to all POLARA systems, that coordinates the processing of requests made by the various instrument servers for get and put operations. RAD ensures that only one instrument server has control of the robot at one time and that an instrument server completes its use of the robot before control is of the robot is given to another instrument server.

RAD also keeps track of any sample in the robot gripper.

RAD passes back the status of the robot for the Monitor window of POLARA User Interface and for the log files.

Communication between RAD and the robot is handled by SimSockD.

robot controller The specialized computer that controls and supplies power to the robot arm and optional track. It also provides digital I/O.

See alsogeneral purpose I/O (GPIO).

robot gripper The device at the working end of the robot arm that holds containers. It can be fitted with special fingers for gripping specific containers (microplates, tubes, crucibles, stacks). It is powered by a servo-electric motor (with most lab systems moving plates) or pressurized air (with systems moving stacks).

robot system The transportation system used to move containers from one instrument to another.

robot track A linear device, with a saddle to hold the robot arm, that enables the entire arm to move along an additional axis. A robot track is usually mounted in a lab table to enable the arm to move along the length of the table.

RTM log file See messages.

RTM See Run-Time Manager (RTM)

run recovery The process of restarting a run that was aborted. If the operator chooses to recover an aborted run, he or she must specify how POLARA should handle each container that was in process when the run was aborted. POLARA then generates a new schedule from which to resume the run.

Run-Time Manager (RTM) A POLARA component, common to all POLARA systems, that monitors and controls the operation of the instrument servers. The RTM, which runs under CROSnt, logs and time-stamps the beginning and end of each unit operation. It also logs any warnings or errors from the dispatcher, instrument servers, or other servers. It manages run-time operations, and is the only CROSnt component that communicates directly with the POLARA Windows application.

safeloc A location variable that stores the position and orientation of the robot arm at a position near an instrument nest that enables it to move freely from this location to the safeloc location of any other instrument, avoiding any obstacles as it moves.

sample The set of containers required for a chemical or biological protocol, or method. A method in POLARA consists of the sequence of unit

operations to be performed on one sample. A batch in POLARA consists of the set of samples on which a single method is performed.

The number of containers required for a single sample varies, depending on the method. A screening method might only require one microplate to store each sample. A DNA amplification method, on the other hand, might require several microplates for each sample: a mother microplate and several daughter microplates.

You add the container instances required for the samples in your methods in the Profile window.

See also container and method.

sample stagger A sample stagger delays the start of each sample in a batch.

A sample stagger by step delays the start of each sample in the batch until the previous sample has started the specified step.

A sample stagger by time delays the start of each sample in the batch until the specified time has passed. The time is from the start of processing of the last sample.

schedule The timed sequence of all unit operations (with get and put actions) for all samples being processed in a run.

A schedule is stored in a workspace and modified from within a workspace.

See also dispatcher.

schedule extension A batch of samples added to the schedule controlling a run.

scheduling parameters Factors that affect a schedule, entered by the operator to make processing more uniform and to avoid log jams.

The two types are batch stagger and sample stagger. Each of these staggers can be adjusted by step or by time.

See also batch stagger and sample stagger.

serial robot protocol (SimSockD) See SimSockD (simple socket daemon).

server A process that provides a service requested by a client. In POLARA, a server manages an instrument, a robot, a PLC, or provides utility services.

See also client and daemon.

shadow instrument An instrument whose unit operations are automatically inserted into methods by the POLARA scheduler, rather than by the user. The scheduler displays the added unit operation in the Gantt charts it generates.

shadow lidder with shelf An instrument that lids and de-lids microplates. It is not necessary to insert a lid/de-lid unit operation in a method. The scheduler knows when a lid/de-lid is necessary, according to each instrument's specifications, and automatically inserts this unit operation when it schedules a method.

shadow re-grip station An instrument that changes the orientation of a plate from portrait to landscape, or landscape to portrait. It is not necessary to insert a re-grip operation in your method. The scheduler knows when a re-grip is necessary, and automatically inserts this unit operation when it schedules a method.

SimSockD (simple socket daemon) During a run, SimSockD is a process running on CROSnt that manages communication between the computer and the controller.

slip time The difference between the estimated time to perform a run and the actual time. When a system is suspended, slip time accumulates.

socket See inter-process communication.

stack See stacker magazine.

stacker magazine A container that holds a number of microplates (for example: 20) stacked on top of each other. Also called a stack.

The robot puts the stacker magazine into a stacker/de-stacker which is part of an instrument. The stacker/de-stacker takes the plates one at a time and feeds them to another part of the instrument for the unit operation and then returns them to the stacker magazine.

When a robot gets or puts a stacker magazine of 20 microplates, (rather than doing 20 gets or puts for the 20 microplates), the use of the robot is significantly reduced.

stagger See batch stagger and sample stagger.

step An instance of a unit operation in a method. Each instance has its own properties, such as a step dependency or required lid state.

See also step dependency.

step dependency A property of a particular step specifying that this step cannot be started until another step in the method has been completed.

step duration The preferred or expected length of time for the step to take. For a unit operation where the time is variable, such as an incubation, the step duration is based on the researcher's plan. For a unit operation where the time does not vary and is determined by the performance of the instrument, such as sealings, piercings, and many dispensings, the step duration is based on actual timing of the unit operation.

When creating a schedule, the scheduler uses the step duration.

During a run, when a step takes more time than the step duration plus the buffer time, the system issues an error message.

See also buffer time.

step stagger See batch stagger and sample stagger.

teaching In POLARA, the process of setting robot variables for each instrument in the workspace. You teach a location variable using ash (the application shell) from the CROSnt command prompt.

All robot variables are stored in the instrument server's .v3 file.

See also location (instrument) and v3 file.

template A workspace that has been copied to the template directory. You typically use templates to create subset workspaces of a large base workspace for testing, method development, or production.

throughput The number of samples processed per unit of time.

time stagger See batch stagger and sample stagger.

track robot A robot arm mounted on a linear axis, usually installed in a lab table to permit the arm to move the entire length of the table.

transfer station An instrument that transfers material from one container to another.

unit operation The smallest definable task that can be performed on a sample.

Each instance of a unit operation in a method is called a step. Each step has properties you must set,

such as the time for an incubate operation or the wavelength to use in a fluorometer read operation.

See also property, instrument, method and step duration.

utility A POLARA component running under CROSnt that provides general services.

All POLARA systems have RTM, RAD, SimSockD.

Some POLARA systems also have a PLC server.

See also programmable logic controller (PLC), robot administration server (RAD), SimSockD (simple socket daemon), Run-Time Manager (RTM), and region.

v3 file A file that stores locations and other teachable robot variables. Each instrument has its own .v3 file that stores the robot variable values for that instrument. The .v3 file is stored in the instrument's directory in the CROSnt /app directory.

You use CROSnt application shell to create, and set the values of, the variables in the .v3 file.

See also teaching and location (instrument).

variable file See v3 file.

virtual instrument A POLARA instrument component that has no physical counterpart in the lab system. POLARA permits you to develop schedules that use a virtual instrument, so that you can see how it will affect throughput. However, you cannot perform a run with a schedule that uses a virtual instrument.

See also instrument.

warning message Signals occurrence of an event that requires the operator's attention, but which is not serious enough to suspend the run in either attended or unattended mode. Suppose, for example, that an instrument's reagent liquid level is low. The run continues but the Run beacon's yellow light starts flashing.

workspace A POLARA file that contains the configuration of the lab system, and the profiles, methods, and schedules that have been created for it.

The lab system configuration is set by the POLARA administrator; the profiles, methods, and schedules are usually created by researcher or operator users, though administrators can do these activities as well.

See also profile, method, and schedule.

Index

L

M

O

P

R

S

T

U

W

Y