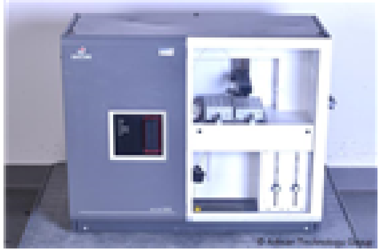


INSTRUMENT HANDBOOK

Biacore®2000

BIACORE

BIACORE 2000
Instrument Handbook

January 2001

BIACORE® 2000 Instrument Handbook, January 2001 (version AD).

Biacore AB reserves the right to make changes to the design and specification of BIACORE® 2000 and to the content of the Instrument Handbook without prior notice.

Note
Previous to October 1st 1996, the company Biacore AB was named Pharmacia Biosensor AB. In connection with the change of company name, the convention of writing the instrument name as BIAcore 2000 was changed to BIACORE 2000, with upper-case letters throughout. These changes have not been implemented throughout the illustration material in the current edition of the Instrument Handbook.

Contents

1. Introduction

2. Description

3. Installation

4. Operation

5. Application Wizards

6. Customized Applications

7. Maintenance

8. Troubleshooting

9. BIACORE Control Software reference

A. System specifications

B. Application guidelines

C. Chemical resistance and recommended solutions

D. Status and event log messages

E. Writing BIACORE 2000 methods

F. Method language reference

Index

1. Introduction

1.1 Biomolecular interaction analysis

BIACORE® 2000 from Biacore AB is a system for real-time biomolecular interaction analysis (BIA) using surface plasmon resonance (SPR) technology.

Real-time BIA monitors the formation and dissociation of biomolecular complexes on a sensor surface as the interaction occurs. By covalently attaching one molecule (the ligand) to the surface, the binding of another molecule in solution (the analyte) with the ligand is followed. Measurements are made under conditions of continuous flow, where the sensor surface forms one wall of the flow cell. For the majority of applications, the biospecific surface can be regenerated and reused for an extended series of analyses.

SPR is a non-invasive optical measuring technique which measures the mass concentration of biomolecules in close proximity to a specially prepared surface. The technique does not require any labelling of the interacting components. The response is largely independent of the nature of the biomolecule, so that all steps in an interaction analysis may be followed with the same detection technique.

Real-time BIA measures surface protein concentrations from a few pg/mm^2, and can monitor changes in the surface concentration on a time scale down to 0.1 s. The technology allows determination of binding specificity, kinetics and concentration with analytes ranging in size from about 150 to 10^6 g/mole (daltons) or more. Flow-through technology with integrated microfluidics gives reproducible sample handling and simplifies kinetic calculations.

Real-time BIA can be used for studying the interactions of proteins, protein conjugates, nucleic acids, lipid micelles and even larger particles such as viruses and whole cells. The technology is valuable in a wide range of application areas, ranging from basic research to drug design, monoclonal antibody production and infectious disease research.

Some of the major uses of real-time BIA are:

- Measurement of kinetic constants, both rate and equilibrium, and kinetic ranking of related molecular species.
- Detection of binding partners in screening applications.
- Analysis of multiple binding patterns, where the ability of several analytes to bind in sequence to a ligand can reveal interference and cooperativity between analytes.
- Epitope mapping studies, where interference between different molecules for binding to the same interactant is used to probe the structure of the interactant (typically used with antibody-antigen and receptor-ligand binding).
- Concentration measurements, using the biospecific surface as a selective probe to measure the concentration of a target substance in complex mixtures.
- Monitoring of enzymatic processes, e.g. proteolysis, phosphorylation, nucleic acid synthesis, ligation and cleavage.

Since the detection technique does not require any labelling of the interacting components, many of these analyses can be carried out without purifying the ligand. For example, kinetic ranking and epitope specificity studies can be performed for monoclonal antibodies in unfractionated hybridoma culture supernatant.

Operation of BIACORE 2000 can be preprogrammed for unattended analysis of up to 192 samples (2 × 96-well microtitre plates).

1.2 Safety information

BIACORE 2000 Processing Unit contains mains voltage and handles liquids which may be hazardous. Before operating or maintaining the system, **you must read the manual and be aware of the hazards described below.** Follow the instructions provided to avoid personal injuries or damage to the equipment. Do not use the equipment in any other way than described in this manual.

WARNING!
For your own safety, always wear appropriate protective clothing during operation and maintenance of BIACORE 2000.

WARNING!
Use required safety equipment when handling hazardous substances.

WARNING!
If the instrument is contaminated with biohazards, please contact your local Biacore User Support for further information about decontamination procedures.

WARNING!
BIACORE 2000 Processing Unit contains mains voltage of up to 265 Vac.

Do not open covers. Disconnect mains cord before replacing fuses.

1.3 Conformance with standards

BIACORE 2000 Processing Unit meets the following safety standards and directives:

- IEC 1010-1 (EN 61010-1), Safety requirements for electrical equipment for measurement, control and laboratory use.

- 89/336/EEC, European directive concerning electromagnetic compatibility.

- 73/23/EEC, European directive concerning low voltage devices.

2. Description

BIACORE 2000 consists of a processing unit with liquid handling and optical systems and a PC running BIACORE Control Software. Sensor chips and other consumables are available from Biacore AB.

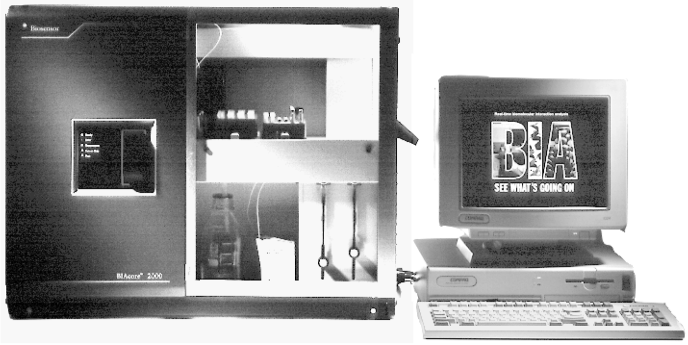


Figure 2-1. *BIACORE 2000 system processing unit and controlling computer.*

2.1 BIACORE 2000 processing unit

BIACORE 2000 processing unit houses the following components:

- two liquid delivery pumps, one for maintaining a constant flow of liquid over the sensor chip surface and the other for handling samples in the autosampler
- the autosampler, for preprogrammed sample handling including mixing and injection to the detector unit as well as sample recovery and fraction collection
- the Integrated µ-Fluidic Cartridge (IFC), containing liquid delivery channels, sample loop and valves
- the detector unit, including optical and electronic components for creating and measuring SPR response
- 4 detector flow cells formed by the IFC pressing against the sensor chip
- microprocessors for controlling pumps, autosampler and IFC valves, and for basic processing of the SPR signal.

The removable sensor chip is inserted into the cassette port on the detector unit, and docks into the instrument to form one side of the detector flow cell. The IFC is connected to the buffer supply through the connector block (sometimes referred to as the *recovery block* in

BIACORE 2000), which has an injection port for loading samples into the IFC.

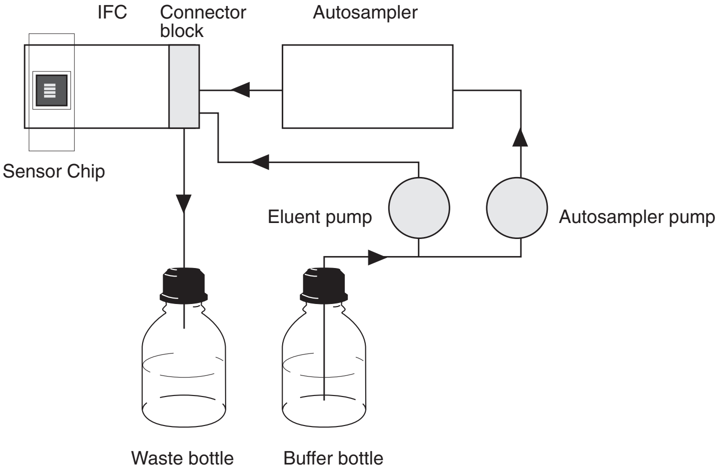


Figure 2-2. *The liquid delivery system with two pumps and an autosampler transports liquid to the connector block where samples and buffers are injected to the IFC.*

The door to the autosampler and buffer compartment is locked during instrument operation, to protect the operator from injury by the moving needle of the autosampler. The instrument cannot be operated with the door open.

2.1.1 Pumps

Two identical pumps are used for liquid handling in BIACORE 2000. The left-hand pump maintains a continuous flow of liquid through the detector flow cell, while the right-hand pump is used for transferring, diluting and mixing samples in the autosampler and for injecting samples into the IFC. The system switches automatically between these pumps as required. In this way, continuous flow through the flow cell is not interrupted by sample handling operations. Pump tubings are marked A for the right-hand (autosampler) pump and BB for the left-hand (buffer) pump.

The pumps are syringe pumps, specially designed for accurate pulse-free flow at rates down to 1 µl/min. Each pump has a stroke capacity of 500 µl.

The continuous flow pump takes liquid from a buffer vessel in the bottle compartment and pumps it directly into the IFC inlet through the connector block. The autosampler pump is also primed with buffer from the buffer vessel, but is normally used for aspirating and dispensing samples in the autosampler. The autosampler pump can also be programmed to dispense buffer from the buffer vessel into autosampler vials. Samples are prevented from mixing with priming buffer by small air segments maintained in the pump tubing.

The buffer compartment can hold two 500 ml containers, one for buffer and one for collecting waste effluent from the flow cells.

2.1.2 Autosampler

The autosampler is used for transferring, diluting and mixing samples, injecting samples into the IFC, and for sample recovery and fraction collection. It is situated directly above the buffer compartment.

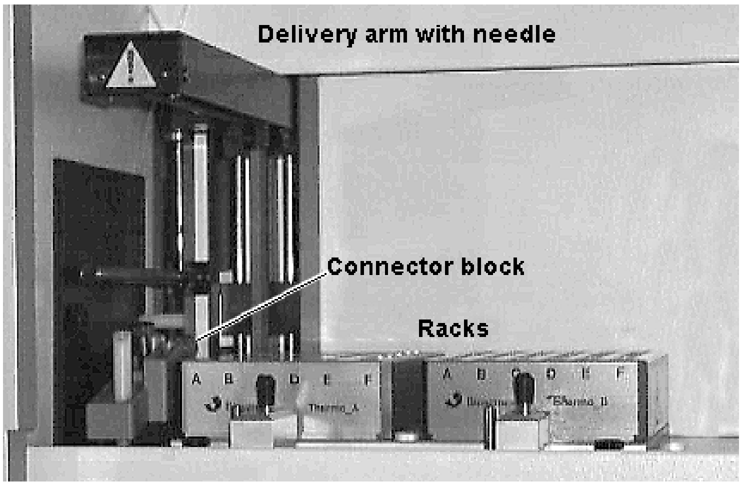


Figure 2-3. *The autosampler compartment.*

The autosampler rack base can accommodate two microtitre plates or aluminium thermo-racks. Samples are placed directly in microtitre plate wells or in vials in the thermo-racks. The stainless steel autosampler needle is mounted on a positioning arm driven by three stepper motors, which carry the needle to sample positions or to the connector block for injection of sample into the IFC. The needle is directly connected to the autosampler pump. Operation of the pump is coordinated with needle movements.

Autosampler operation is controlled from BIACORE Control Software (see Chapters 4 and 9) through simple commands like TRANSFER, MIX and INJECT.

The temperature of the autosampler rack base can be controlled by connecting to an external thermostat pump. Aluminium thermo-racks provide the most efficient heat transfer to or from samples. The use of thermo-rack vials with septa is recommended to prevent evaporation or condensation.

2.1.3 Connector block

The connector block has two inlet ports, a pump port for buffer flow with a screw-nipple fitting for the pump tubing and an injection port for the autosampler needle to inject sample and reagents. These inlets connect directly with flow channels in the IFC. There are outlet ports for waste and for recovered sample.

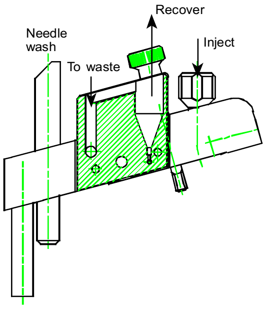


Figure 2-4. *Side view of the connector block, showing the injection port, wash station and recovery cup.*

2.1.4 Integrated µ-Fluidic Cartridge (IFC)

The IFC consists of a series of channels and pneumatic valves encased in a plastic housing, and serves to control delivery of liquid to the sensor chip surface. Samples are injected from the autosampler into the IFC, which connects directly with the detector flow cells.

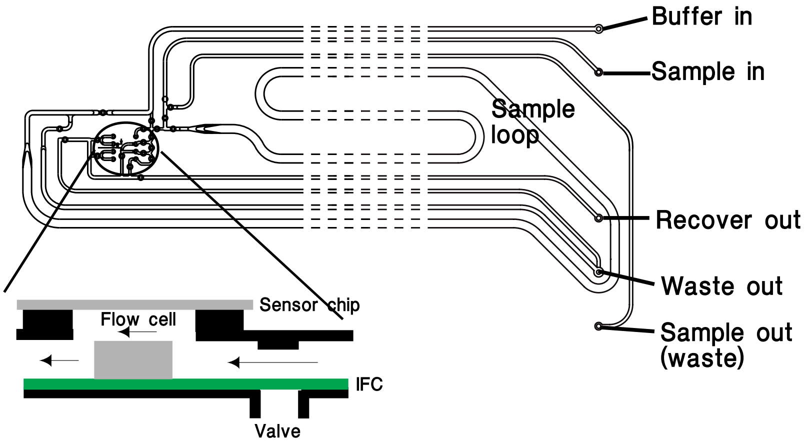


Figure 2-5. *Diagram of the channels and valves in the IFC as seen from above. The inset shows a side view of how the flow cells are formed by pressing the sensor chip against the IFC. Not to scale.*

There are two operation modes for sample injection with the IFC: 1) direct injection from the autosampler needle and tubing, and 2) loop injection using the sample loop in the IFC. These are illustrated schematically in Figure 2-6. The normal mode is direct injection, which can handle sample volumes up to 325 µl in a single injection. The sample loop has a capacity of 100 µl, and is loaded in the reverse direction from sample injection to minimize sample waste with smaller volumes. Loop injection is used in special circumstances, including

- Injection of volumes larger than 325 µl, which are handled by a sequence of direct injection—loop injection—direct injection giving a maximum sample volume of 750 µl (325 µl direct + 100 µl loop + 325 µl direct).
- Injection of one sample immediately following another (co-inject) which is used e.g. for monitoring dissociation in the presence of excess ligand to prevent rebinding of analyte to the surface. For co-inject, the first sample is injected in loop mode and the second in direct mode.
- Manually controlled injections, which are always performed in loop mode.

For kinetic measurements it is important that the sample is delivered to the flow cell in a well-defined plug with minimum dispersion at the sample-buffer boundaries. The sample plug is created by switching between sample and buffer flow in the IFC with the help of pneumatic valves. Dispersion is minimized by the construction of the IFC, where dead volumes between valves and the flow cell are kept as small as possible. To prevent undue diffusion between sample and buffer at the beginning and end of the sample plug in the autosampler tubing, sample aspiration automatically places a number of air segments between the buffer and the sample (see Figure 2-6). The air segments at the beginning of the sample are flushed to waste when the injection is primed, and injection is clipped before the air segments at the end enter the flow cell. The number of segments varies with different types of injection (e.g. one for the fastest injection, three for the most accurate).

The detector flow cells are formed directly on the IFC by docking the sensor chip on to the IFC flow cell block. This consists of four grooves, rectangular in cross section, moulded into the surface of the IFC. Each groove has its own liquid delivery channel and control valve in the IFC. The block is pressed against the sensor chip, forming four separate flow cells with sensor chip surface as one wall (Figure 2-7). Effluent from the flow cells is collected in an outlet channel and directed either to waste or the recovery cup.

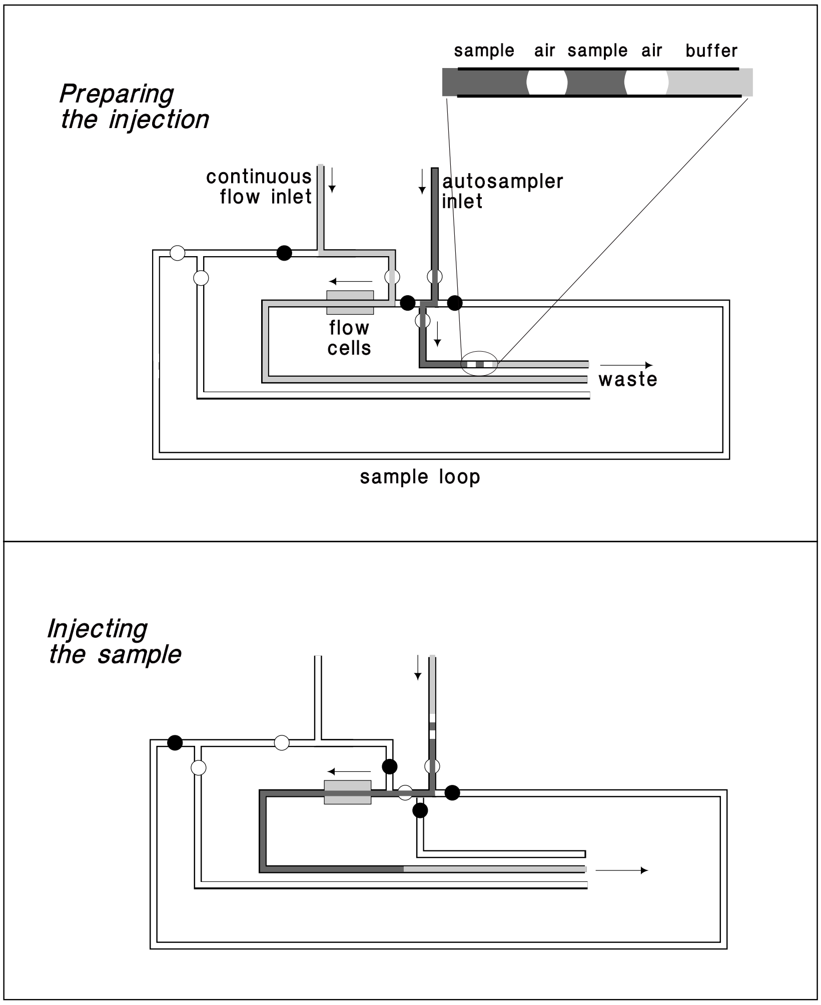


Figure 2-6a. *Schematic illustration showing the flow paths during direct injection. The symbols ○ and ● represent open and closed valves respectively. Not to scale.*

During preparation, the leading sample is flushed to waste to ensure minimum dispersion at the front of the injected sample plug. The continuous flow pump maintains a constant flow of buffer through the flow cell. The enlargement shows two air segments separating the sample from buffer which prevent mixing in the autosampler tubing and IFC channels.

During sample injection, the sample is driven directly through the flow cell by the autosampler pump.

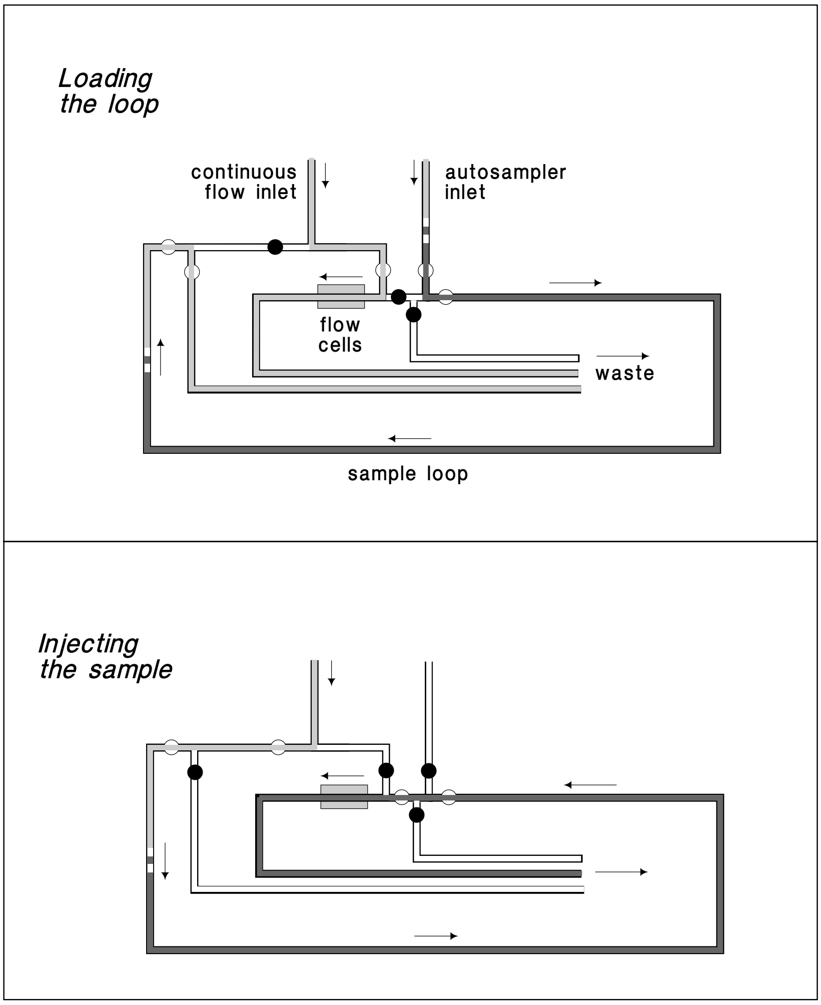


Figure 2-6b. *Schematic illustration showing the flow paths during loop injection. The symbols ○ and ● represent open and closed valves respectively. Not to scale.*

During loop loading, the continuous flow pump maintains a constant flow of buffer through the flow cell.

During sample injection, buffer from the continuous flow pump pushes the sample out of the sample loop over the sensor surface in the opposite direction from loading.

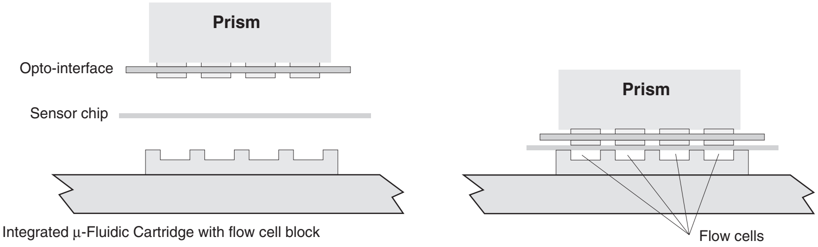


Figure 2-7. *Precision-cast channels in the surface of the IFC define four flow cells on the sensor chip surface. These are matched by ridges of transparent polymer on the opto-interface, ensuring good optical contact between the components of the system. Not to scale.*

The IFC allows single or multichannel analysis in up to four flow cells. Flow cells 1 and 3 can be used as on-line reference cells, allowing blank-subtracted data to be presented directly on the screen during analysis.

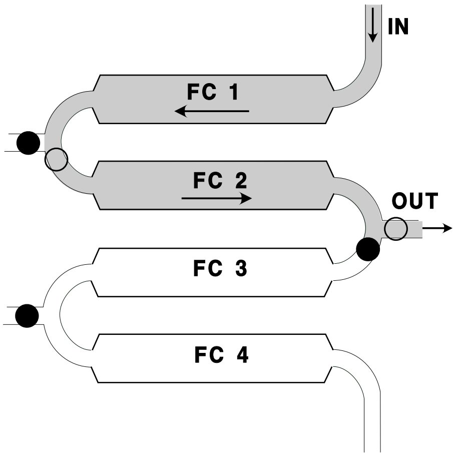


Figure 2-8. *Simplified diagrammatic illustration of sample flow through two flow cells. The symbols ○ and ● represent open and closed valves respectively. Valves and channels not used in other flow path arrangements are not included. Not to scale.*

Sample injection commands activate a sequence of operations covering aspiration of sample from the autosampler vial, delivering the required amount of sample to the flow cell(s), and flushing the IFC with buffer ready for the next injection.

2.1.5 Sensor chips

Sensor Chip CM5 is a general purpose chip coated with carboxymethylated dextran to which biomolecules can be linked with a variety of established chemical methods (see BIAtechnology Handbook for a description of immobilization chemistry). Other sensor chips are also available for specific application areas.

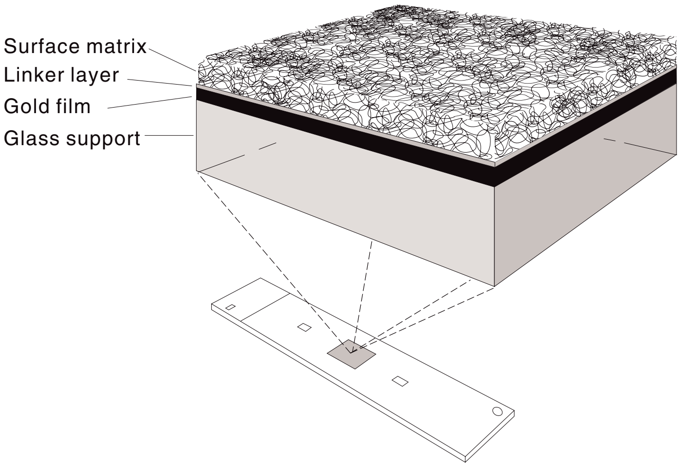


Figure 2-9. *Sensor Chip CM5 is a glass slide coated with a thin gold film to which the carboxymethyl dextran surface matrix is covalently bonded. The matrix surface forms one wall of the detector flow cell.*

The sensor chip is mounted on a plastic support frame in a protective cassette. The cassette is inserted into the sensor chip port on the front of the processing unit, where the conveyor mechanism slides the support frame and sensor chip out of the cassette. Software-controlled docking presses the sensor chip in place between the glass prism and opto-interface unit on one side and the IFC flow cell block on the other (see Figure 2-7). A green LED marked "Sensor Chip" on the front panel of the instrument indicates the status of the sensor chip (no light=no sensor chip, flashing light=not docked, steady light=docked).

One sensor chip can be used for repeated measurements (50-100 runs depending on the properties of the immobilized ligand and the regeneration conditions used), and can remain docked in the processing unit for up to one week. A chip can also be removed from the instrument and re-docked at a later stage (the shelf life of a used chip will depend on the nature of the immobilized ligand).

2.1.6 Optical system

The glass side of the sensor chip is pressed into contact with a glass prism in the optical unit. A silicone opto-interface between the sensor chip and the prism, matched in refractive index to the glass, ensures good optical coupling between the prism and the sensor chip.

Light from a near-infrared light-emitting diode (LED) is focused through the prism on to the sensor chip surface in a wedge-shaped beam, giving a fixed range of incident light angles. Light reflected from the sensor chip is monitored by a linear array of light-sensitive diodes covering the range of incident light angles to a resolution of approximately 10^{-5}°.

By using a wedge of incident light and a fixed array of detectors, the SPR angle is monitored accurately in real time, with no physical movement of light source, sensor chip or detector. The SPR angle is determined at a frequency of 160 Hz (measurements/s), and then averaged by the microprocessor in BIACORE 2000 and transmitted to BIACORE Control Software at rates up to 10 Hz.

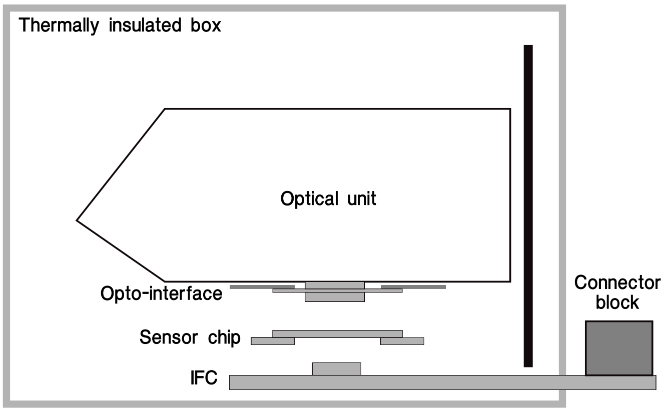


Figure 2-10. *The detection unit houses the optical unit with light source and detector, the opto-interface and the removable sensor chip. Flow cells on the sensor chip are connected to the liquid handling system through the Integrated µ-Fluidic Cartridge. The broken line shows the boundaries of the thermally insulated box.*

2.1.7 Temperature control

SPR measurements are sensitive to changes in temperature. It is important that a constant temperature is maintained at the sensor chip surface throughout the run.

The temperature control uses Peltier elements to maintain a constant preset temperature at the sensor chip surface. If the temperature is not stable, the yellow LED on the processing unit flashes and the temperature indication in BIACORE Control Software shows alternately the preset and actual temperature. The temperature control can maintain a preset temperature in the range 4-40°C as long as the operating temperature is not more than 20°C below the ambient temperature.

The temperature at the sensor chip surface is recorded automatically by BIACORE Control Software at the beginning and end of every run.

2.1.8 LED status indicators

There are five LED status indicators on the front panel of the instrument.

Ready (green)
Lights when power is on and the instrument is ready for use. A flashing light indicates that communication with the PC is broken.

Error (red)
Lights to indicate a serious error in the instrument.

This LED is lit for about 30 seconds after the power is switched on. If the LED lights in other circumstances, turn off the instrument and call your Biacore service representative.

Temperature (yellow)
Lights steadily when the temperature at the flow cell is stable at the preset temperature. Flashes when the temperature is not stable.

Sensor chip (green)
Lights when a sensor chip is docked and ready. Unlit when there is no sensor chip, and flashing if a sensor chip is inserted but not docked.

Run (green)
Lights when a method-controlled or manual sensorgram is running.

2.2 BIACORE 2000 system control

Operation of the processing unit is controlled by an internal microcomputer, which is also responsible for recording and processing raw data from the detection unit. This internal microcomputer is in turn controlled from a separate personal computer, the system controller.

All BIACORE 2000 functions, except for mechanical operations such as inserting the sensor chip cassette in the cassette port, are controlled from BIACORE Control Software running under Microsoft Windows 95 or Windows/NT on an IBM-compatible PC. This software provides an integrated interface for BIACORE 2000 operation and data collection. Information is passed between the system controller and the microcomputer in the processing unit via a standard serial communication line.

During a run, data from the detection unit are collected and processed by the internal microcomputer in the processing unit. These measurements are temporarily stored in the microcomputer's primary memory, and averaged values are passed to the system controller. In the system controller, the measurements are displayed on the screen as a sensorgram and stored permanently on the computer hard disk.

Because collection of raw data is managed by a separate microcomputer in BIACORE 2000, the time resolution of measurements is not affected by computing activity on the PC running BIACORE Control Software. Simultaneous use of the PC for BIACORE control and for other programs is however not recommended.

2.3 BIACORE Control Software

BIACORE Control Software runs under Microsoft Windows 95 or Windows/NT version 4 on an IBM-compatible PC. The software includes the following functions:

- toolbar- and menu-driven interface for interactive manual control
- application wizards for guidance in standard analysis procedures
- method editor for pre-programmed operation
- direct real-time display of sensorgrams
- continuous display of sensorgram coordinates, elapsed time and response, flow rate and path, measuring temperature, and detection mode
- automatic event markers for e.g. start and end of sample injection, pump refill and extra wash

- reference line for display of elapsed time and absolute response at any point on the sensorgram
- event log which records all events in the current run
- report point table which records pre-defined sensorgram information at any requested point on the sensorgram
- notebook for experimental details
- user tools for standard operations and maintenance.

2.4 BIAevaluation software

BIACORE Control Software includes an interface to the separate BIAevaluation Software for help in interpreting sensorgram data. Routines are currently supplied for:

- comparing sensorgrams from separate runs by generating overlay plots
- deriving binding rates from sensorgram data, primarily for use in concentration measurements (from analyses under conditions where the binding rate is directly proportional to the analyte concentration)
- deriving kinetic rate constants from association and dissociation phases of sensorgrams
- evaluating concentration measurements.

The operation of BIAevaluation is described in a separate handbook.

2.5 BIAsimulation software

BIACORE Control Software includes an interface to simulation routines which can generate expected sensorgrams for the most common experimental situations:

- kinetic analysis of association and dissociation phases, with optional simulation of signal noise, baseline drift and mass transport limitations
- kinetic analyses with heterogeneous ligand or analyte
- multiple binding studies
- epitope mapping matrices
- concentration determination.

BIAsimulation provides valuable assistance in understanding sensorgrams and can help in interpreting experimental results.

The operation of BIAsimulation is described in a separate handbook.

3. Installation

BIACORE 2000 will be installed by a Biacore representative.

3.1 Instrument placing

Place the instrument on a firm bench clear of strong electrical or magnetic fields. Do not place vibrating equipment (e.g. bench centrifuges) on the same bench. If possible avoid placing the instrument close to a window where direct sunlight or ventilation may cause sudden temperature changes.

For correct functioning of the temperature control system, make sure that the fan inlet (underneath the instrument at the front left-hand side) and outlet (on the back panel at the left-hand side) are not obstructed. There should be at least 20 cm clearance behind and above the instrument to allow adequate air circulation.

3.2 Moving BIACORE 2000

3.2.1 Within the laboratory

If BIACORE 2000 is to be moved locally within the laboratory, undock the sensor chip and shut down the system. See Sections 4.4.15 and 4.7 for a full description of these operations.

Lift the instrument by the handles (one at each side). Keep the instrument upright during moving.

Caution
The instrument is heavy (50 kg) and may need two people to lift it.

3.2.2 Longer distances

Contact your Biacore representative for instructions.

3.3 Connections

1. The **MAINS INLET**, the **MAINS POWER** switch and the mains fuses are located on the mains input panel on the left side of the instrument. Input voltage range, maximum power consumption and fuse ratings are printed on the mains input panel.
   Connect the power cord to the mains inlet.

2. Install computer peripheral equipment according to the respective instruction manuals. Use a parallel port for printer connection (normally LPT1) and serial port COM1 for BIACORE 3000 (right-hand side of the instrument). Serial port COM2 is for service purposes.

Note: On Compaq personal computers, the COM1 serial port is marked “IOIOIA”. On other makes of computer, the port may be marked “COM1”, “Serial Port 1” or similar, or not marked at all. Refer to your computer documentation if you are uncertain about the location of the serial port.

3. Check that the voltage selector on the PC is set correctly.

3.4 Software installation

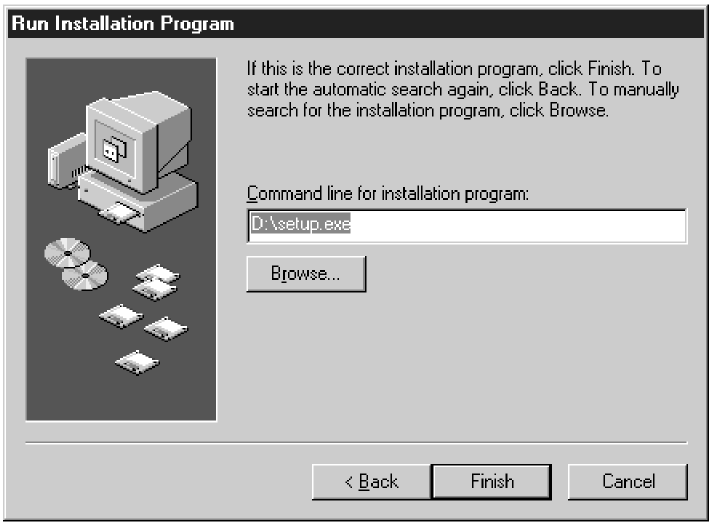
BIACORE Software is supplied on CD-ROM. The software will be pre-installed on delivery. If the software needs to be re-installed for any reason, follow the steps below:

1. Turn on the PC and start Microsoft Windows 95.

2. If you have a memory-resident virus protection program installed, you may need to disable this program before installing BIACORE Software.

3. Insert the BIACORE Software CD-ROM in the CD drive.

4. If the setup program does not start automatically, choose **Add/Remove Programs** from the Windows 95 Control Panel and click on the **Install** button. The installation program is **setup.exe** on the CD-ROM disk.

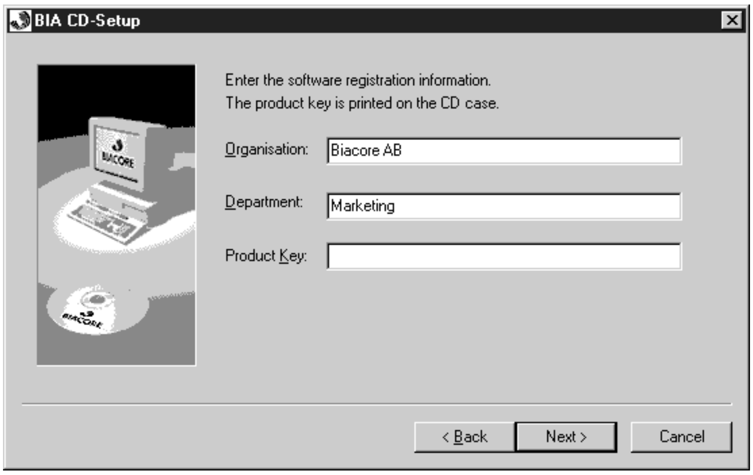


Run Installation Program

If this is the correct installation program, click Finish. To start the automatic search again, click Back. To manually search for the installation program, click Browse.

Command line for installation program: D:\setup.exe

5. Read the instructions on the screen, then click **Next>**. Enter the user registration information. The product key number is printed on the CD box.



BIA CD-Setup

Enter the software registration information.
The product key is printed on the CD case.

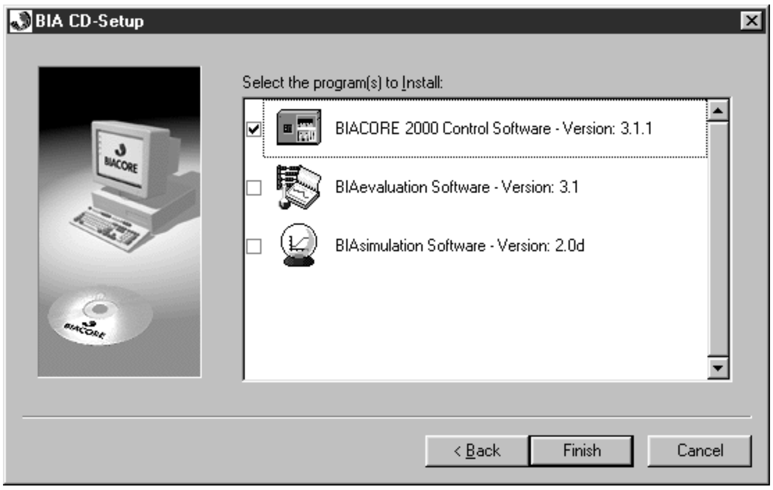
Organisation: Biacore AB

Department: Marketing

Product Key:

6. The product key is valid for the software programs you have purchased. Software cannot be installed without a valid product key. Check the boxes for BIACORE Control Software, BIAevaluation Software and BIAsimulation and click on **Install**. The software will be installed in default folders under **C:\Program Files**.

Note: If you want to install the software in different folders, click on the appropriate software item with the right mouse button and choose **Customized installation**. This will allow you to specify the destination folder. Installation in the default folders is however recommended.



BIA CD-Setup

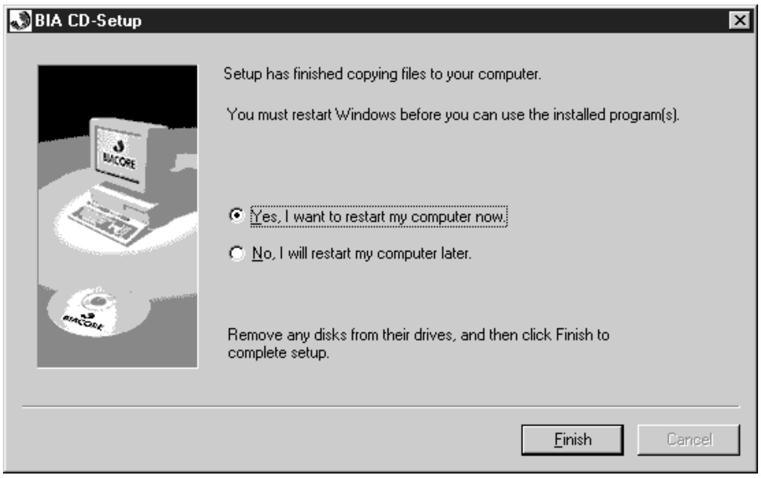
Select the program(s) to Install:

☑ BIACORE 2000 Control Software - Version: 3.1.1

□ BIAevaluation Software - Version: 3.1

□ BIAsimulation Software - Version: 2.0d

7. Click on **Finish** when the installation is complete.



BIA CD-Setup

Setup has finished copying files to your computer.

You must restart Windows before you can use the installed program(s).

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

8. Re-enable your virus protection program if you turned it off in step 2. If you are updating BIACORE Software, some virus protection programs may detect that executable files have been altered and warn you that you may have a virus. Ignore this warning: the files have been “altered” because they are updated, not because they are infected. Do not however ignore virus warnings that appear at times when the software has not just been updated.

3.5 Power up

1. Switch on BIACORE 2000. The LEDs on the front panel should light in the following sequence:

   a) All the LEDs at the front light for about 10 seconds and then go out.

   b) The green Ready LED lights.

   c) The yellow Temperature LED flashes to indicate non-stabilized temperature, then is steadily lit when the temperature at the detection unit is stable. The time required for temperature stabilization will depend on instrument and ambient temperatures.

2. Start BIACORE Control Software. The instrument software is downloaded to BIACORE 2000, which takes several minutes. (This downloading procedure is performed only if the instrument has been switched off.)

3.6 Liquid system start-up

1. Put buffer and waste reservoirs in place. Insert the inlet tubing into the buffer reservoir. HBS buffer is recommended as a general-purpose running buffer, and is supplied by Biacore AB in disposable ready-to-use packs.

   A 500-1000 ml flask or bottle is recommended as a waste reservoir.

2. Insert a new sensor chip. Close the instrument door and choose **Command:Dock** in the BIACORE Control Software menu. Run **Working Tools:Prime** after docking.

   Note: The instrument door will be locked during operation. This is a safety measure to prevent injury by the sharp autosampler needle.

3. Run **Working Tools:Rinse** to flush the liquid handling system a second time. Check that there are no leaks at tubing connectors.

4. Run **Test Tools:System check** to check the system performance. See Section 7.2.2 for details of this procedure.

4. Operation

This chapter describes the operations involved in using BIACORE 2000. The chapter is organized into the following sections:

- Using the software
- Preparing the system
- Using application wizards
- Manual operation
- Queuing manual commands
- Method-controlled operation
- Viewing and evaluating results

4.1 Using the software

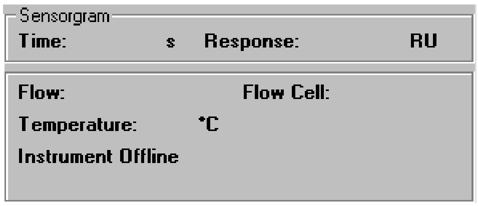
4.1.1 General Windows techniques

BIACORE Control Software runs under Microsoft Windows 95 and follows the general Windows standards for handling screen windows, menus and dialogue boxes.

4.1.2 Starting BIACORE Control Software

A shortcut to the BIACORE Control Software is installed under **BIA Programs** on the **Start** menu.

It does not matter whether you switch on the instrument or start the software first. If you start the software first, the status window will display the message **Instrument Offline**. Connection will be established when the instrument is switched on.



Sensorgram

Time: s Response: RU

Flow: Flow Cell:

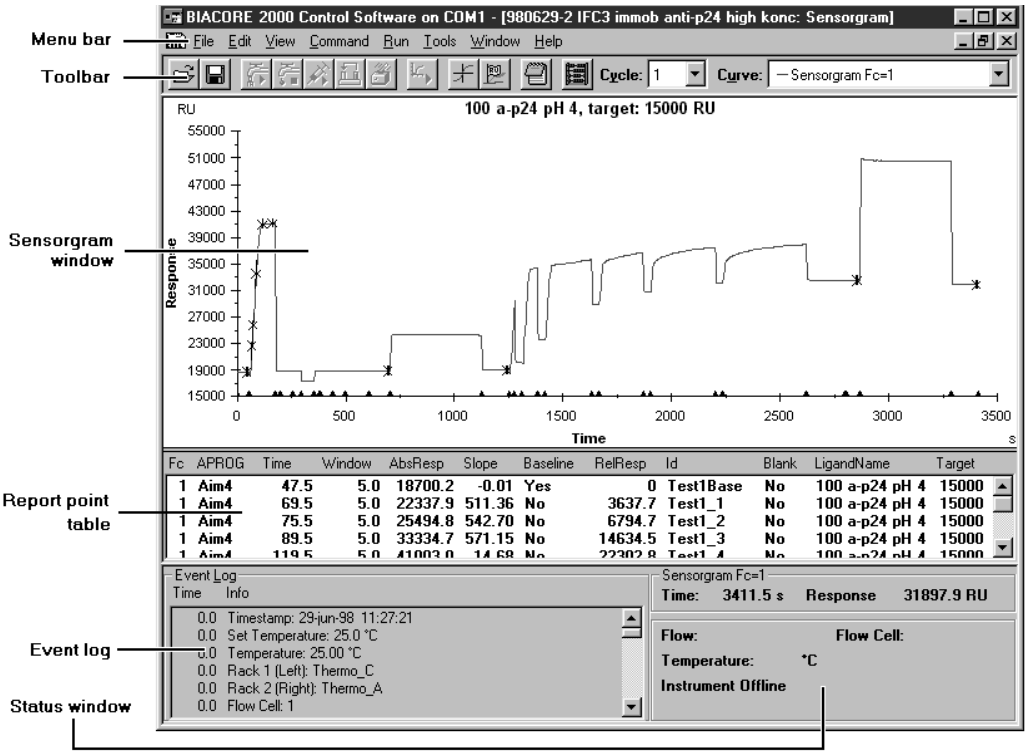
Temperature: °C

Instrument Offline

Depending on the current instrument status when the software is started, you will be prompted for the appropriate set-up action (e.g. dock a sensor chip, prime the system). See Section 4.2 for further details of these actions.

4.1.3 BIACORE 2000 software screen regions

The BIACORE 2000 screen is divided into six main areas:



Fc	APROG	Time	Window	AbsResp	Slope	Baseline	RelResp	Id	Blank	LigandName	Target
1	Aim4	47.5	5.0	18700.2	-0.01	Yes	0	Test1Base	No	100 a-p24 pH 4	15000
1	Aim4	69.5	5.0	22337.9	511.36	No	3637.7	Test1_1	No	100 a-p24 pH 4	15000
1	Aim4	75.5	5.0	25494.8	542.70	No	6794.7	Test1_2	No	100 a-p24 pH 4	15000
1	Aim4	89.5	5.0	33334.7	571.15	No	14634.5	Test1_3	No	100 a-p24 pH 4	15000
1	Aim4	119.5	5.0	41003.0	14.68	No	22302.8	Test1_4	No	100 a-p24 pH 4	15000

Event Log

```
Time   Info
 0.0   Timestamp: 29-jun-98  11:27:21
 0.0   Set Temperature: 25.0 °C
 0.0   Temperature: 25.00 °C
 0.0   Rack 1 (Left): Thermo_C
 0.0   Rack 2 (Right): Thermo_A
 0.0   Flow Cell: 1
```

Sensorgram Fc=1

Time: 3411.5 s Response 31897.9 RU

Flow: Flow Cell:

Temperature: °C

Instrument Offline

- The **menu bar** provides access to BIACORE 2000 menu commands. Many of the commands can also be accessed through shortcut keys (see Section 4.1.4 or 9.3).

- Buttons on the **toolbar** duplicate frequently used menu commands for more convenient access.

- The **sensorgram window** displays one to four sensorgrams, dependent on the detection mode selected. This area also displays the **method editor window**.

- The **report point table** records sensorgram information at each report point.

- The **event log window** records important parameters at the start and end of each run and at events during the run.

- The **status window** displays the current instrument status, including current response, flow rate and temperature. System status messages are displayed at the bottom of this window.

See Chapter 9, BIACORE Control Software for a more detailed description of the screen regions.

4.1.4 Commands and toolbar buttons

In the descriptions in this chapter, commands are referred to by name, with the menu name separated from the command name by a colon (e.g. **Command:Dock**). Many of the commonly used commands are associated with toolbar buttons. Many commands also have shortcut keys. A list of toolbar buttons and shortcut keys is given on the next page. Commands not included in this list are available only from the menu bar. Toolbar buttons are also shown in the margin where appropriate in this chapter.

Menu/Command	Toolbar button	Shortcut key
File		
Open	📂	Ctrl-O
Save	💾	Ctrl-S
Print		Ctrl-P
Edit (sensorgram)		
Copy Graph		Ctrl-H
Add Report Point	🏳	
Edit (method)		
Undo		Ctrl-Z
Cut		Ctrl-X
Copy		Ctrl-C
Paste		Ctrl-V
Find		Ctrl-F
Replace		Ctrl-R
View		
Unzoom		Ctrl-U
Reference Line	┼	F8
Baseline		F9
Notebook	🗒	
Wizard Results		Ctrl-R
Command		
Inject/Stop Inject	▶ ■	
Transfer	⤒	
Mix	🧪	
Flow/Stop Flow	▶ ■	
Run		
Run/Stop Sensorgram	▶ ■	
Stop Sensorgram		
Tools		
Evaluation	▦	Ctrl-E

4.2 Preparing the system

Do not put papers under the instrument, particularly at the left-hand side. Paper is easily sucked against the fan inlet, blocking the air intake. Make sure that the space between the bottom of the instrument and the bench is not obstructed by laboratory equipment, notebooks etc.

4.2.1 Preparing solutions

HBS buffers are ready-to-use HEPES-buffered saline solutions supplied by Biacore AB which is recommended as running buffer for most general BIA applications. They are sterile-filtered, degassed and supplied in disposable packs (200 ml) which can be used as a buffer reservoir. HBS buffers are available with and without EDTA and surfactant.

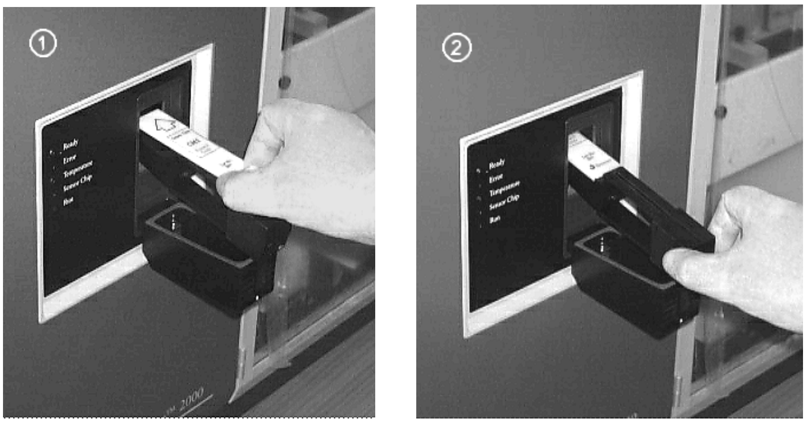
Other solutions for use in BIACORE 2000 should be degassed and filtered through a 0.22 µm filter to remove particles. Buffer solutions should contain 0.005% Surfactant P20[1] to minimize non-specific adsorption of proteins to the autosampler tube and the IFC channels. You may omit surfactant if your sample is detergent-sensitive: however, you should then take special care to clean the IFC channels regularly (see Chapter 7, Maintenance).

4.2.2 Docking the sensor chip

Before BIACORE 2000 can be used, a sensor chip must be docked in the instrument and the flow system primed with buffer.

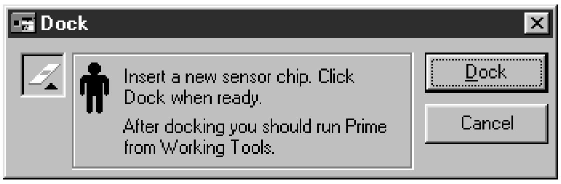
To dock the sensor chip, open the sensor chip port cover, pull out the slide and insert a sensor chip cassette in the docking station with the label facing upwards and the arrow pointing into the instrument. ***Do not force the cassette into the station. Do not slide the sensor chip out of the protective cassette.***

[1] Surfactant P20 is supplied as a a 10% stock solution. To prepare buffer containing 0.005% surfactant, add 50 µl stock solution to 100 ml buffer.



1. Place the sensor chip cassette on the slide with the label facing upwards and the arrow pointing into the instrument. 2. Push the slide into the instrument.

Push in the slide and close the sensor chip port cover, then choose **Command:Dock** in the BIACORE Control Software to dock the chip.



Insert a new sensor chip. Click Dock when ready.

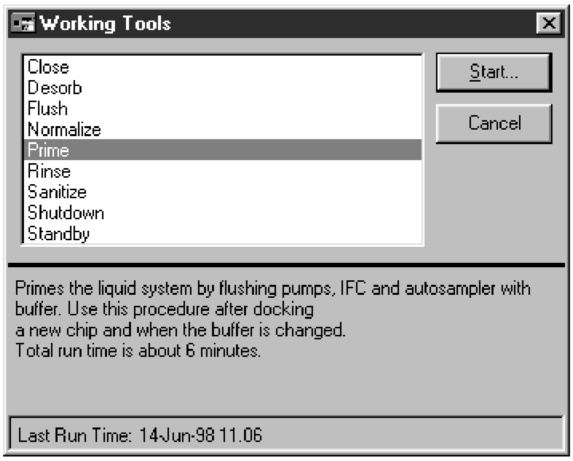
After docking you should run Prime from Working Tools.

4.2.3 Initiating the liquid system

After docking a new sensor chip, run the working tool **Prime** to prime the flow system with buffer. The working tools dialogue box is automatically opened after a docking operation.

Place running buffer and a waste bottle in the buffer compartment beside the pumps. Place the buffer to the right and the waste bottle to the left. Insert the pump tubings (make sure these are clean!) from both pumps into the buffer carton and the waste outlet tubing into the waste bottle.

In the working tools dialogue box, make sure **Prime** is selected and click on **Start**. Follow the instructions on the screen.



Primes the liquid system by flushing pumps, IFC and autosampler with buffer. Use this procedure after docking
a new chip and when the buffer is changed.
Total run time is about 6 minutes.

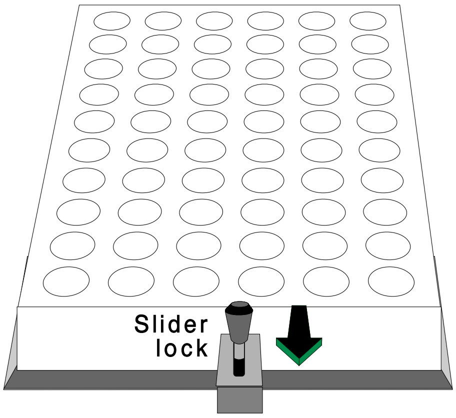
Last Run Time: 14-Jun-98 11.06

If you do not want to prime the system immediately after docking, click on **Cancel** in the Working Tools dialogue box. Note however that you should run **Prime** before using the sensor chip for ligand immobilization or interaction analysis.

4.2.4 Setting autosampler rack types

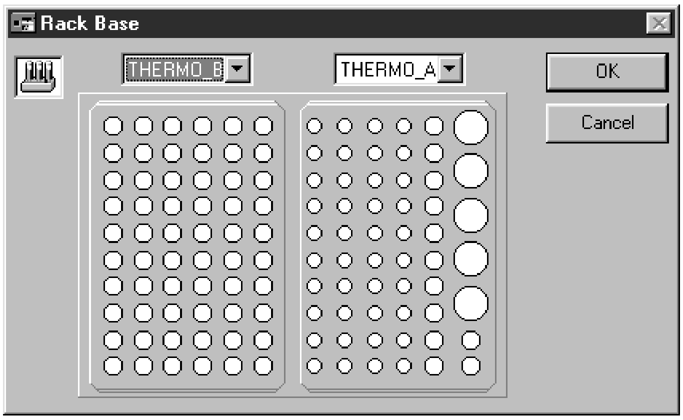
The autosampler can hold two sample racks, which may be either microtitre plates or one of three thermo-racks (see Section A.3).

To place racks on the autosampler rack base, pull back the slider lock at the front of the base and place the rack on the base with the text at the front. The rack will be locked in place when you release the slider. Make sure that the rack is properly seated on the base.



Pull the slider lock towards you to insert or remove autosampler racks.

Choose **Command:Define Rack Base** to tell the control software which rack types are being used. This setting determines which rack positions are valid for commands like **Transfer**, **Mix**, **Dilute** and **Inject**.



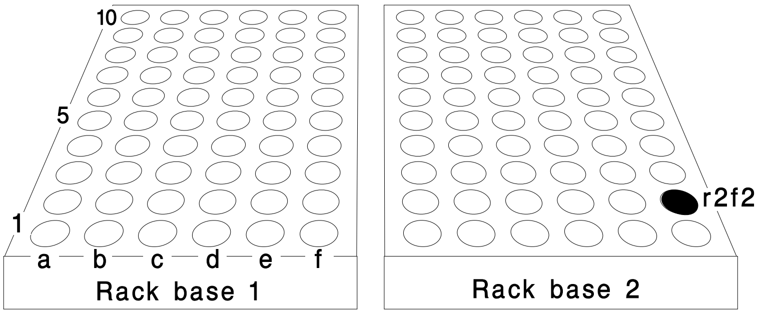
Select appropriate rack types from the pull-down lists and click on **OK**.

Note: There is no automatic sensing of rack type in the software. If you set the wrong rack type, the autosampler needle may be directed to the wrong or non-existent positions. Remember to re-define the rack base using this command if you change the type of rack in the autosampler.

4.2.5 Normalizing the signal response

Variations in reflectance characteristics between different sensor chips affect the SPR signal. These are compensated by a normalization procedure, using a standard glycerol solution to create a total reflection. You should run the normalization procedure when you change the detection temperature (see Section 4.2.6) or when you dock a new sensor chip and require maximum sensitivity.

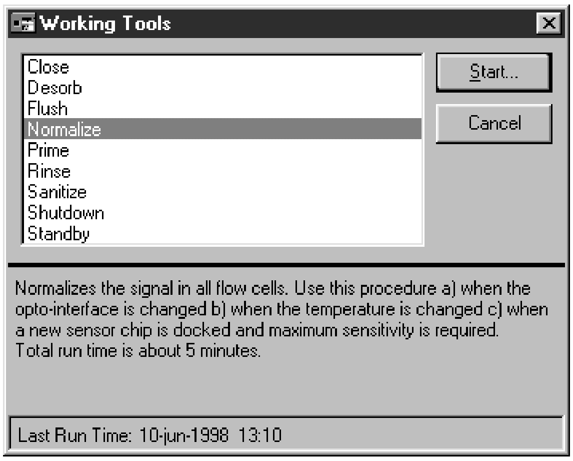
BIAnormalizing solution (40% w/w glycerol in water) is included in the BIAmaintenance Kit. Transfer 0.5 ml to a 9 mm vial and place the vial in position R2F2, i.e. rack position 2 (right-hand rack), sample position F2 (second position from the front in row F). You should have rack type A or B in rack position 2 for this procedure. The procedure will prompt for rack base definition if any other rack type is currently defined in position 2.



The autosampler compartment viewed from the front, showing the position R2F2 used for BIAnormalizing solution. Thermo-rack type B is shown in both base positions.

Note: The glycerol concentration (40% w/w) is critical for correct normalization.

Close the door to the autosampler compartment. Select **Tools: Working Tools.** Select **Normalize** in the list and click on **Start.** Follow the instructions on the screen.

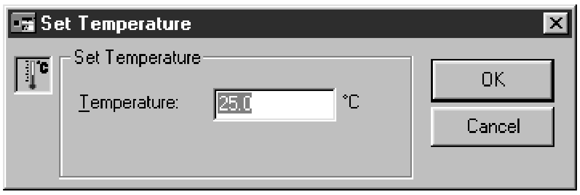


Normalizes the signal in all flow cells. Use this procedure a) when the opto-interface is changed b) when the temperature is changed c) when a new sensor chip is docked and maximum sensitivity is required.
Total run time is about 5 minutes.

Last Run Time: 10-jun-1998 13:10

4.2.6 Setting the temperature

Select **Command:Set Temperature** and enter the required analysis temperature.



The temperature display in the status window toggles between the set temperature and the actual temperature and the temperature LED on the processing unit flashes until the measured temperature is stable.

Wait until the sensor chip surface temperature is stable before starting a sensorgram. A temperature change of 5°C takes about 60 minutes.

Note: The **Set Temperature** command controls the detection temperature at the sensor chip surface. Samples and buffer equilibrate to the detection temperature during passage through the IFC, even at high flow rates (100 µl/min). You may want to maintain the samples in the autosampler compartment at a different temperature for stability reasons. Thermostatting liquid can be circulated through the rack base in the autosampler using an external pump.

4.3 Using application wizards

Application wizards help both new and experienced users to perform their analyses quickly and confidently. An application wizard is a software tool that guides you through the steps in an experiment, from start to final analysis. Procedures and recommendations in the application wizards are based on accumulated expertise from BIACORE analyses. Detailed on-line help can be accessed from the dialogue boxes in the wizards.

Application wizards provided with BIACORE 2000 are described in more detail in Chapter 5.

4.4 Manual operation

This section describes how to operate BIACORE 2000 manually through the software interface. See Section 4.6 for a description of method-controlled operation.

Note: The instrument door is locked for safety reasons during operations which use the autosampler (e.g. sample mixing and injection). Make sure all samples and vials are in place before initiating such operations. These operations cannot be performed if the door is not closed.

4.4.1 Transferring, mixing and diluting samples

Samples can be transferred between vials, mixed and diluted using the autosampler. These operations can be performed without starting a sensorgram.

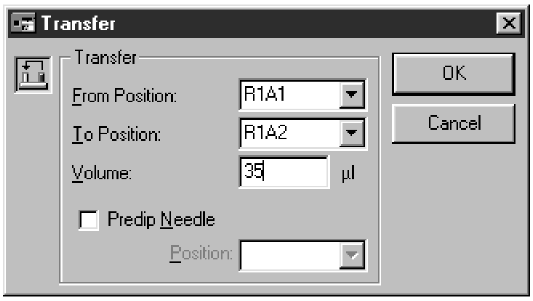
Note: If you perform sample transfer operations while a sensorgram is running, the operations will be recorded in the event log for the sensorgram. This can provide valuable documentation of sample operations for analyses where e.g. the time of sample preparation is critical.

The dialogue boxes for these operations include fields for vial positions in the autosampler racks. The pull-down list associated with each position entry includes all valid rack positions for the currently installed racks (see Section 4.2.4), as well as positions such as **BUFFER** (running buffer), **REC** (recovery cup) and **WASTE**.

There is also a **Predip Needle** option, allowing you to select a position at which the needle will be dipped before the transfer operation. This operation also aspirates 10 µl from the predip position into the autosampler tubing to prevent the sample from coming into contact with running buffer. Use this feature to minimize carry-over between sample vials, e.g. when solutions of widely different composition are used in succession, and also when it is important that the sample in the vial does not become contaminated with running buffer.

Transferring samples

To transfer sample from one rack position to another, select **Command:Transfer**.



Transfer

Transfer

From Position: R1A1

To Position: R1A2

Volume: 35 µl

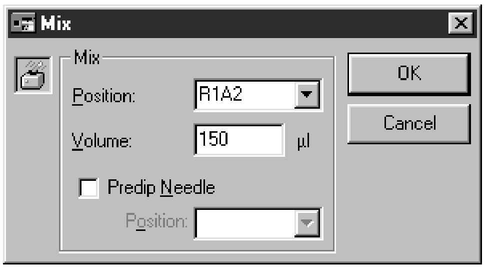
□ Predip Needle

Position:

Enter the required rack positions and volume, then click on **OK**.

Mixing samples

To mix sample in one position, select **Command:Mix**. Use this command for example after transferring two different solutions to a common vial to prepare a homogeneous mixture.



Mix

Mix

Position: R1A2

Volume: 150 µl

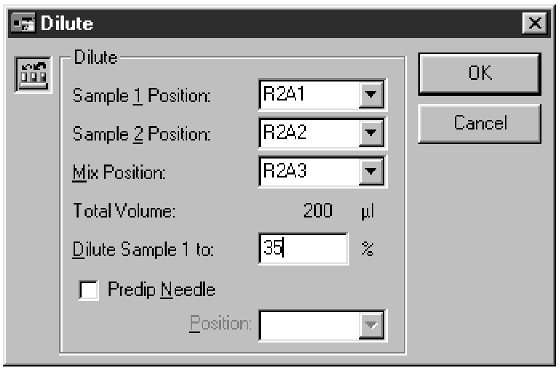
□ Predip Needle

Position:

Enter the required rack position and appropriate volume to be mixed (maximum 450 µl), then click on **OK**. As a general guideline, mix using 75% of the total volume in the vial. The mixing procedure aspirates the specified volume of sample and then dispenses it to the same vial. This is performed three times.

Diluting samples

To dilute a sample, select **Command:Dilute**. This command can also be used instead of two successive **Transfer** operations to prepare a mixture of two samples.



Dilute

Sample 1 Position: R2A1

Sample 2 Position: R2A2

Mix Position: R2A3

Total Volume: 200 µl

Dilute Sample 1 to: 35 %

□ Predip Needle

Position:

This command prepares a dilution of sample 1 in sample 2 at the specified mixing position. The final volume of the diluted sample is always 200 µl (for example, requesting a dilution to 20% will take 40 µl sample 1 and 160 µl sample 2). The dilution value must be in the range 5-95%. For best precision, use values in the range 10-90%.

One **Dilute** command performs the equivalent of two **Transfer** commands followed by a **Mix**. For example,

Dilute sample 1 R1A1, sample 2 R1A2, mix position R1A3, dilution 20%.	*is equivalent to*	1. Transfer 40 µl from R1A1 to R1A3. 2. Transfer 160 µl from R1A2 to R1A3. 3. Mix in R1A3.

Note: **Dilute** is optimized for best performance and should be used where possible in preference to **Transfer** and **Mix**.

For best repeatability between dilution operations, always specify the solutions in the same order in the dialogue box.

4.4.2 Data collection rate

The data collection rate determines the number of points collected per unit time during the sensorgram (i.e. the time resolution of the analysis). Data collection rate can be set to **Low**, **Medium** or **High** with the **Run:Data Collection Rate** command. Actual collection rates for various settings are listed in the table below.

	Detection mode	
Data collection rate	Single	Multichannel
Low	0.1 point/s	0.1 point/s
Medium	1.0 point/s	1.0 point/s
High	10.0 points/s	2.0 points/s

Note: Values in multichannel mode refer to the collection rate per channel.

The default setting of **Medium** is adequate for most purposes. Set the rate to **High** for applications requiring very high resolution. Set the rate to **Low** to reduce the size of the result file for long runs.

A reminder is issued if you start a run with **Data Collection Rate** set to **High** or **Low**.

4.4.3 Starting a manual sensorgram

With a sensor chip docked, choose **Run:Run Sensorgram** to start recording data.

Whenever you start a sensorgram manually, you will be asked to select the detection mode and flow rate. See Sections 4.4.4 and 4.4.5 for a description of the dialogue boxes. The most recently used values are suggested for these settings.

If **Autosave** is enabled for manual sensorgrams (see Section 9.9), you will also be asked to specify the filename for saving the results.

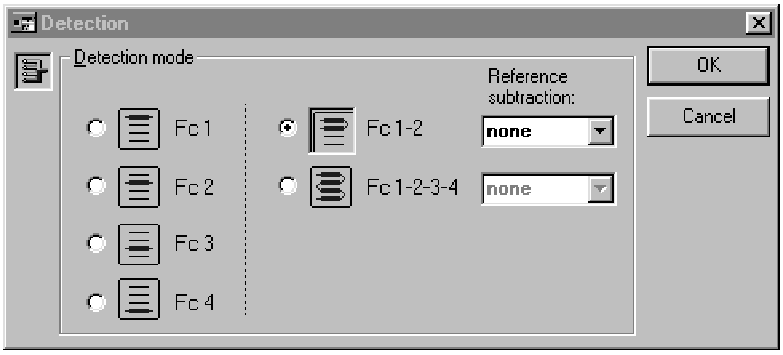
The sensorgram will start a few seconds after you have accepted the last dialogue box. During this delay, the pumps are filled and the needle is washed briefly in preparation for the run. Various flow rate and flow path settings appear in the status window at the sensorgram start, regardless of the settings you have requested. This is a normal part of the washing procedure. Your requested flow settings will be established before the sensorgram actually starts.

Once the sensorgram is started, data will be recorded until you choose **Run:Stop Sensorgram**. Events such as sample injections during the sensorgram are recorded in the event log. You can save the sensorgram with **File:Save** or **File:Save As** at any time without interrupting recording (see Section 4.4.12). However, once you have stopped a sensorgram you cannot resume recording except as a new sensorgram, either in a new file or appended to the previous file.

See Section 4.8 for a description of sensorgram display facilities.

4.4.4 Setting the detection mode and flow path

You set the detection mode and flow path at the start of the sensorgram from the **Detection** dialogue box:



Detection

Detection mode

○ Fc 1

○ Fc 2

○ Fc 3

○ Fc 4

● Fc 1-2 — Reference subtraction: none

○ Fc 1-2-3-4 — Reference subtraction: none

The choice in this dialogue box determines both the flow path and detection mode at the start of the run. Sensorgrams are recorded only from the flow cells in the chosen flow path. You can change the flow path during a run with **Command:Flow path**: the detection mode will however not be changed and the new flow path must a subset of the original (e.g. you can change from Fc 1-2-3-4 to Fc 1-2 but not vice versa). The current detection mode and flow path are shown in the status window during a sensorgram.

When multiple flow cells are used, buffer flows through the cells in series (see Section 2.1.4, Figure 2-8). The dead volume between the flow cells is 0.3 µl). At a flow rate of 20 µl/min, the delay between the flow cells is about 1 s, corresponding to 1 data point at default data collection rate.

Reference curve subtraction

In multichannel mode, flow cell 1 and/or 3 can be used as a blank cell for on-line reference subtraction, providing direct recording and display of blank-corrected sensorgrams. This choice is made in the **Detection** dialogue box at the start of the sensorgram and cannot be changed during a sensorgram. Reference curve subtraction automatically corrects for the small time delay between the flow cells.

The available settings for reference curve subtraction are summarized below:

Flow path	Reference subtraction
Fc 1-2	2-1*
Fc 1-2-3-4	2-1 and 4-3 *or* 2-1, 3-1 and 4-1

* '2-1' means that the response in flow cell 1 is subtracted from that in flow cell 2, i.e. Fc 1 is the blank for Fc 2.

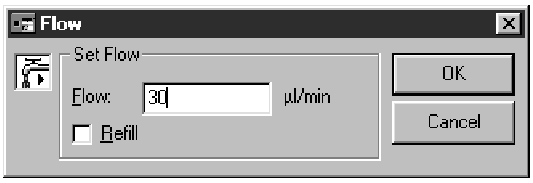
Choose **None** in the **Reference cell** box if you do not want reference subtraction.

Note: If you run a sensorgram without reference curve subtraction and want to subtract the reference curve on a later occasion, use BIAevaluation software.

Multiple sensorgram display
You can use the **View:Overlay, View:Plot Curve Classes** and **Window:New Window** commands to control how sensorgrams are displayed. See Section 4.8 for details. Creating multiple windows with **Window:New Window** is particularly useful when reference curve subtraction is used, to display original and subtracted curves simultaneously on different scales.

4.4.5 Setting the flow rate

To change the flow rate, choose **Command:Flow** and enter the required flow rate in the dialogue box. The new flow rate will apply when you click on **OK**.



Flow

Set Flow

Flow: 30 µl/min

□ Refill

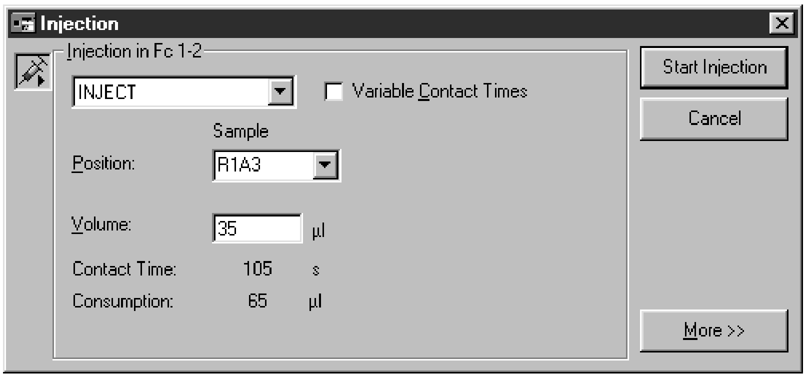
The flow will continue at the set rate until you stop or change the flow. To stop the flow, either choose **Command:Stop Flow** or set the flow rate to zero. You can stop the flow while a sensorgram is running. Note however that injections cannot be performed while the flow is stopped.

Filling the continuous flow pump
The pumps are filled when a sensorgram is started, and are refilled automatically at the end of every pump stroke (or at the start and end of each cycle in method control). The stroke volume is 500 µl, and the refill operation takes about 7 seconds. To avoid disturbances arising from the refill operation during a critical analysis, you can refill the pump at any time in a manual run. Choose **Command:Flow** and check the **Refill** box. The pump will be refilled when you click on **OK**, even if you have not changed the flow rate.

The start and end of the refill operation (both automatic and manually requested) are recorded in the event log if a sensorgram is running, so that you can identify possible signal disturbances arising from this source.

4.4.6 Injecting sample

To inject sample over the sensor chip surface, choose **Command:Inject**. You cannot choose this command unless a sensorgram is running.



Injection in Fc 1-2

INJECT

□ Variable Contact Times

Sample

Position: R1A3

Volume: 35 µl

Contact Time: 105 s

Consumption: 65 µl

Inject commands for different application purposes are summarized below.

Command	***Comments***	***Injected volume***	***Sample consumption***	***Max flow rate***
INJECT	General purpose inject.	5-325 µl	Injected vol +30 µl	100 µl/min
KINJECT	Low dispersion inject for kinetics and for samples sensitive to dilution with running buffer. Controlled dissociation time after sample injection.	10-250 µl	Injected vol +40 µl	100 µl/min
QUICKINJECT	Low sample consumption. Less preparation time than INJECT but more sample dispersion.	5-325 µl	Injected vol +10 µl	100 µl/min
COINJECT	Injection of second sample immediately following the first (e.g. for dissociation in the presence of ligand). Variable contact times for first sample only. Recovery of either first or second sample, but not both.	Sample 1: 10-100 µl Sample 2: 15-250 µl	Sample 1: injected vol +40 µl Sample 2: injected vol +40 µl	100 µl/min
BIGINJECT	Large volume inject. Direct inject up to 325 µl, sample loop between 326 and 425 µl, direct between 426 and 750 µl.	326-750 µl	Injected vol +52 µl	50 µl/min*
MANUAL INJECT	Manually controlled inject. See Section 4.3.9.	1-100 µl / loading	Loaded vol +30 µl	100 µl/min

*BIGINJECT cannot be performed at higher flow rates because the autosampler tubing must be refilled during injection from the sample loop for samples larger than 425 µl.

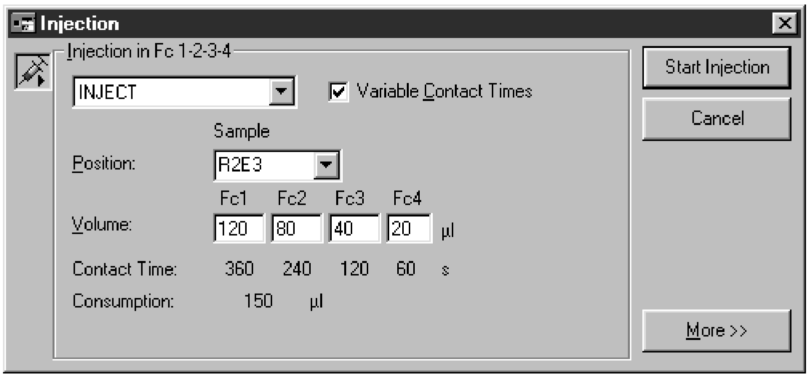
The dialogue boxes for the various inject commands (except MANUAL INJECT, see Section 4.4.8) have a number of common features:

- **Position** fields have a pull-down list of valid positions according to the rack settings.
- You enter the volume of sample to be injected in the **Volume** field(s). The dialogue box also displays contact time of the sample with the sensor surface (contact time = injected volume/flow rate) and the sample consumption for the injection (see table above).
- If the **Variable Contact Times** box is checked, volume fields appear for the flow cells in the current flow path (e.g. if you have set flow path 1-2-3-4 there will be four volume fields for an INJECT command). Each volume must be less than or equal to the volume in

the preceding flow cell. For example, with four flow cells:

Permitted:	40 30 20 10
	40 40 20 20
	40 25 25 10
Not permitted:	40 20 ***30*** 10
	40 30 20 ***30***

Variable contact time injections are implemented by switching additional flow cells into the flow path as the injection proceeds. Injections in the separate flow cells thus start at different times and end at the same time. This is illustrated by the variable contact time sensorgrams shown below.



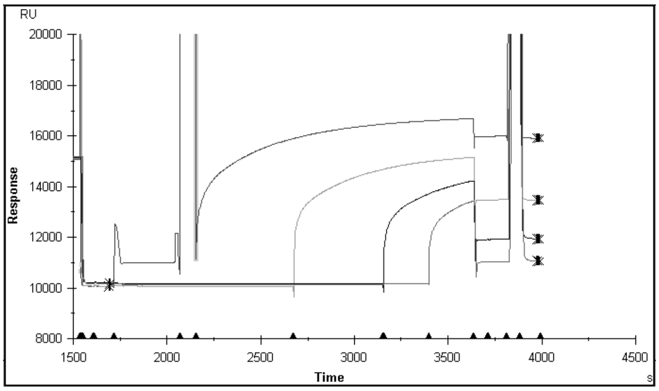
Injection in Fc 1-2-3-4

INJECT ☑ Variable Contact Times

Position: Sample R2E3

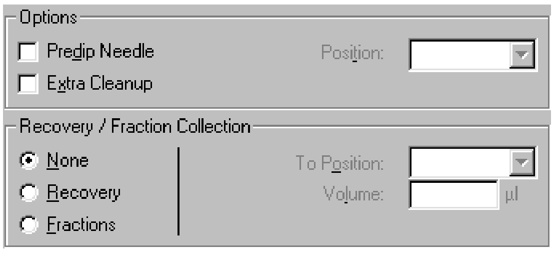
	Fc1	Fc2	Fc3	Fc4	
Volume:	120	80	40	20	µl
Contact Time:	360	240	120	60	s

Consumption: 150 µl



Variable Contact Times is only available if you have chosen a multichannel flow path and are not using reference curve subtraction.

The **More>>** button opens additional options for the injection:



Options

□ Predip Needle Position:

□ Extra Cleanup

Recovery / Fraction Collection

● None To Position:

○ Recovery Volume: µl

○ Fractions

Check **Predip Needle** and enter a rack position to dip the needle in the specified position before aspirating sample. This operation also aspirates 10 µl from the predip position into the autosampler tubing to prevent the sample from coming into contact with running buffer. Use this feature to minimize carry-over between sample vials, e.g. when solutions of widely different composition are used in succession. Use of **Predip Needle** in an extra vial of sample can also help to minimize sample dispersion in the injection by improving the separation between injected sample and running buffer.

Check **Extra Cleanup** to perform an additional wash of the needle and IFC after the injection. Use this feature to minimize carry-over when the next injection uses sample of widely different composition. The IFC is routinely washed as part of the standard injection sequence. **Extra Cleanup** should be used if particular care is necessary, e.g. after injection of regeneration solution.

The sample recovery and fraction collection options are described below (Sections 4.4.9 and 4.4.10).

Click on **Start Injection** to start the injection. The system takes a while to prepare the injection, varying from about 0.5 up to 7 minutes depending on the kind of injection, the sample volumes and the options chosen (see table below). The event log records both the time when the injection was requested and the time at the actual start and end of sample injection.

Inject command	*Injected volume (µl)*	*Preparation time (s)*	*Finishing time (s)*	*No. of air segments*
INJECT	20 100	60 70	40 45	2
QUICKINJECT	20 100	40 30	25 30	1
KINJECT	20 100	80 90	40 40	3
COINJECT	20, 20 100, 100	185 250	80 80	3
BIGINJECT	350 750	345 340	90 95	3
MANUAL INJECT	Preparation time for loop loading 100 µl: 80 s		85	2

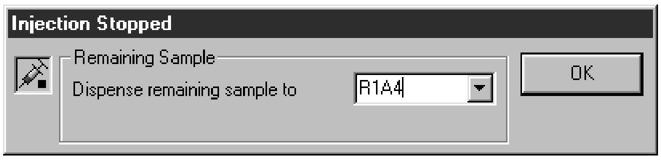
Approximate times for preparation and completion of various inject commands at a flow rate of 20 µl/min. Preparation time is the time between the inject command and the actual start of sample injection. Finishing time is the time between the end of sample injection and completion of the command, during which the flow system is washed. Finishing times for KINJECT do not include the dissociation time specified in the command.

***Predip Needle** adds about 8 s to the preparation time. **Extra Cleanup** adds about 30-40 s to the finishing times for INJECT, QUICKINJECT and KINJECT and about 70 s to the finishing times for COINJECT and BIGINJECT.. **Recover** of 25 µl adds about 210 s to the finishing times.*

4.4.7 Stopping an injection

To stop an injection, choose **Command:Stop Inject** or click on the **Stop Inject** toolbar button. The injection stops as soon as the command is chosen. Once an injection has been stopped in this way it cannot be resumed.

Choose what to do with the remaining sample. You may dispense remaining sample either to waste or to a specified rack position. The door is unlocked as long as the dialogue box is displayed so that you can insert an empty vial if required to receive the remaining sample.



Injection Stopped

Remaining Sample

Dispense remaining sample to R1A4

When the remaining sample has been dispensed to the chosen position, the flow system is washed in the same way as when the injection is completed normally. You cannot stop an injection during the washing procedures which follow the normal sample injection.

Notes: If you stop an injection for which sample recovery is requested (see Section 4.4.9), the volume of recovered sample will be reduced in accordance with the lower volume injected. If you stop an injection for which fraction collection is requested (see Section 4.4.10), fractions will not be collected.

During the first injection for COINJECT, the **Stop Inject** command changes to **Skip Inject**. Choosing this command starts the second injection immediately. Sample remaining in the sample loop from the first injection is flushed to waste after the injection and cannot be recovered. The second injection can be stopped in the same way as other injection commands.

For BIGINJECT, you can recover sample which is currently in the autosampler tubing when the injection is stopped, but not that which is in the sample loop in the IFC.

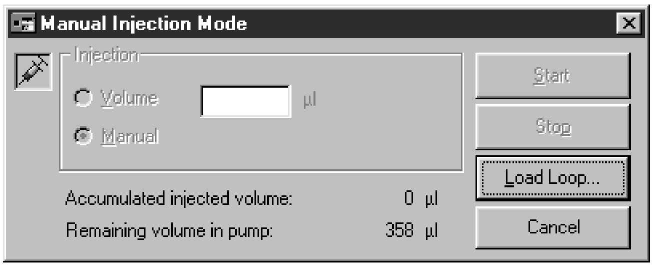
See Section 4.4.14 for emergency stop procedures.

4.4.8 Manual injections

The MANUAL INJECT feature provides complete manual control of sample injection, with facilities for pausing and resuming the injection, changing flow rate, refilling the sample loop etc without ending the injection. This mode is useful for development work, e.g. for scouting for the minimum volume of regeneration solution needed to remove bound analyte completely from the surface.

Manual injections use the sample loop in the IFC, allowing uninterrupted injection of up to 100 µl. Direct injection from the autosampler tubing is not used in this mode.

Choose **MANUAL INJECT** from the pull-down list of inject commands in the **Inject** dialogue box and click on **Continue>>**.



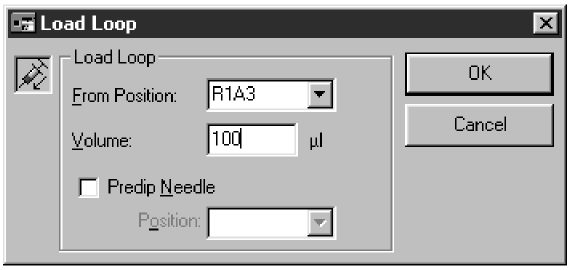
○ Volume [ ] µl
● Manual

Accumulated injected volume: 0 µl
Remaining volume in pump: 358 µl

The dialogue box shows the accumulated injected volume and the volume of buffer remaining in the continuous flow pump. This latter figure gives you an indication of when the pump will be refilled, so that you can avoid refill operations during injections.

Follow the steps below to perform a manual injection.

1. Click on **Load Loop**, enter the sample position and the volume to be loaded (maximum 100 µl). The loop is loaded using 2 air segments between sample and buffer to reduce sample dispersion. You can also choose to predip the needle in a specified position to reduce carry-over from solutions of widely differing composition. Click on **OK** to load the loop and return to the manual injection mode dialogue box.



Load Loop

Load Loop

From Position: R1A3

Volume: 100 µl

□ Predip Needle

Position:

Note: The volume specified in this dialogue box is the volume that can be injected. Approximately 30 µl more than this volume will be taken from the sample vial for air segments and loop priming (see Section 2.1.4).

2. Note the volume remaining in the sample injection pump. If this volume is smaller than the volume you intend to inject, you may wish to refill the pump to avoid a pump filling operation during the injection. To refill the pump, choose **Command:Flow** from the menu or click on the **Flow** button on the main toolbar, check the **Refill** box and click on **OK** (see Section 4.4.5).

3. Choose either **Volume** or **Manual** mode:

   Volume The injection will stop automatically after the specified injected volume. Use this mode for injection of predetermined volumes.

   Manual The injection will continue until you click on **Stop**. Use this mode if you want to stop the injection interactively according to the appearance of the sensorgram. ***The injection can continue indefinitely in* Manual *mode. Be careful not to inject more than the volume loaded in the loop.***

   You can change the flow rate during a manual mode injection, but not during a volume mode injection. You can change the flow path in multichannel mode and stop the injection at any time in both modes.

4. Click on **Start**. The dialogue box displays both the accumulated injected volume (i.e. the sum of all volumes injected since the start of manual injection) and the volume remaining in the sample

injection pump. The volume injected since the last **Start** is shown in the sensorgram status window.

As a rule of thumb, do not inject more than 80-90% of the loaded volume (more at higher flow rates) to minimize the risk of sample dispersion effects at the end of the injected sample.

5. To interrupt the injection at any time, click on **Stop**. This will switch from sample to buffer flow over the sensor surface, but will not empty the sample loop or wash the IFC. You can re-load the sample loop while the injection is stopped. You can also use the autosampler to prepare samples using **Mix**, **Transfer** and **Dilute** while the injection is stopped (use the commands in the menu or the buttons on the main toolbar for these operations). Click on **Start** to resume the injection.

6. To end the manual injection, click on **Exit**. This will empty the sample loop to waste and wash the IFC.

4.4.9 Recovering sample

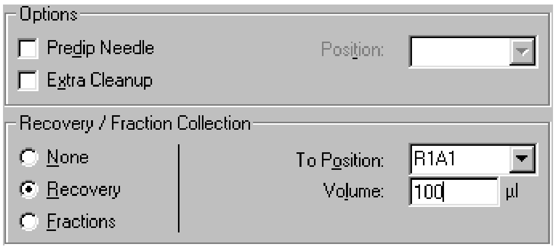
The sample recovery feature can be used either:

- to recover the injected sample, or
- to recover bound analyte that is released during regeneration (this is really recovering “sample” during injection of regeneration solution).

> *IMPORTANT*
> Before starting an injection with sample recovery, perform a **Wash** with **Recovery** option to wash the recovery cup and IFC recovery channel and empty the recovery channel (see Section 4.4.11). If you do not do this the recovered sample may be diluted with up to 15 µl buffer from the recovery channel.

When sample recovery is activated, effluent from the IFC is diverted to the recovery cup on the connector block. From there it is collected by the autosampler and transferred to a specified rack position.

Select recovery in the **More>>** panel of the **Injection** dialogue box. This feature cannot be used in combination with BIGINJECT or MANUAL INJECT or with fraction collection (see below).



Options

□ Predip Needle — Position:

□ Extra Cleanup

Recovery / Fraction Collection

○ None — To Position: R1A1

● Recovery — Volume: 100 µl

○ Fractions

Enter the vial position to which the recovered sample is to be transferred and the volume to be recovered (permitted range 5-350 µl, recommended >25 µl for best precision). The volume delivered to the recovery cup is the same as the injected volume: however, if you want to be sure of recovering all the injected sample, specify 5-10 µl more for recovery than the volume of sample injected so that a small volume of air is “recovered” after the sample.

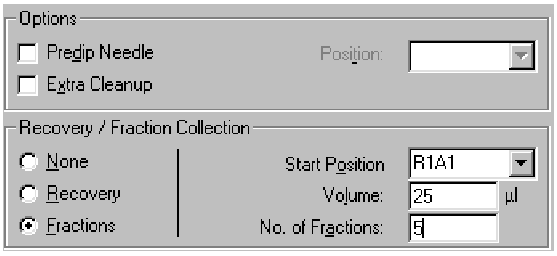
You can use the **Transfer** command to prime the recovery position with neutralizing buffer before the injection. This is particularly relevant when you use recovery following a regeneration injection to collect material released from the surface.

4.4.10 Collecting fractions

The fraction collection feature collects effluent from the IFC during buffer flow immediately following an injection. Use this feature to collect and/or fractionate material which is released from the surface during the dissociation phase.

IMPORTANT
Before starting an injection with fraction collection, perform a **Wash** with **Recovery** option to wash the recovery cup and IFC recovery channel and empty the recovery channel (see Section 4.4.11). If you do not do this the first fractions may be diluted with up to 15 µl buffer from the recovery channel.

Select fraction collection in the **More>>** panel of the **Injection** dialogue box. This feature cannot be used in combination with BIGINJECT, MANUAL INJECT or COINJECT injections or with sample recovery (see above).



Options

□ Predip Needle — Position:

□ Extra Cleanup

Recovery / Fraction Collection

○ None — Start Position: R1A1

○ Recovery — Volume: 25 µl

● Fractions — No. of Fractions: 5

Enter the vial position for the first fraction to be collected, the fraction volume and the number of fractions. Fractions will be collected in sequence from the specified first position. The next fraction after the end of a row is the beginning of the next row (e.g. r1a10 → r1b1). The sequence does not extend between racks. If the end of a rack is reached before the specified number of fractions has been collected, fraction collection is terminated and the effluent goes to waste.

Note: Fractions are marked with event markers on the sensorgram and in the event log. Each marker is placed at the correct time on the sensorgram, but does not appear until the fraction is actually collected from the recovery cup. The delay represents the IFC channel volume between the flow cell and the recovery cup (about 15 µl). For example, at a flow rate of 30 µl/min, the event marker for a fraction collected at 10.0 minutes will not appear until 10.5 minutes, but will be placed at the correct time of 10.0 minutes on the sensorgram.

4.4.11 Washing liquid handling system components

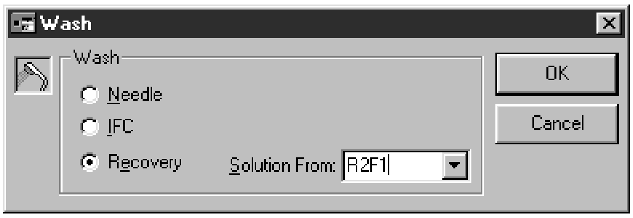
The liquid handling system is washed automatically as required during operation:

- The autosampler needle is washed after each sample handling operation (**Inject**, **Transfer**, **Mix**, **Dilute**).

- The IFC channels are washed after each injection.

- The recovery cup is washed after each injection that uses sample recovery or fraction collection.

The **Wash** command is provided to allow additional washing of the needle and recovery cup. The needle is washed with running buffer. The recovery cup may be washed either with buffer or with solution from a specified vial position. Use this command as follows:



○ Needle
○ IFC
● Recovery
Solution From: R2F1

- Wash the needle whenever you suspect that there is a risk for carry-over which will not be handled by the regular wash, predip or extra cleanup features. This situation can arise especially when solutions of widely different composition (particularly ionic strength or pH) are used, or when different samples contain incompatible substances (e.g. enzyme and substrate). The needle washing operation takes about 20 seconds. Note that the predip feature available with all needle operations greatly reduces carry-over and can in many cases be sufficient.

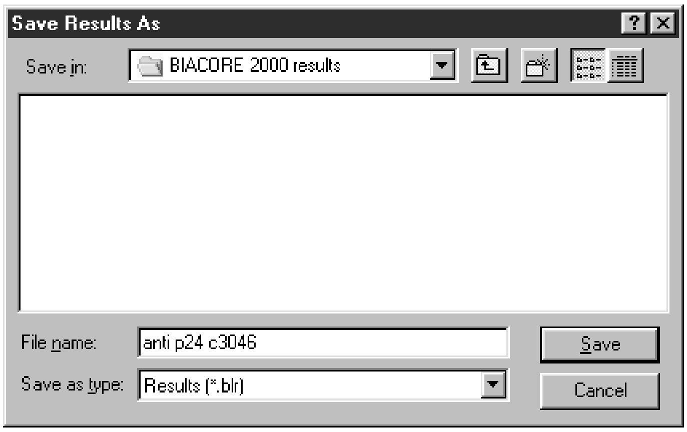
- Wash the recovery cup before any injection using sample recovery or fraction collection, unless you are sure that the cup was last used for the same or a compatible solution. Wash the cup with solution from a specified position when you want to prepare the recovery cup for receipt of sample and to be sure that any residual liquid in the cup has the correct composition. Approximately 0.5 ml liquid is used to wash the recovery cup and recovery channel. The recovery channel is emptied of buffer after the wash operation. This operation takes about 3 minutes.

To wash the IFC flow channels manually, use **Working Tools:Flush** (see Section 7.2.1). This may be necessary if you are using a viscous regeneration solution such as ethylene glycol.

4.4.12 Saving the sensorgram

To save a sensorgram as a BIACORE result file, choose **File:Save** or **File:Save As**. The first time a new sensorgram is saved, these commands are equivalent. Thereafter, **Save** saves the sensorgram under the same name, overwriting the earlier version, while **Save As** allows you to save the sensorgram under a new name, leaving any previous version unchanged.

Result files are saved with the default extension .blr.



Save in: BIACORE 2000 results

File name: anti p24 c3046

Save as type: Results (*.blr)

If the **Autosave** function under **Tools:Preferences** is turned on, the sensorgram will be saved automatically at the specified interval (see Section 9.9). We recommend this feature as a precaution against losing data if the computer should lock up or lose power during a run. If

Autosave is turned on, you must specify the file name for the sensorgram at the start of the run.

To save a sensorgram and associated data or the report point table in a format which can be opened by other programs, choose **File:Export** and select either **Curves** or **Report Point Table** from the sub-menu. This creates a text file with tab-separated data, suitable for use in spreadsheets and other applications.

You can save or export sensorgram data at any time during a run without interrupting data collection.

4.4.13 Stopping a sensorgram

To stop data collection, choose **Sensorgram:Stop Sensorgram** or click on the **Stop Sensorgram** toolbar button. The IFC flow channels are washed automatically if an injection has been performed, and the buffer flow is turned off.

Note: Once you have stopped a sensorgram, you cannot resume data collection in the same sensorgram. If you need to combine data from two separate runs, you must use curve presentation features in BIAevaluation or equivalent software.

4.4.14 Emergency stop procedure

To abort a run at any time, press **Ctrl-Break**. (The **Break** key is also marked **Pause** on standard keyboards.) This will immediately stop all current operations including buffer flow and data collection. You should normally wash the liquid handling system in connection with the emergency stop since no automatic washing procedures are performed. There are no facilities for recovering sample in this situation.

Do not use Ctrl-Break unless you need to stop all operations *immediately* (e.g. if continued operation constitutes a risk of damaging the instrument or losing valuable material). A controlled stop using **Stop Inject** and **Stop Sensorgram** is always preferable to Ctrl-Break.

4.4.15 Undocking the sensor chip

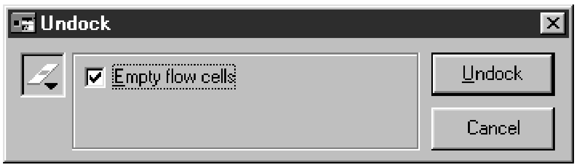
You can leave the sensor chip docked in the instrument for several days provided that liquid is kept flowing over the surface. You will however need to undock the sensor chip:

- when all four flow cells are used and you need a new chip.

- when the stability of your immobilized ligand requires that the chip should be stored under special conditions.

- when you are shutting down the system for more than 5 days.

To undock the sensor chip, choose **Command:Undock**. ***Do not attempt to remove the chip without first undocking.***



If the **Empty flow cells** box is checked, the system will empty the flow cells before undocking. Always check this option unless it is essential to keep the sensor surface wet (some ligands and special surfaces may deteriorate on contact with air). Liquid will not leak into the optical system even if you undock the chip without emptying the flow cells.

When the undocking is complete (the sensor chip LED on the instrument front panel flashes), open the sensor chip port cover and remove the sensor chip cassette. The software will automatically prompt you to insert and dock a new chip.

4.5 Queuing manual commands

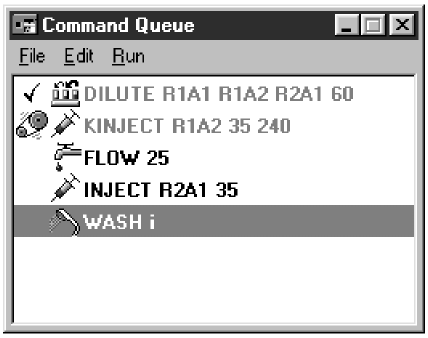
The command queue allows you to queue manual commands so that you do not have to wait until the current command is completed. Commands issued in manual mode are placed in the queue window and are executed as soon as the instrument activity permits. If the instrument is busy (e.g. during an injection), queued commands wait in order until the current operation is completed. You can also save the commands in the queue window as a simple method for later use. This means that you can use an interactive manual control session as the starting point for creating customized methods.

If the queue is not active, instrument commands in the toolbar and the **Command** menu are grey and cannot be used as long as the instrument is busy. Commands issued when the queue is not active cannot be used to generate a method.

4.5.1 Starting the command queue

Depending on the setting under **Tools:Preferences**, the command queue may be started automatically when a sensorgram is started. If this is not the case, start the queue by choosing **Command:Command Queue**. You can only start the queue while a sensorgram is running.

When the command queue is active, the queue window is shown either minimized on the Windows taskbar or expanded in the sensorgram window area. The expanded window shows all commands in the queue, including those which have been completed (you may have to scroll the window to see all commands if the queue is long).



```
✓ DILUTE R1A1 R1A2 R2A1 60
  KINJECT R1A2 35 240
  FLOW 25
  INJECT R2A1 35
  WASH i
```

The command queue window. A KINJECT command is in progress.

To expand the minimized queue window, simply click on the button on the taskbar. You can resize the expanded window by dragging on the window borders (the size of the minimized window is fixed). To minimize the expanded window, click on the **Minimize** button on the window toolbar (standard Windows functionality). The expanded window can be moved anywhere on the screen by dragging. The minimized window can also be moved, but will always snap to the nearest corner of the sensorgram window.

4.5.2 Command queue window display

Commands are displayed in the command queue window as follows:

The minimized window displays an icon for the command currently being executed. The icon is the same as that which appears in the dialogue box for the command (and on the toolbar for commands with toolbar buttons). The window shows a **Stop** icon if the queue is stopped, or no icon if execution of the queue is complete.

The expanded window shows 3 columns for each command (see illustration above), with command status, command icon and method command(s). Commands which have been executed are grey.

Command status
The command status is indicated by an icon:

✓ indicates that the command has been completed (the command text is also grey)

⚙ indicates the command currently being executed

✋ indicates that the queue will stop at this point

STOP indicates that the queue has stopped at this point

💣 indicates that this command has failed (either through instrument failure or because the user has pressed **Ctrl-Break**)

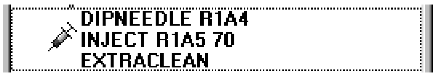
Commands which are waiting to be processed have no status icon.

Command icon
This icon is the same as that which appears in the dialogue box for the command (and on the toolbar for commands with toolbar buttons). The following icons are not included in the toolbar but are used in the command queue:

Dilute

Wash

Method command(s)
The method command(s) column shows the command as it would appear in a method. For some commands, there may be more than one method command for a single command icon: for example, an **Inject** command with options **Predip needle** and **Extraclean** appears as follows:



```
DIPNEEDLE R1A4
INJECT R1A5 70
EXTRACLEAN
```

4.5.3 Managing the command queue

Adding commands to the queue
To add a command to the queue, simply choose the command from the toolbar or the **Command** menu. The command will be placed at

the end of the queue as soon as you click on **OK** in the dialogue box for the command.

Note: The **Manual inject** command cannot be queued.

If the queue window is minimized, you will not see commands added at the end of the queue. Expand the window to display all queued commands.

Normally, commands in the queue are executed as soon as possible. If the queue is empty, a requested command will be executed immediately. If you want to hold the queue while you add commands, choose **Run:Stop Queue** from the queue window menu before you add instrument commands (see below).

Deleting commands from the queue

To delete commands from the queue, select one or more commands in the queue window and choose **Edit:Remove** from the command queue window menu. You will be asked to confirm deletion of the selected command(s). A temporary **Stop** icon will be placed at the first selected command, and execution will stop if the queue reaches this command while the confirmation dialogue box is open.You can delete commands that have been executed ("grey" commands) and commands that are waiting to be executed, but you cannot delete the command that is currently being executed in the instrument.

To delete all commands in the queue (including "grey" commands), choose **Edit:Clear Queue**. The queue will be cleared immediately, but execution of the current command will continue to completion.

Editing commands in the queue

You can edit commands in the queue provided that they have not been executed or are currently being executed. Commands that are completed ("grey" commands) and the current command cannot be edited.

To edit a command, double click on the command or select the command and choose **Edit:Edit Command**. This will place a temporary **Stop** icon at the command and open the appropriate dialogue box for the command. Execution will stop if the queue reaches this command while the **Edit** dialogue box is open. Change the settings as required in the dialogue box. The **Stop** icon will be removed when you click on **OK** to implement the changed command.

You can only change the settings within the same command in this way. To change one command to another (e.g. to change **Transfer** to **Dilute**), you must delete the old command and insert the new one.

Inserting commands in the queue

To insert a command at a given place in the queue, select the command directly after the insertion position and choose **Edit:Insert Command**. This will insert a blank line in the queue with a temporary **Stop** icon on the blank line. Execution will stop if the queue reaches this point while the command dialogue box is open. With the blank line selected, choose the command you wish to insert. Execution will stop if the queue reaches this point while the command dialogue box is open. The **Stop** icon will be removed when you click on **OK** to implement the inserted command.

You can only insert commands in the queue after the command currently being executed.

Stopping the queue

To stop the queue before a selected command, click on the command and choose **Run:Stop at Command** from the command queue window menu. This will insert a **Stop at Command** line in the queue immediately before the selected command. When execution reaches a **Stop at command**, the queue stops and the line status changes to a **Stop** icon. To continue execution after a **Stop**, choose **Run:Start queue** from the command queue window menu.

To remove a **Stop at Command** line which has not yet been reached in the queue, select the line and choose **Edit:Remove**.

To stop the queue after execution of the current command has completed, choose **Run:Stop Queue** from the command queue window menu. This places a **Stop at Command** line directly after the current command. You can hold the queue before inserting any instrument commands by choosing **Stop Queue** when the queue is empty.

To stop the queue immediately, press **Ctrl-Break** (see Section 4.4.14). This will stop the sensorgram and may abort the current command, which will then be marked with a **Failed** status icon in the queue.

Note: The queue window remains open after a **Ctrl-Break** stop, and must be closed before a new sensorgram can be started. You can generate a method from the queue before closing the queue window.

Closing the queue

To close the command queue, choose **File:Close** from the queue window menu or **Command:Command Queue** from the main window menu. Closing the queue will clear any queued commands. The sensorgram will however continue to run.

Stopping the sensorgram
You can only stop the sensorgram while commands are waiting for execution if the command queue is stopped. If there are commands in the queue, you will be warned that stopping the sensorgram will clear the queue. If you want to generate a method from the queued commands, do this before stopping the sensorgram. Stopping the sensorgram closes the command queue window.

Emergency stop procedure (Ctrl-Break)
If you press **Ctrl-Break** while commands are queued, the run will be aborted but the commands will remain in the queue window and can be saved as a method. You must close the queue window before you can start another sensorgram.

4.5.4 Generating methods from the command queue

To generate a method from the command queue, choose **File:Generate Method** from the command queue window menu. This will generate a method in a method editor window containing:

- introductory comments including the control software version from which the method was generated and a date and time stamp

- One `APROG` ("CommandQueue") containing the commands from the queue window. If the queue contains a command with **Failed** status (as a result of either technical failure or user intervention with Ctrl-Break), this will be marked with a comment in the method.

- a `MAIN` block which sets the thermo-racks, flow path and flow rate, calls the `APROG` CommandQueue and finishes with an `APPEND standby` instruction.

Stop at command lines do not generate any command in the method.

The method is complete in itself, and can be saved to disk. Running it will repeat the actions in the command queue. If you want to use a generated method as the starting point for your own method development, bear the following points in mind:

- Each method generation creates a self-contained method with an `APROG` with the same name. If you want to combine `APROG`s from different generated methods (using cut-and-paste in the method editor), remember to change the names in the `DEFINE APROG` blocks and the corresponding calls in the `MAIN` block.

- A generated method includes all commands in the queue at the time of generation. You may want to delete superfluous commands from the queue before generating the method.

- Commands in the generated method are not time controlled, and report points added during the manual (queue-controlled) run are not included in the method. You may wish to add time specifications and report points to the generated method.

Note: Method commands are generated exactly as they appear in the queue window. If you choose **Stop inject** during a queued injection, the generated method will include an `INJECT` command for the volume originally requested, not for the volume actually delivered. If you generate a method in connection with development work where you have stopped injections interactively, you may need to modify the generated method in accordance with the event log of the actual run.

4.5.5 Command queue window menu

In the expanded form, the command queue window has a menu bar with the following functions:

File:Generate Method	Generates a method from the commands in the queue. The method will appear in a new untitled method editor window. See Section 4.5.4 for details.
File:Close	Closes the command queue. This operation clears all commands from the queue. When the queue is closed, commands requested from the **Command** menu are executed directly without being queued.
Edit:Copy	Copies marked commands from the queue to the method editor clipboard. Copied commands can be pasted in a method editor window. They cannot however be pasted in another position in the queue.
Edit:Insert	Inserts a new command in the queue immediately before the marked command. Commands cannot be inserted before the command currently being executed.
Edit:Delete	Deletes the marked command(s) from the queue.

Edit:Clear Queue	Deletes all commands from the queue, including any which have not yet been executed. Does not close the command queue window.
Edit:Edit Command	Opens the appropriate dialogue box for editing parameters for the marked command. Only commands that have not yet been executed can be edited.
Run:Start queue	Starts queue execution if it is currently stopped (see Section 4.4.3). This command changes to **Stop queue** when the queue is running.
Run:Stop Queue	Stops the queue after execution of the current command. This command changes to **Start queue** when the queue is stopped. Note that stopping the queue is not the same as closing it: commands issued while the queue is stopped will be placed in the queue and executed when the queue is started.
Run:Stop at Command	Places a **Stop queue** instruction immediately before the marked command. The queue will stop when this point is reached.

4.6 Method-controlled operation

BIACORE 2000 can be controlled from a pre-programmed method for unattended operation. A method can include any number of analysis cycles, each represented by a sensorgram in the result file. This section describes how to operate BIACORE 2000 under method control.

Note: The instrument door is locked during method operation, except when operator access to the autosampler is required. Method instructions that require this access are placed in a special `DEFINE ALERT` block in the method (see Appendix E).

4.6.1 Creating a method

To create a new method, choose **File:New Method** from the BIACORE Control Software menu. Follow the steps below to enter an elementary method which can be used to illustrate method-controlled operation. Refer to Appendices E and F for details of method programming.

Note: You cannot use the method editor while a sensorgram is running.

Type the following lines in the method editor window:

```
define aprog practice
flow 30
dilute r1a1 r1a2 r1a3 50
inject r1a3 25
end

main
detection 1
aprog practice
end
```

This is a complete method which will prepare in position r1a3 200 µl of a 1:1 mixture of samples from r1a1 and r1a2, then inject 25 µl of this mixture over the sensor chip surface in flow cell 1 at a flow rate of 30 µl/min.

Choose **Edit:Adjust Method:Minutes**. This will adjust the method appearance so that keywords are set in capitals, parameter names are in lower case, and commands are indented to show the block structure of the method.

```
DEFINE APROG practice
    FLOW 30
    DILUTE r1a1 r1a2 r1a3 50
    INJECT r1a3 25
END

MAIN
  DETECTION 1
  APROG      practice
END
```

The **Seconds** and **Minutes** options adjust time references in the method to the respective time base. There are no time references in this example, so the choice of option is irrelevant in this case. See Section 9.5 for a description of time base adjustment.

4.6.2 Checking and saving the method

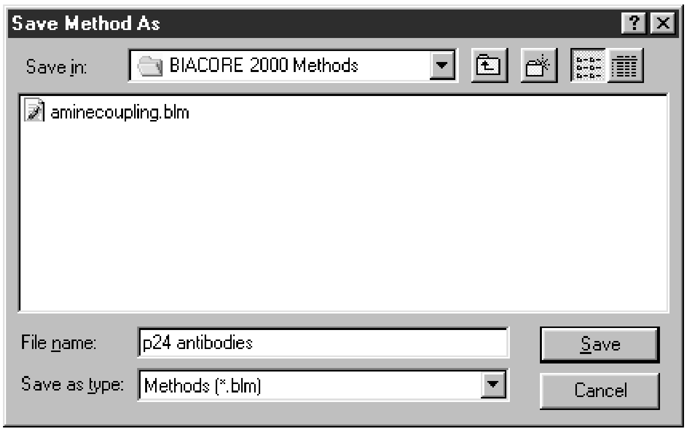
It is important that method instructions are entered correctly, since a method with syntax errors cannot be run. The prerun facility checks the method syntax and identifies incorrect instructions.

Note: The prerun facility can only check the method *syntax*, not the *intention* of the method. It is your responsibility to ensure that the method instructions perform the right operations (e.g. that

INJECT commands address the correct rack position and sample volume). A method with syntax errors can be saved but cannot be run.

Choose **Run:Prerun Method**. Refer to Section E.5 if you receive any syntax error messages. Correct syntax errors and repeat the prerun procedure.

Choose **File:Save As** to save the method. Method files are saved with the extension .blm.



Save Method As

Save in: BIACORE 2000 Methods

aminecoupling.blm

File name: p24 antibodies

Save as type: Methods (*.blm)

4.6.3 Running a method

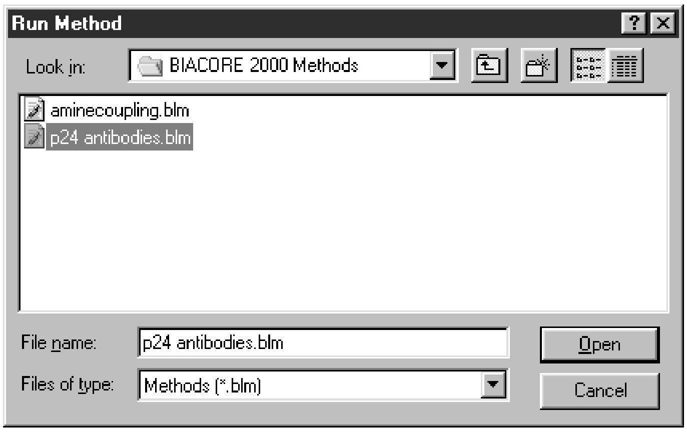
Methods are executed with the command **Run:Run Method**.

If a method editor window is active, this command will attempt to run the method in the window. If you have made changes to the method since it was last saved (or if the method has not been saved at all), you are given an opportunity to save the method before the run starts. The method definition is also saved in the result file from the run.

Before starting a method, check that the rack base definition is correct for the method and that the correct racks are placed in the autosampler compartment. See Section 4.2.4 and the description of the RACK command in Appendix F for more details.

Note: When the run starts, the method editor window will be hidden, whether or not you have chosen to save the method. The window will reappear when the method ends.

If no method window is active, the **Run Method** command prompts for a method to open.



Run Method

Look in: BIACORE 2000 Methods

aminecoupling.blm

p24 antibodies.blm

File name: p24 antibodies.blm

Files of type: Methods (*.blm)

A prerun is performed automatically whenever you execute a method. If there are syntax errors in the method, these will be identified in the method editor window. You must correct the errors and choose **Run:Run Method** again to start the method.

If the **Autosave** function is activated for method runs (under **Tools:Preferences**, see Section 9.9), you will be asked to provide a result file name at the beginning of the run. This file will be used to save the results at the end of each method cycle.

If the detection mode and flow path are not specified with a `DETECTION` instruction at the beginning of the `MAIN` block (see Section E.3.1), you will be prompted for these settings when the method starts.

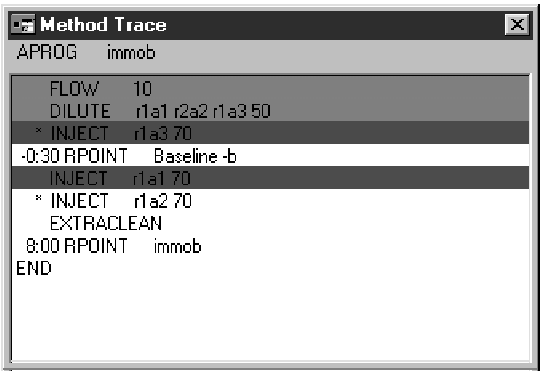
You can use the sensorgram display functions described in Section 4.8 at any time during the run.

At the end of each `APROG` in a method, the needle is washed and the flow rate is set to 10 µl/min. The flow path is set to the path that applied when the `APROG` was started. At the end of the method, the IFC flow channels are washed automatically and the flow rate is set to 0. The flow path is set to an undefined state (all valves are open). Completion of the method is indicated by a **Method ready** message box.

Choose **File:Save As** to save the sensorgrams from the method run in a result file. The result file will contain one or four sensorgrams for each cycle in the method according to the detection mode.

4.6.4 Tracing method operation

During a method-controlled run, the **View** menu includes a **Method Trace** command. Choosing this command displays a method trace window where the progress of the method can be followed.



```
APROG    immob
    FLOW      10
    DILUTE    r1a1 r2a2 r1a3 50
  * INJECT    r1a3 70
-0:30 RPOINT    Baseline -b
    INJECT    r1a1 70
  * INJECT    r1a2 70
    EXTRACLEAN
 8:00 RPOINT    immob
END
```

The method trace window shows the current MAIN instruction at the top of the window. If an APROG is in progress, the APROG instructions are displayed below the MAIN instruction. Instructions are colour-coded as follows:

Grey Instructions which have already been executed (for technical reasons, this includes instructions which have not yet been reached but which do not result directly in instrument activity, such as the RPOINT instructions in the example above).

Green The instruction which is currently being executed in the instrument.

Red Instructions which have been sent to the instrument from the control software and are waiting for instrument resources to become available.

4.6.5 Stopping a method

To stop a method, choose **Run:Stop Method**. The method will continue to the end of the current command in the MAIN block (usually the end of the current APROG) and then stop. The flow system is washed automatically when the method stops.

You can stop a method immediately using the emergency stop Ctrl-Break (see Section 4.4.14). Note that if the current command in the MAIN block cannot be completed (e.g. if the temperature requested by a TEMP command cannot be achieved), you must use the emergency stop Ctrl-Break to stop the method.

4.7 Closing down the system

4.7.1 BIACORE 2000 Processing Unit

Follow one of the procedures below to close down the BIACORE 2000 Processing Unit, according to how long the system will be idle.

Overnight

The instrument can be left overnight with the sensor chip docked and the IFC and flow cells filled with buffer. We recommend that the pumps are allowed to run at a low flow rate (5 µl/min) by means of the **Standby** procedure (see Section 7.2.1). Make sure that there is sufficient buffer in the buffer reservoir (the procedure uses about 5 ml more buffer than required for continuous flow, i.e. 18 h operation requires about 10 ml buffer). As an extra precaution, particularly if your buffer has a tendency to precipitate, you may replace the running buffer with water. Use the working tool **Rinse** to flush the IFC and flow cells with water and then run **Standby** to leave water flowing through the system.

You can append **Standby** to a method to start the procedure automatically when the method is completed (see Appendix F).

Up to five days

To close down the system for up to five days, run **Working Tools:Close** (see Section 7.2.1). This flushes the system with liquid (water is recommended) from a predefined autosampler rack position then empties the IFC and flow cells.

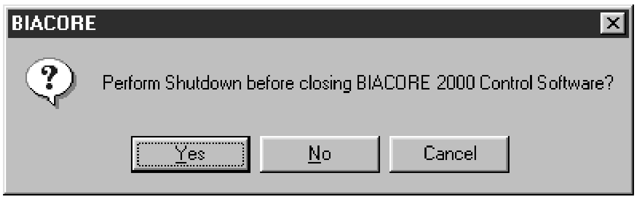
More than five days

To close down the system for more than five days, run **Working Tools:Shutdown** (see Section 7.2.1). Follow the instructions on the screen. You will need water and 70% ethanol for this procedure.

4.7.2 Control Software

To close BIACORE Control Software, choose **File:Exit** or close the window using standard Windows techniques. If the current sensorgram or method has not been saved, you will be prompted to do so before closing the software. You cannot close BIACORE Control Software while a sensorgram is running.

If you ask to close BIACORE Control Software with a sensor chip docked, you are given an opportunity to perform the shutdown procedure.



BIACORE

Perform Shutdown before closing BIACORE 2000 Control Software?

Choose **Yes** to shut down the instrument and close the software, **No** to close the software without shutting down, or **Cancel** to return to the software.

Note: Closing the BIACORE Control Software will not close the BIAevaluation and BIAsimulation applications, even if these were started from the control software.

4.8 Viewing and evaluating results

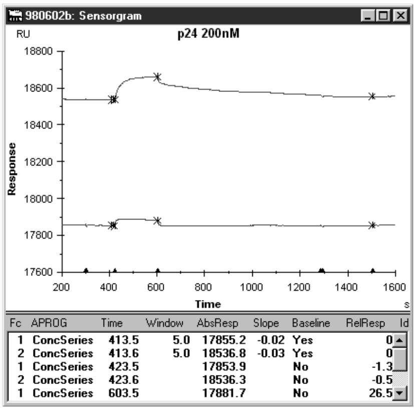
The sensorgram window displays the results of the current run or the contents of a saved result file. Sensorgrams from both sources are examined in the same way. To open a result file, choose **File:Open** and select the required file.

4.8.1 Using multiple windows

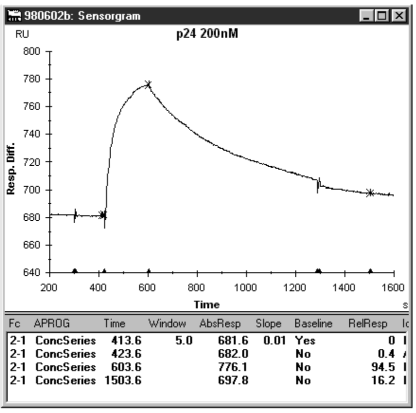
You can display multiple sensorgram windows on the screen to show the sensorgram at different scales, different parts of the same sensorgram or different curves in the same run.

The display properties of separate windows are independent of each other with the exception of cycle number (the cycle can be chosen independently in a saved result file, but all windows show the current cycle during a run).

Choose **Window:New Window** to create a new window. Arrange windows either by dragging the appropriate window borders or choosing one of the **Tile** commands from the **Windows** menu.



Fc	APROG	Time	Window	AbsResp	Slope	Baseline	RelResp	Id
1	ConcSeries	413.5	5.0	17855.2	-0.02	Yes	0	
2	ConcSeries	413.6	5.0	18536.8	-0.03	Yes	0	
1	ConcSeries	423.5		17853.9		No	-1.3	
2	ConcSeries	423.6		18536.3		No	-0.5	
1	ConcSeries	603.5		17881.7		No	26.5	

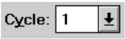


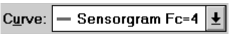
Fc	APROG	Time	Window	AbsResp	Slope	Baseline	RelResp	Ic
2-1	ConcSeries	413.6	5.0	681.6	0.01	Yes	0	I
2-1	ConcSeries	423.6		682.0		No	0.4	A
2-1	ConcSeries	603.6		776.1		No	94.5	I
2-1	ConcSeries	1503.6		697.8		No	16.2	I

Note: Multiple sensorgram windows can only show different views of a single result file. To combine sensorgrams from several result files, open the files in BIAevaluation and combine the separate results in a BIAevaluation project window.

4.8.2 Selecting sensorgrams

A *cycle* is defined as the result of one run, either a manual run or an APROG in a method (see Section E.3.2). A manual run generates one cycle only. A method run may generate any number of cycles. Each cycle contains either one or four sensorgrams (single and multichannel detection modes respectively), and may also contain additional curves (e.g. temperature or buffer volume) if these are requested in the method (see Section E.3.5).


Cycle: 1


Curve: — Sensorgram Fc=4

Choose the cycle to display in the **Cycle** window on the toolbar. The current cycle is always displayed during a method run, but you can change the displayed cycle when the method is complete. Choose the active curve in the **Curve** window on the toolbar. The active curve is the one to which the reference line and baseline functions apply (see Section 4.8.5), and the one for which coordinate values are displayed in the status window.

If you have several curves in the sensorgram window, the **View:Overlay** command will toggle between display of all curves and display of the active curve only.

4.8.3 Viewing curve classes

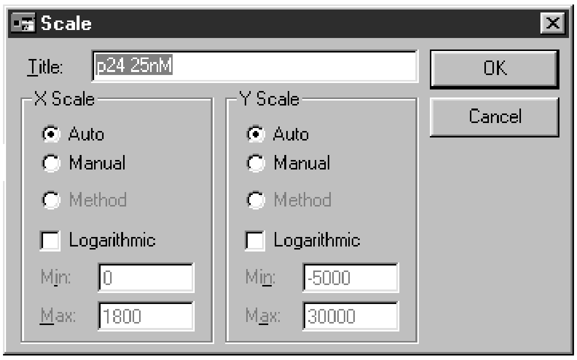
If you have used on-line reference subtraction, selecting the **View:Plot Curve Classes** command will plot all curves in the same class (original or reference-subtracted) as the currently selected curve. This simplifies automatic scaling since the original and reference subtracted curves are usually widely separated in terms of response values.

4.8.4 Scaling the display

To enlarge (zoom) a part of the sensorgram window, drag the cursor over the area to be enlarged (hold down the left mouse button at one corner of the area, move the mouse to the opposite corner, then release the mouse button). If you do this while a sensorgram is running and scaling is set to **Auto**, the scale(s) will be adjusted as soon as the sensorgram goes outside the display.

To revert to the previous display after zooming, select **View:Unzoom** or double-click in the window with the left mouse button. Up to 10 levels of zooming can be restored in this way.

Select **View:Scale** to open a dialogue box where fixed limits can be set for the x- and y-axes. Choose **Auto** to adjust the scales to include the whole sensorgram. Choose **Method** to display the sensorgram according to the scale defined in the method (this option is only available if the method includes a `SCALE` instruction, see Appendix F). Choose **Manual** to set the scale limits manually. You can also choose to display the sensorgram on a logarithmic scale.



Scale

Title: p24 25nM

X Scale
- ● Auto
- ○ Manual
- ○ Method
- □ Logarithmic
- Min: 0
- Max: 1800

Y Scale
- ● Auto
- ○ Manual
- ○ Method
- □ Logarithmic
- Min: -5000
- Max: 30000

Select **View:Adjust Scale** to adjust the scale so that the entire sensorgram is displayed in the window, regardless of any previous fixed or automatic scaling.

4.8.5 Reading coordinate values

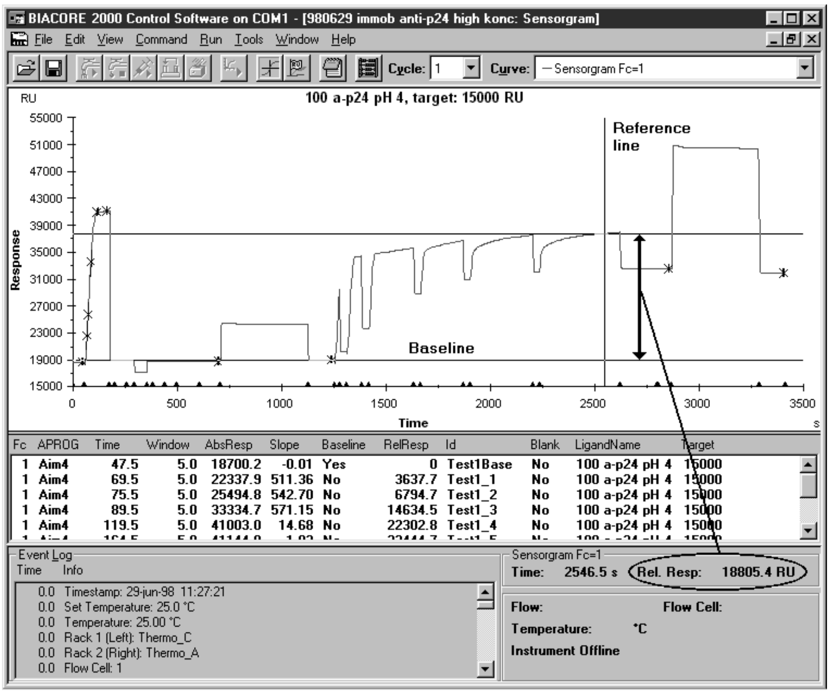
You can read response and time values on the active curve using the reference line and baseline.

Select **View:Reference Line** to display the reference line. You can drag the (vertical) reference line with the cursor: the horizontal reference line follows the y-coordinate of the active curve, and the absolute coordinates of the reference line are displayed in the status window. You can also move the reference line with the arrow keys on the keyboard.

With the reference line displayed, select **View:Baseline** to fix a baseline at the current position of the reference line. This baseline will remain in position as you move the reference line, and the status window will now display the y-coordinate at the reference line relative to the fixed baseline.

Thus to read a relative response value for example at the end of an injection:

1. Choose **View:Reference Line**.
2. Move the reference line to a position before the start of the injection.
3. Choose **View:Baseline**.
4. Move the reference line to the end of the injection and read the relative response from the status window.


BIACORE 2000 Control Software on COM1 - [980629 immob anti-p24 high konc: Sensorgram]
File Edit View Command Run Tools Window Help
Cycle: 1
Curve: — Sensorgram Fc=1
RU
100 a-p24 pH 4, target: 15000 RU
55000
51000
47000
43000
39000
35000
31000
27000
23000
19000
15000
Response
0
500
1000
1500
2000
2500
3000
3500
Time
s
Reference line
Baseline

Fc	APROG	Time	Window	AbsResp	Slope	Baseline	RelResp	Id	Blank	LigandName	Target
1	Aim4	47.5	5.0	18700.2	-0.01	Yes	0	Test1Base	No	100 a-p24 pH 4	15000
1	Aim4	69.5	5.0	22337.9	511.36	No	3637.7	Test1_1	No	100 a-p24 pH 4	15000
1	Aim4	75.5	5.0	25494.8	542.70	No	6794.7	Test1_2	No	100 a-p24 pH 4	15000
1	Aim4	89.5	5.0	33334.7	571.15	No	14634.5	Test1_3	No	100 a-p24 pH 4	15000
1	Aim4	119.5	5.0	41003.0	14.68	No	22302.8	Test1_4	No	100 a-p24 pH 4	15000

Event Log

```
Time   Info
 0.0   Timestamp: 29-jun-98  11:27:21
 0.0   Set Temperature: 25.0 °C
 0.0   Temperature: 25.00 °C
 0.0   Rack 1 (Left): Thermo_C
 0.0   Rack 2 (Right): Thermo_A
 0.0   Flow Cell: 1
```

Sensorgram Fc=1

Time: 2546.5 s Rel. Resp: 18805.4 RU

Flow: Flow Cell:

Temperature: °C

Instrument Offline

To set a new baseline value, choose **View:Baseline** to turn the baseline off and repeat steps 2 and 3 above.

Note: The reference line coordinates shown in the status window are values from single data points. Use report points (see Section 4.8.6) to obtain values averaged over several data points to eliminate noise variations.

4.8.6 Setting report points

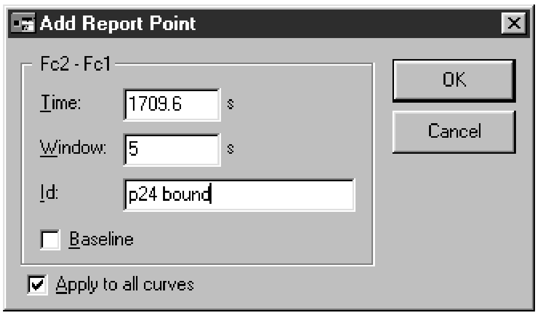
A report point records the time, response and other parameters (defined with the **View:Keywords** command, see Section 9.6) at a specified point in the sensorgram. In multichannel mode, report points can be set either for individual sensorgrams or for all four sensorgrams together. Report points are displayed in the report point table.

You set report points:

- interactively in manual runs, completed method runs and saved sensorgrams with the **Edit:Add Report Point** command or the report point toolbar button.

- in method runs with the `RPOINT` instruction. You cannot set report points interactively during a method run.

The **Edit:Add Report Point** command opens a dialogue box for specifying the report point characteristics:



Fc2 - Fc1

Time: 1709.6 s

Window: 5 s

Id: p24 bound

□ Baseline

☑ Apply to all curves

Enter the time for the report point. If the reference line is displayed (see Section 4.8.5), the position of the reference line is suggested as the time for the report point. Use this feature as an aid in placing report points interactively.

The **Window** parameter determines the window over which the response values will be averaged for the report point. The default value of 5 s is suitable for most purposes. You may want to increase the window if your sensorgram is noisy.

The **Id** parameter appears as an identification of the report point in the report point table. You can enter up to 31 characters in this field.

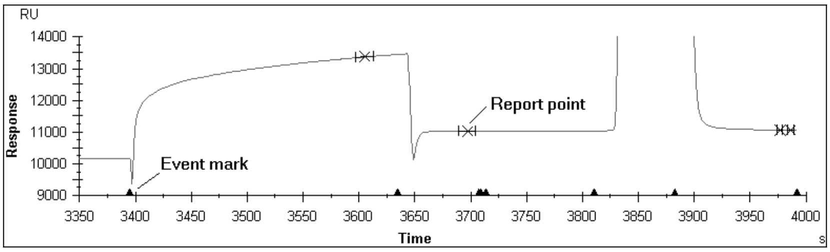
Check the **Baseline** box if you want this report point to be used as a baseline for calculating relative response values in subsequent report points.

Check **Apply to all curves** if you want the report point to be set for all sensorgrams in multichannel mode. If this box is not checked, the report point will be set on the active sensorgram only. This setting is irrelevant in single channel mode.

Editing report points
To change the settings for an existing report point, double-click on the appropriate line in the report point table. This will open the **Edit Report Point** dialogue box, which has the same fields as **Add Report Point** described above. Change the settings as required and click on **OK**.

4.8.7 Report point and event log markers

Report points are marked with an **x** on the curve and an indication of the report point window. Events recorded in the event log (see Section 4.8.8) are marked as small triangles on the x-axis of the sensorgram.



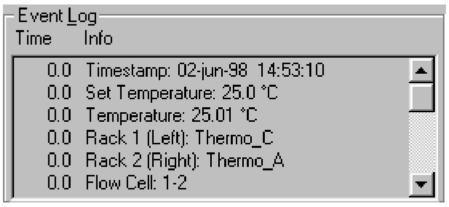
Choose **View:Markers** to toggle the display of report point and event markers on or off.

To identify a report point or event marker, turn on the reference line (see Section 4.8.5) and click on the appropriate line in the report point table or event log respectively. The reference line will jump to the marker for the selected point or event provided that the marker is within the current window.

Note: Event log markers are placed at the exact time of the event, which may not correspond precisely to a data point. The reference line can however only be placed on data points. There may thus be a slight discrepancy between positions of the event marker and reference line. The discrepancy is more pronounced when the data collection rate is set to **Low** (1 point/10 s).

4.8.8 Event log

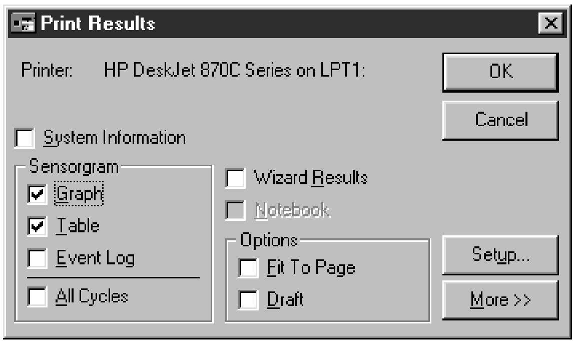
The event log records all events during a run, including a time-stamp at the beginning of the run and the flow cell temperature at the beginning and end of the run. Choose **View:Extended Event Log** to display a full version of the event log including low data quality events. This can be useful in instrument troubleshooting situations.



```
Event Log
Time    Info
   0.0  Timestamp: 02-jun-98  14:53:10
   0.0  Set Temperature: 25.0 °C
   0.0  Temperature: 25.01 °C
   0.0  Rack 1 (Left): Thermo_C
   0.0  Rack 2 (Right): Thermo_A
   0.0  Flow Cell: 1-2
```

4.8.9 Printing sensorgrams

To print sensorgrams and associated data, choose **File:Print**. Select the items to print from the dialogue box.



Print Results

Printer: HP DeskJet 870C Series on LPT1:

□ System Information

Sensorgram
☑ Graph
☑ Table
□ Event Log
□ All Cycles

□ Wizard Results
□ Notebook

Options
□ Fit To Page
□ Draft

Check the **Fit To Page** box to fit each item to a separate page. Click on the **More>>** button for page setup options. For more details, see **File:Print** (Section 9.4).

Note: Sensorgrams and report point tables will be printed as they are displayed on the screen with respect to scaling, overlay and active curve.

If you want to use sensorgrams as illustrations in other Windows programs (e.g. word processing programs), choose **Edit:Copy Graph**. This will copy the current sensorgram window to the Windows clipboard, from where it can be pasted in to other applications.

4.8.10 Evaluating sensorgrams

You can use BIACORE Control Software to display and print sensorgrams and to set report points as described above. For more advanced evaluation, choose **Tools:Evaluation**. This gives access to BIAevaluation software, which provides extensive facilities for evaluating sensorgram data, with model fitting procedures for calculating kinetic constants from the sensorgrams. BIAevaluation also includes functions for sensorgram presentation and evaluation of analyte concentration from rate or binding level measurements. These functions are described separately in the BIAevaluation Software Handbook.

You can also choose to export sensorgram data or report point tables to text files for use in other spreadsheet or data evaluation programs.

5. Application Wizards

Application Wizards help both new and experienced users perform their analyses quickly and confidently. An application wizard is a software tool that guides you through the steps in an experiment, from start to final analysis. When you use an application wizard, you enter all required settings for the run as you step through the dialogue boxes. Operation of the instrument does not start until you click on **Start** in the final dialogue box of the wizard sequence. You are then asked to specify a file name for saving the results, and all the operations required for your analysis are performed automatically in a pre-programmed sequence.

Procedures and recommendations in the application wizards are based on accumulated expertise from BIACORE analyses. Detailed on-line help can be accessed from the dialogue boxes in the wizards.

Settings for wizard-supported analyses can be saved as templates to simplify repeated or similar analyses. The settings are also saved with the result file so that the analysis can be repeated even if no wizard template was saved.

Application wizards complement the already established ways of running BIACORE. Interactive operation through menus and dialogue boxes, command queues for method development and pre-programmed method control using MDL are still fully supported in the software.

Application wizards currently provided with BIACORE Control Software give support for:

Surface preparation

- Immobilization pH scouting: find a suitable immobilization pH based on ligand preconcentration experiments.
- Immobilization of ligand: either specify flow rate and injection time for the immobilization or let the wizard find conditions for reaching a specified target immobilization level.
- Regeneration scouting: find suitable regeneration conditions.
- Surface performance test: determine the analyte binding capacity of the surface and test how well the immobilized ligand withstands regeneration conditions.

Binding analysis

Monitor binding levels in up to four consecutive analyte injections. Results are presented as bar charts.

Kinetic analysis

- Concentration series: determine kinetic parameters from analysis of analyte binding over a range of concentrations. The application wizard evaluates the results according to a simple 1:1 interaction model and presents the apparent kinetic constants automatically.
- Baseline trends: display baseline response as a function of cycle number to help assess surface performance and regeneration during the analysis.
- Mass transfer control experiment: analyse binding as a function of flow rate to identify mass transfer limitations.
- Linked reactions control experiment: analyse dissociation as a function of time at steady state to identify linked reaction schemes.

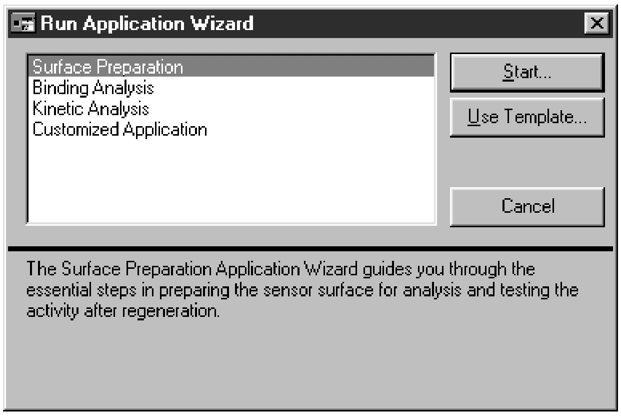
Customized applications

- Program your own command sequences for applications that are not covered by the other application wizards.

5.1 Using application wizards

5.1.1 Starting application wizards

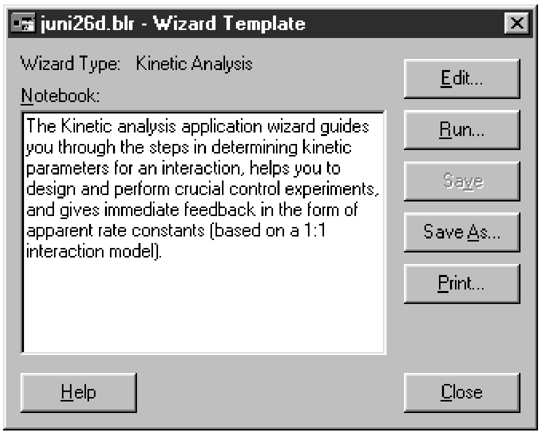
To start an application wizard, choose **Run:Application Wizards** from the main menu. Choose the wizard you want to run and click on **Start**.



The Surface Preparation Application Wizard guides you through the essential steps in preparing the sensor surface for analysis and testing the activity after regeneration.

To use a saved wizard template (see below), click on **Use Template** in the wizard selection dialogue box and select the template file to use. When you use a template, the saved settings for all the wizard dialogue boxes appear as default: you can make any changes you require to the settings as you step through the wizard. Using a template thus provides a convenient way of repeating an analysis either exactly or with minor changes. Note that if you use a template the choice of wizard is determined by the content of the template file: the wizard selected in the dialogue box is irrelevant.

You can also start an application wizard from the result file saved from a wizard-supported run. This will open the appropriate application wizard with settings as used in the result file. To do this, open the result file, choose **View:Wizard Template** and click on **Edit** or **Run** in the dialogue box:



Wizard Type: Kinetic Analysis

Notebook:

The Kinetic analysis application wizard guides you through the steps in determining kinetic parameters for an interaction, helps you to design and perform crucial control experiments, and gives immediate feedback in the form of apparent rate constants (based on a 1:1 interaction model).

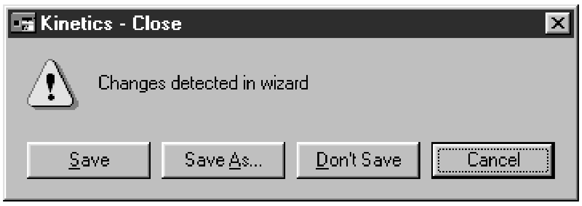
The **Edit** option opens the wizard template at the first screen, so that you can step forwards through the wizard making changes to the settings as required. The **Run** option opens the wizard template at the end of the wizard, so that you simply need to click **Start** in the wizard dialogue in order to repeat the run exactly. You may of course step backwards through the wizard to make any changes you need.

When you close a wizard where you have entered sample details, you are given the option of saving the wizard template for future use.

5.1.2 Wizard templates

The settings used in an application wizard can be saved in a template for later use. This makes it easy to repeat a wizard-based analysis.

If you close a wizard after having made changes to the template, you will be given an opportunity to save it:



Changes detected in wizard

You can also save the template from the **Menu** option that is available in some wizard dialogue boxes (see Section 5.5). Wizard templates are saved by default with file type .blw in the **BIA Users\Templates** folder.

To create a wizard template and save it for later use, you can also choose **New:Application Wizard** from the **File** menu. This will allow you to select a wizard and step through it, then save the template when all parameters are defined.

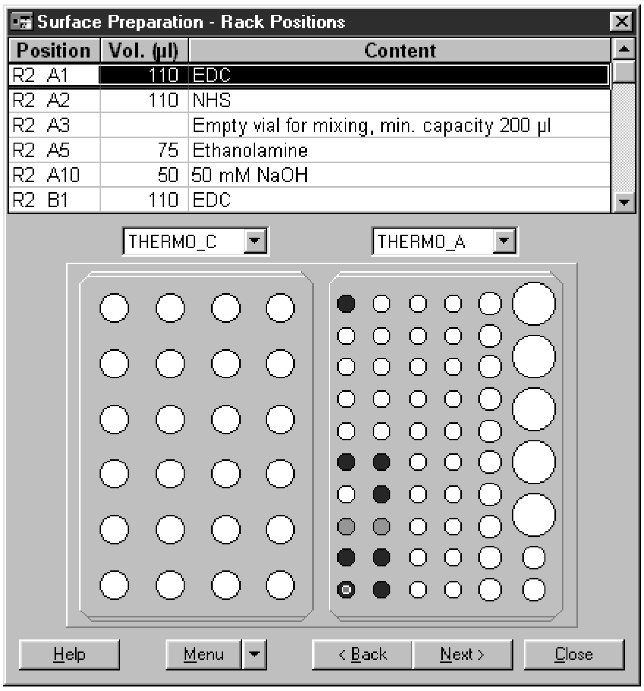
A saved wizard template contains all settings up to and including the point at which it was saved, and these settings are presented as default when the template is used.

5.1.3 Common wizard functions

The functions below are common to most application wizards:

Autosampler rack positions

The wizard suggests autosampler rack positions for samples, reagents and empty vials (if necessary) based on the requirements of your application. You can change the placing of vials in the racks simply by dragging the appropriate filled vial position. You cannot move a vial to a position which does not have sufficient volume capacity.



Surface Preparation - Rack Positions

Position	Vol. (µl)	Content
R2 A1	110	EDC
R2 A2	110	NHS
R2 A3		Empty vial for mixing, min. capacity 200 µl
R2 A5	75	Ethanolamine
R2 A10	50	50 mM NaOH
R2 B1	110	EDC

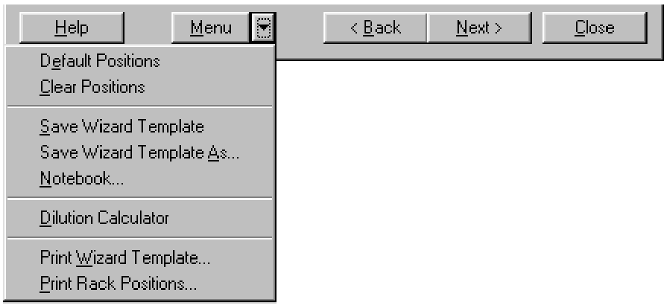
THERMO_C

THERMO_A

Note that the volumes listed in the table at the top of the dialogue box may change if you change vial positions. The wizard presents minimum volumes required for standard vials for each rack type and position (as listed in the Biacore Product Catalogue).

You can also change the rack type(s) from the pull-down lists above the rack illustrations. If you change a rack type, all positions are cleared. To place samples in the new rack, either click on a rack positions to place the sample that is highlighted in the table (the highlight moves automatically through the table as you do this), or choose **Default Positions** from the menu button at the bottom of the dialogue box. This option will assign vials to their default positions in the new rack.

Menu options
Some of the application wizard dialogue boxes have an additional **Menu** button for auxiliary functions:



Default Positions
Places samples in their default positions in the currently selected rack(s). Only available in the **Rack Positions** dialogue box.

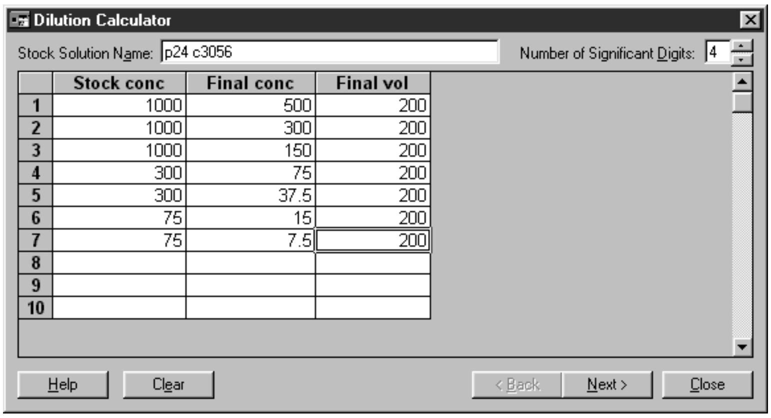
Clear Positions
Clears all sample positions from the racks. You must place all samples in positions in the racks before you can continue. Only available in the **Rack Positions** dialogue box.

Save Wizard Template (As)
Saves the current wizard settings in a wizard template file.

Notebook
Opens the run notebook where you can enter comments on the wizard and/or the run. See Section 9.6 for a description of the run notebook. Only comments in a section identified by a **Wizard Notes** header will be saved with the template: all comments will be saved with the result file.

Dilution Calculator

Opens a utility for calculating volumes required for preparing a dilution series. To use the dilution calculator, enter your stock solution concentrations, final concentrations and final required volumes in the first screen. You can also enter a stock solution name if desired:

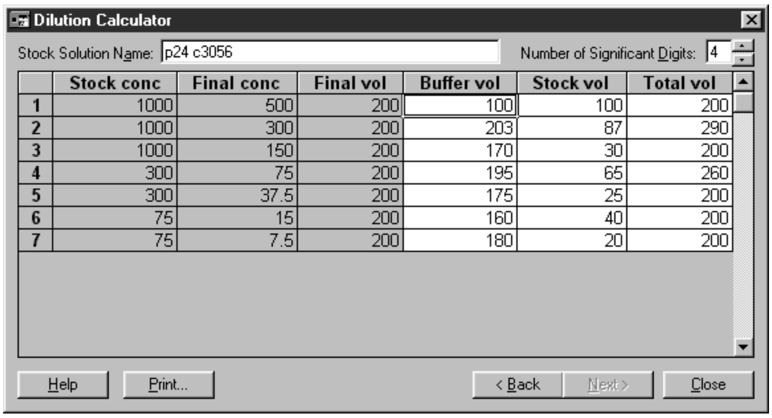


Stock Solution Name: p24 c3056

Number of Significant Digits: 4

	Stock conc	Final conc	Final vol
1	1000	500	200
2	1000	300	200
3	1000	150	200
4	300	75	200
5	300	37.5	200
6	75	15	200
7	75	7.5	200
8			
9			
10			

To calculate required volumes, click on **Next**:



Stock Solution Name: p24 c3056

Number of Significant Digits: 4

	Stock conc	Final conc	Final vol	Buffer vol	Stock vol	Total vol
1	1000	500	200	100	100	200
2	1000	300	200	203	87	290
3	1000	150	200	170	30	200
4	300	75	200	195	65	260
5	300	37.5	200	175	25	200
6	75	15	200	160	40	200
7	75	7.5	200	180	20	200

In this second screen you can adjust the calculated volumes to more convenient values simply by typing new values where required. All related values are recalculated every time you change a value. For example, you may wish to adjust the stock volume in row 2 in this example to 90: the buffer and total volumes are adjusted automatically to 210 and 300 respectively. You will be alerted if you enter a volume that is not sufficient for your requirements.

No units are specified for either concentration or volume: it is assumed that all values are entered in the same units.

Note that the total calculated volume is the sum of your final volume and any additional amount that is needed for further dilutions (in the example, the final concentration 300 is used as stock for two additional concentrations, and requires a total volume of 200+90).

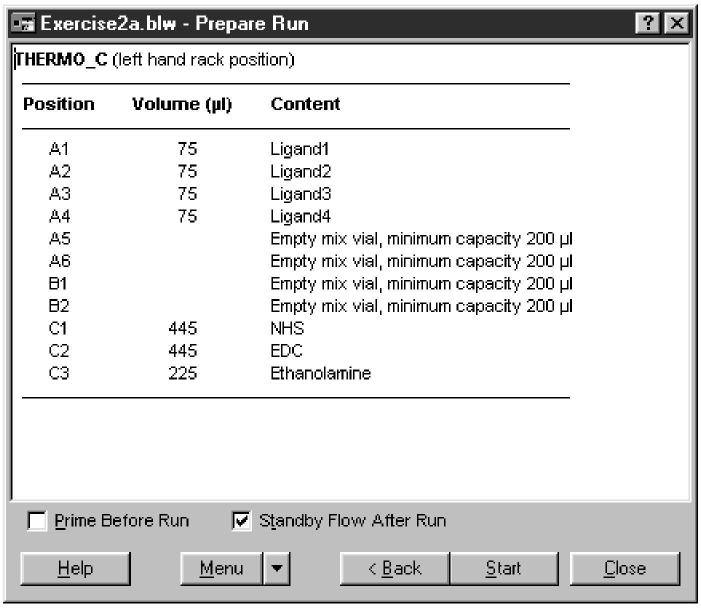
Click on **Print** to print the dilution protocol.

Print Wizard Template
Prints all current settings for the wizard.

Print Rack Positions
Prints a summary of the samples and rack positions required for the analysis.

Prepare Run summary

The last dialogue box before starting a wizard-controlled run summarizes the samples required for the run and their positions in the autosampler. Take time to check through this summary to make sure that all samples are in place before starting the run.



THERMO_C (left hand rack position)

Position	Volume (µl)	Content
A1	75	Ligand1
A2	75	Ligand2
A3	75	Ligand3
A4	75	Ligand4
A5		Empty mix vial, minimum capacity 200 µl
A6		Empty mix vial, minimum capacity 200 µl
B1		Empty mix vial, minimum capacity 200 µl
B2		Empty mix vial, minimum capacity 200 µl
C1	445	NHS
C2	445	EDC
C3	225	Ethanolamine

☐ Prime Before Run ☑ Standby Flow After Run

Here you can check the **Prime Before Run** box to prime the system (see Section 7.2) before the run starts, and the **Standby Flow After Run** box to start the **Standby** working tool (see Section 7.2) when the run is completed.

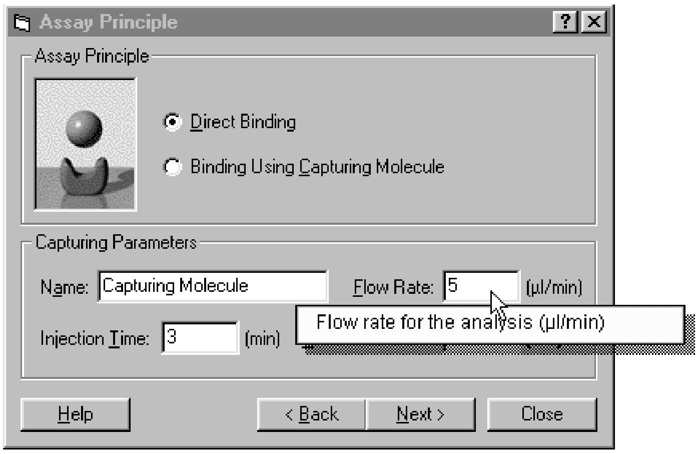
5.2 Getting help on wizards

The application wizards are provided with extensive on-line help, accessed either from the **Help** option in the main menu or directly from the wizard dialogue boxes. This chapter does not document all features of the application wizards: use the on-line help to learn more details of how to use the wizards.

To open help from the menu, choose **Help:Help on Application Wizards**. Navigate through the help screens with standard Windows techniques.

The **Help** button in each wizard dialogue box opens the help at the screen which applies to the box in question.

To get help on a particular setting in a wizard dialogue box, click on the question mark at the right hand end of the dialogue box title and then click on the setting. Help is displayed in a pop-up window.



● Direct Binding

○ Binding Using Capturing Molecule

Name: Capturing Molecule

Flow Rate: 5 (µl/min)

Injection Time: 3 (min)

Flow rate for the analysis (µl/min)

5.3 The surface preparation wizard

5.3.1 Surface preparation wizard functions

The surface preparation wizard provides support for:

- Scouting for a suitable immobilization pH.
- Immobilizing ligand using defined immobilization conditions or to a pre-set target level.
- Preparing reference surfaces for in-line reference subtraction.
- Finding suitable regeneration conditions.
- Testing analyte binding capacity and regeneration conditions.

A flow diagram for the surface preparation wizard is shown in Figure 1.

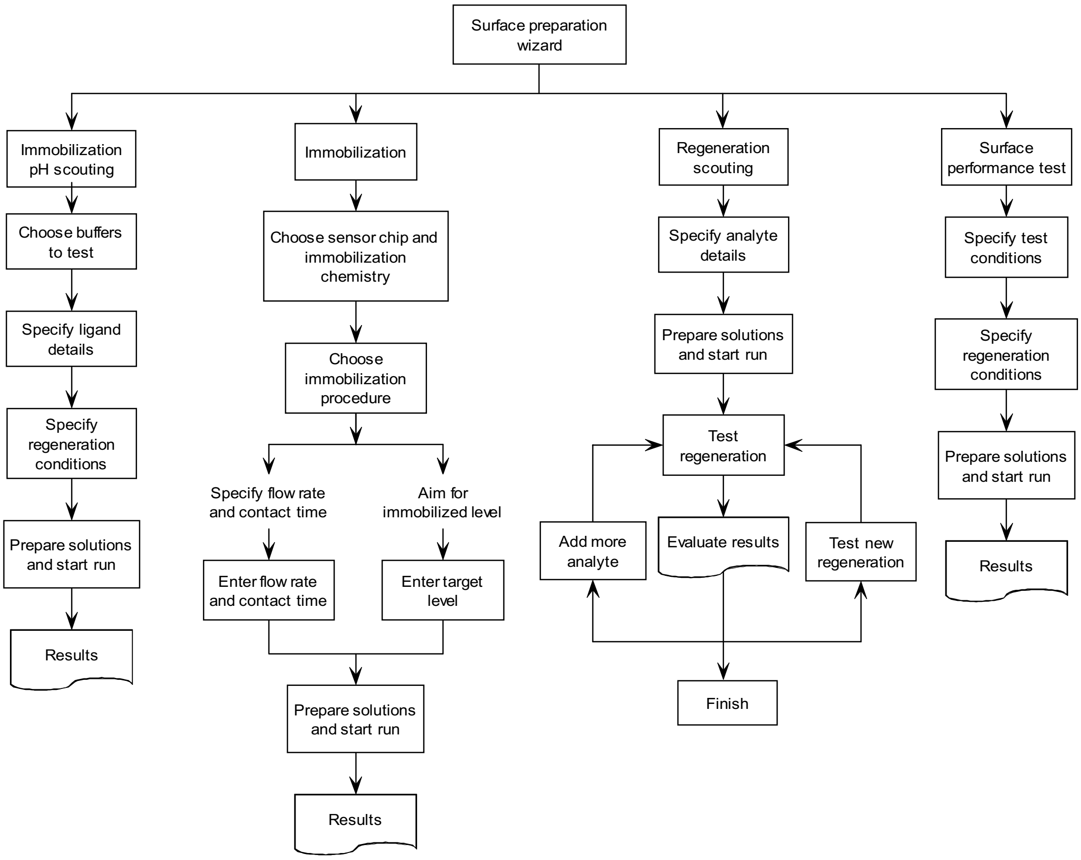
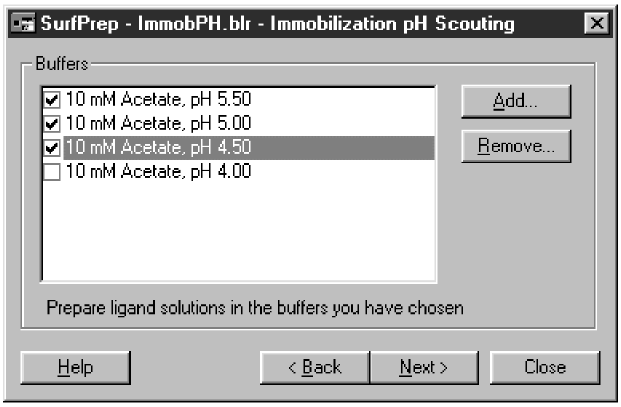

Surface preparation wizard
Immobilization pH scouting
Choose buffers to test
Specify ligand details
Specify regeneration conditions
Prepare solutions and start run
Results
Immobilization
Choose sensor chip and immobilization chemistry
Choose immobilization procedure
Specify flow rate and contact time
Aim for immobilized level
Enter flow rate and contact time
Enter target level
Prepare solutions and start run
Results
Regeneration scouting
Specify analyte details
Prepare solutions and start run
Test regeneration
Add more analyte
Evaluate results
Test new regeneration
Finish
Surface performance test
Specify test conditions
Specify regeneration conditions
Prepare solutions and start run
Results

Figure 1. *Flow chart for the surface preparation wizard.*

5.3.2 Immobilization pH scouting

The **Immobilization pH Scouting** wizard component tests ligand preconcentration in selected buffers. The sensor chip surface is not activated, and may be used for immobilization when suitable buffer conditions have been established. Buffers listed include immobilization buffers supplied by Biacore AB: these cannot be deleted from the list. You can use the **Add** and **Remove** buttons to add and remove your own buffers.

Note: Immobilization pH scouting applies only to CM-series sensor chips.

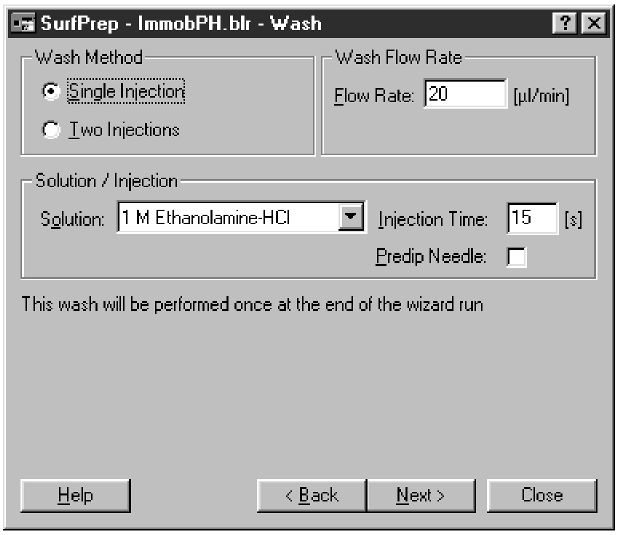

SurfPrep - ImmobPH.blr - Immobilization pH Scouting

Buffers

☑ 10 mM Acetate, pH 5.50
☑ 10 mM Acetate, pH 5.00
☑ 10 mM Acetate, pH 4.50
□ 10 mM Acetate, pH 4.00

Prepare ligand solutions in the buffers you have chosen

The sensor surface is washed with running buffer between ligand injections, which is normally sufficient to remove ligand bound during the preconcentration step. To ensure that the surface is properly cleaned after completion of the immobilization pH scouting, one or two washes with user-defined solutions are performed at the end of the wizard sequence:


SurfPrep - ImmobPH.blr - Wash

Wash Method

● Single Injection
○ Two Injections

Wash Flow Rate

Flow Rate: 20 [µl/min]

Solution / Injection

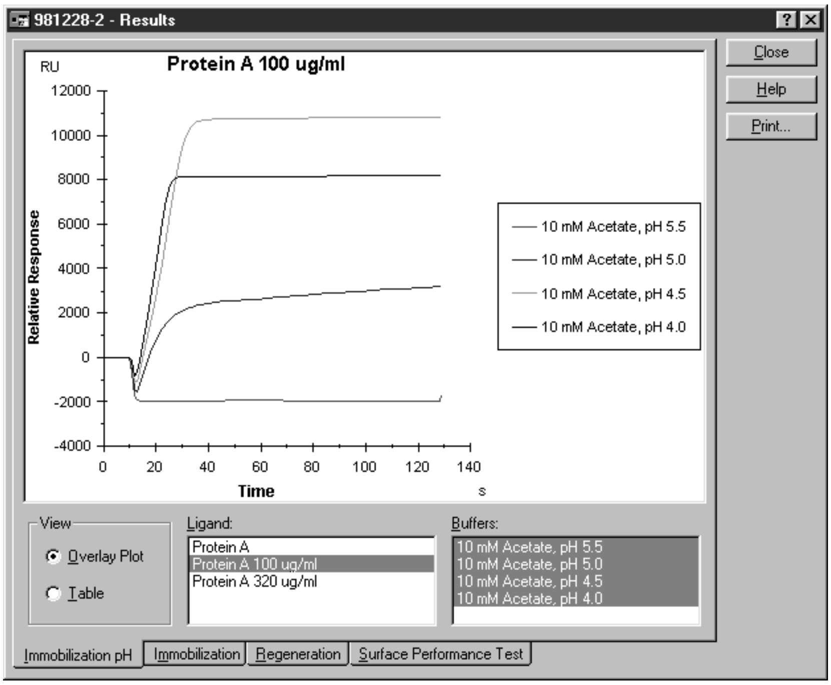
Solution: 1 M Ethanolamine-HCl

Injection Time: 15 [s]

Predip Needle: □

This wash will be performed once at the end of the wizard run

Results are presented either as an overlay plot of the preconcentration sensorgrams or as a table of response levels just before the end of the ligand injection.



View: ● Overlay Plot ○ Table

Ligand:
Protein A
Protein A 100 ug/ml
Protein A 320 ug/ml

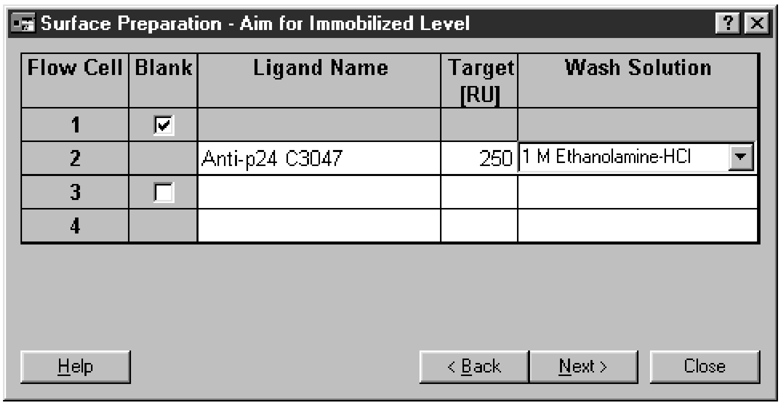
Buffers:
10 mM Acetate, pH 5.5
10 mM Acetate, pH 5.0
10 mM Acetate, pH 4.5
10 mM Acetate, pH 4.0

The results of immobilization pH scouting are intended as guidelines for choice of immobilization conditions, and should be interpreted according to the requirements of the experiment at hand. Typically, amine coupling will result in immobilization levels around 60-80% of the preconcentration levels, although this value varies between individual ligands according to the reactivity of coupling groups on the ligand and the pH at which immobilization is performed.

5.3.3 Immobilization

Choice of sensor chip and immobilization chemistry
Immobilization on Sensor Chip CM5 using amine coupling chemistry and on Sensor Chip SA using direct binding of biotinylated ligand are currently supported.

Preparing in-line reference cells
Blank surfaces for in-line reference cells may be prepared by checking the appropriate box in the dialogue box:



Surface Preparation - Aim for Immobilized Level

Flow Cell	Blank	Ligand Name	Target [RU]	Wash Solution
1	☑			
2		Anti-p24 C3047	250	1 M Ethanolamine-HCl
3	☐			
4				

Flow cell 1 can be used as an in-line reference for flow cell 2, and flow cell 3 for flow cell 4. In this example flow cell 1 is prepared as a reference for flow cell 2, while flow cells 3 and 4 are not used.

Blank surfaces prepared in this way are activated with NHS/EDC and deactivated with ethanolamine, without immobilizing ligand. ***For careful work, particularly applications using low immobilized levels and low response levels, blank surfaces should preferably be prepared by immobilizing an equivalent amount of an irrelevant protein (e.g. non-binding antibody).*** To prepare this kind of blank surface, immobilize the irrelevant protein in the reference cell in a separate surface preparation step: do not check the reference cell box.

Aiming for immobilized ligand level
This option is only available for immobilization on Sensor Chip CM5.

When you use the **Aim for Immobilized Level** option, the wizard programs the following operations:

1. Inject a short pulse of ligand solution without activating the surface, to determine the rate of binding of ligand to the surface.

2. Inject a pulse of wash solution (e.g. 1 M ethanolamine-HCl) to remove bound material from the surface. You can specify different wash solutions for each flow channel.

3. If the binding rate as determined in step 1 is suitable (i.e. the target level can be reached within a reasonable time frame), continue with step 4. Otherwise abort the procedure with an appropriate message.

4. Activate the surface with NHS/EDC.

5. Inject ligand in pulses using a contact time based on the binding rate estimated in step 1 so that the target level is expected to be

reached after a few pulses. The injection procedure is stopped when the target level is reached or if:

- the sample loop is empty
- the target level is not reached within 20 injection pulses
- the response level decreases during an injection pulse.

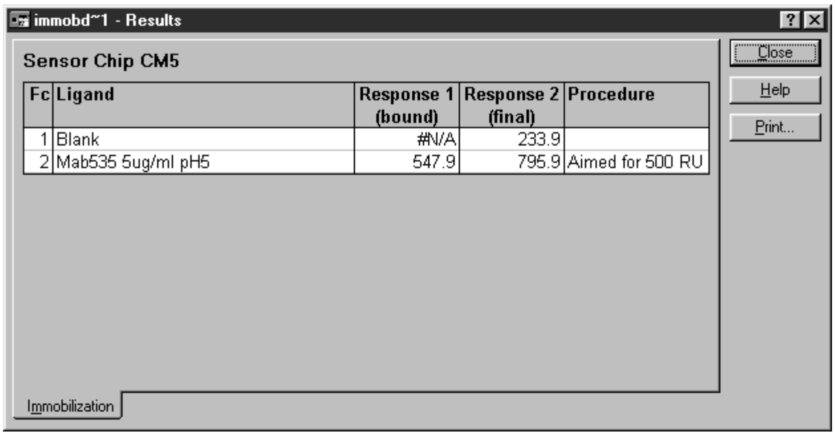
6. Inject ethanolamine to complete the immobilization procedure.

7. Display the results.

For this procedure to be successful, it is important that the rate of ligand binding to the surface is within the range that will allow the target level to be reached. If the binding level overshoots the target in the first injection pulse, reduce the ligand concentration and/or use immobilization buffer with a pH that gives less than the maximum binding rate. As a rule of thumb, use ligand concentrations in the range 5-50 µg/ml for high immobilization levels and 1-20 µg/ml for low levels. Use lower concentrations if the analyte molecular weight is high.

Presentation of results

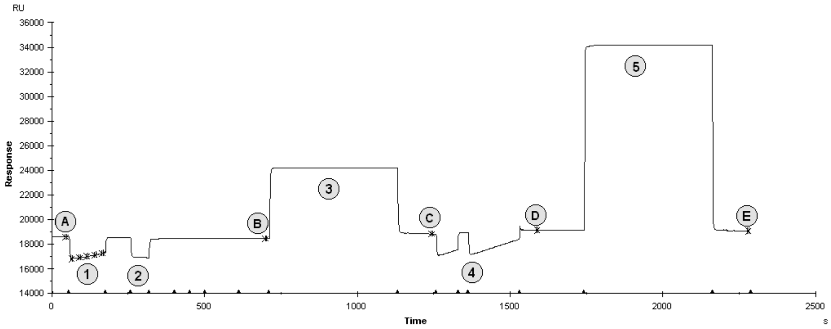
Results from immobilization are presented as a table of response levels. Note that the immobilization results are also shown as a sensorgram in the regular sensorgram window.



Sensor Chip CM5

Fc	Ligand	Response 1 (bound)	Response 2 (final)	Procedure
1	Blank	#N/A	233.9	
2	Mab535 5ug/ml pH5	547.9	795.9	Aimed for 500 RU

A typical sensorgram from immobilization using **Aim for Immobilized Level** is illustrated below:



Steps in the immobilization procedure are as follows:

1. Injection of ligand to determine binding rate (only for **Aim for Immobilized Level**).
2. Short pulse of wash solution to remove any bound material remaining from step 1 (only for **Aim for Immobilized Level**).
3. Activation with NHS/EDC.
4. Injection of ligand (may be several pulses in **Aim for Immobilized Level**).
5. Injection of ethanolamine.

Report points are set as follows for calculating the amount of immobilized ligand:

A. Baseline before start of immobilization procedure (only for **Aim for Immobilized Level**).
B. Baseline before activation with NHS/EDC.
C. Response level after activation with NHS/EDC.
D. Response level after injection of ligand.
E. Response level after deactivation with ethanolamine.

The response levels reported in the wizard results are as follows:

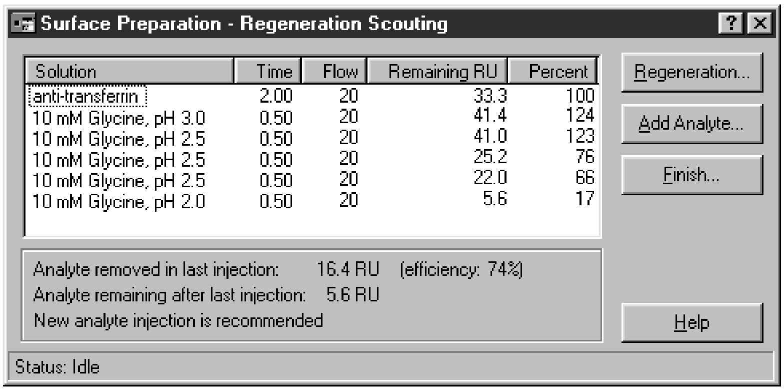
Response 1 (bound)	Difference between report points D and C
Response 2 (final)	Difference between report points E and B

In **Aim for Immobilized Level**, the assessment of the level reached is made on the basis of response 1, which represents the amount of ligand bound to the surface after ligand injection. Note that this will probably differ from the amount of ligand actually immobilized on the surface. In general, high immobilization levels are more correctly represented by response 2, while low levels are closer to response 1.

5.3.4 Regeneration scouting

Running regeneration scouting
The regeneration scouting wizard component helps you to find suitable regeneration conditions for your ligand. The first dialogue box asks for conditions for the first analyte injection, and leads to the rack

editor and run protocol dialogues. Once the run is started, the wizard provides a summary of the results for each injection and the opportunity to inject regeneration solution or new analyte or to finish the scouting run. When appropriate, the dialogue box also indicates that a new injection of analyte is recommended:

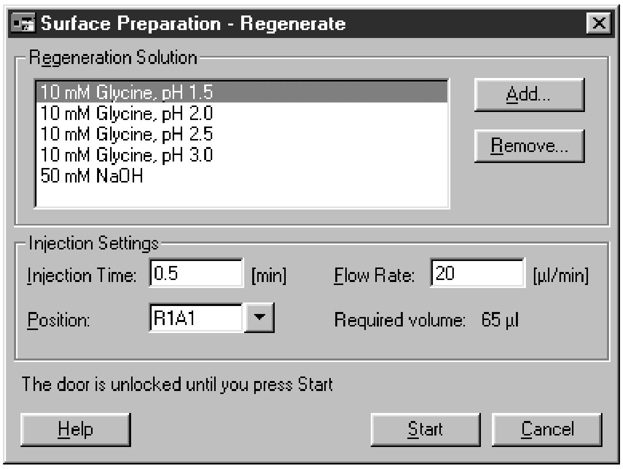


Solution	Time	Flow	Remaining RU	Percent
anti-transferrin	2.00	20	33.3	100
10 mM Glycine, pH 3.0	0.50	20	41.4	124
10 mM Glycine, pH 2.5	0.50	20	41.0	123
10 mM Glycine, pH 2.5	0.50	20	25.2	76
10 mM Glycine, pH 2.5	0.50	20	22.0	66
10 mM Glycine, pH 2.0	0.50	20	5.6	17

Analyte removed in last injection: 16.4 RU (efficiency: 74%)
Analyte remaining after last injection: 5.6 RU
New analyte injection is recommended

Status: Idle

You determine the injection conditions for each new injection of regeneration solution or analyte:



Regeneration Solution
- 10 mM Glycine, pH 1.5
- 10 mM Glycine, pH 2.0
- 10 mM Glycine, pH 2.5
- 10 mM Glycine, pH 3.0
- 50 mM NaOH

Injection Settings

Injection Time: 0.5 [min]   Flow Rate: 20 [µl/min]

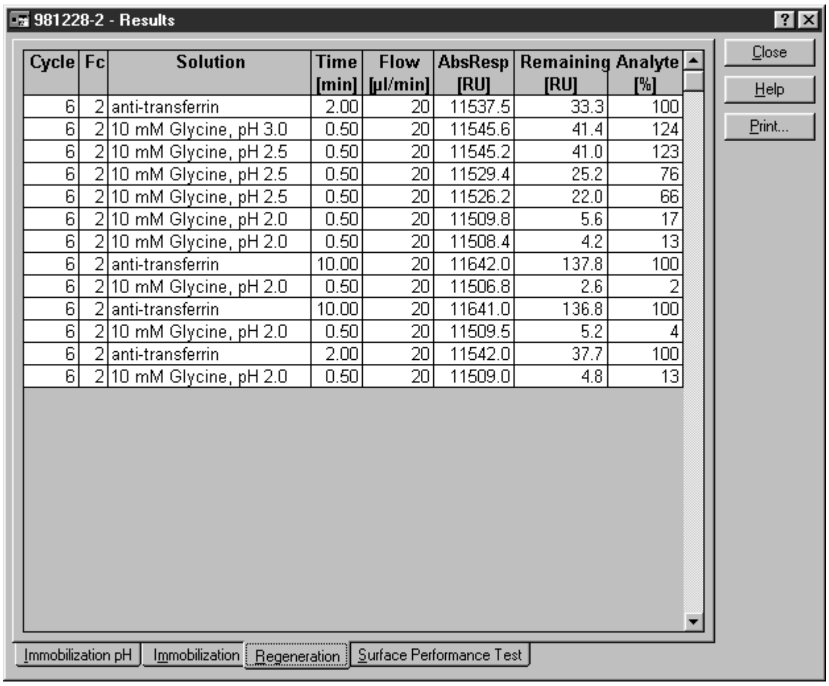
Position: R1A1   Required volume: 65 µl

The door is unlocked until you press Start

Click on the pull-down marker beside the **Position** field to open a rack display where you can select the position from which to inject solution. Note that there is no automatic support for checking that a vial is present in the selected position, or for monitoring the volume of solution remaining in the vial. However, the amount of solution required for the current injection is displayed beside the **Position** field.

Presentation of results

The results of regeneration scouting are presented as a table of analyte binding levels for each injection:



Cycle	Fc	Solution	Time [min]	Flow [µl/min]	AbsResp [RU]	Remaining [RU]	Analyte [%]
6	2	anti-transferrin	2.00	20	11537.5	33.3	100
6	2	10 mM Glycine, pH 3.0	0.50	20	11545.6	41.4	124
6	2	10 mM Glycine, pH 2.5	0.50	20	11545.2	41.0	123
6	2	10 mM Glycine, pH 2.5	0.50	20	11529.4	25.2	76
6	2	10 mM Glycine, pH 2.5	0.50	20	11526.2	22.0	66
6	2	10 mM Glycine, pH 2.0	0.50	20	11509.8	5.6	17
6	2	10 mM Glycine, pH 2.0	0.50	20	11508.4	4.2	13
6	2	anti-transferrin	10.00	20	11642.0	137.8	100
6	2	10 mM Glycine, pH 2.0	0.50	20	11506.8	2.6	2
6	2	anti-transferrin	10.00	20	11641.0	136.8	100
6	2	10 mM Glycine, pH 2.0	0.50	20	11509.5	5.2	4
6	2	anti-transferrin	2.00	20	11542.0	37.7	100
6	2	10 mM Glycine, pH 2.0	0.50	20	11509.0	4.8	13

The table shows the analyte remaining on the surface after each injection in terms of RU and percentage. The percentage is reset to 100% each time new analyte is injected. The baseline level (0%) is initially set to the response before the first analyte injection, and is set to a new level only if the response before an analyte injection is lower than the previous baseline (i.e. the **Remaining Analyte** value is negative).

Regeneration scouting in practice

The recommendations below can help you to get the best results from the regeneration scouting wizard:

- Use purified analyte if possible, to minimise complications from non-specific binding of sample components to the surface.

- Inject analyte at the highest concentration you intend to use in analyses. Assessment of regeneration levels is more reliable with higher levels of bound analyte.

- Prepare the analyte in the buffer you intend to use for analysis. In particular, avoid low ionic strength buffers that can enhance non-specific binding of analyte to the sensor surface.

- Test mild regeneration conditions first, and increase either contact time or the harshness of the conditions (e.g. more extreme pH, higher ionic strength etc) if the first trials are not adequate.

The scouting procedure should be regarded as a rough approach to finding regeneration conditions. Do not attempt to fine-tune regeneration using the scouting procedure: accumulation of analyte through repeated cycles of incomplete regeneration and injection of new analyte can make detailed interpretation difficult. Instead, once suggested conditions have been found using the scouting procedure, test and fine-tune them on a newly-prepared surface if possible.

When suitable regeneration conditions have been found, run the **Surface Performance Test** wizard to check the performance of the immobilized ligand through repeated cycles of binding and regeneration.

5.3.5 Surface performance test

This wizard tests surface performance through repeated cycles of analyte binding and regeneration. The performance is assessed in terms of completeness of regeneration and of retention of analyte binding capacity.

Analyte and run settings

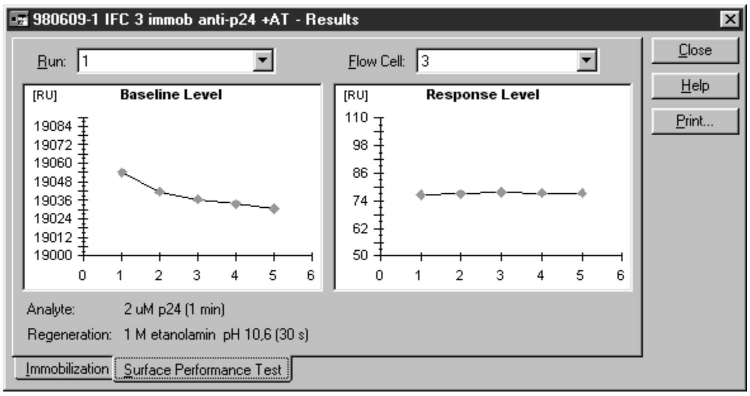
For the surface performance test, use binding conditions that are as close as possible to your analysis conditions. Run several cycles (at least 3, recommended 5) so that any trends in binding and regeneration levels are clear.

Regeneration conditions

Before running the surface performance test, regeneration conditions must be established. The wizard supports regeneration with one or two pulses of the same or different regeneration solutions as well as regeneration by dissociation in buffer (for interactions with rapid dissociation rates).

Presentation of results

The results from the surface performance test are shown as trend diagrams for level of analyte binding and baseline level after regeneration. Ideally, both values should be constant throughout the test cycles. Inadequate regeneration appears as an upward trend in the baseline level, while loss of surface binding capacity gives a downward trend in the response level. A drop in the baseline is normally not a problem provided that the response level remains constant (as in the illustration below).



980609-1 IFC 3 immob anti-p24 +AT - Results

Run: 1

Flow Cell: 3

Analyte: 2 uM p24 (1 min)

Regeneration: 1 M etanolamin pH 10,6 (30 s)

5.3.6 Troubleshooting surface preparation

The **Aim for Immobilized Level** procedure will attempt to achieve the target immobilized level using the ligand solution you provide. If the procedure determines that the immobilization is not feasible, the surface will not be activated and may be used again.

Assessment of whether the immobilization can be carried out is made on the basis of the rate of binding during an injection over the unactivated surface (step 1 in the illustration in Section 5.3.3). If immobilization is judged to be feasible, the surface is activated and ligand is injected. The amount bound during ligand injection is used as an estimate of the amount that is actually immobilized: if ligand binds without being covalently linked, this estimate may be misleading.

There are thus a number of situations where **Aim for Immobilized Level** may give unsatisfactory results. Suggested corrective measures are listed below. If none of these measures works, you may need to determine suitable conditions in independent pilot experiments and then use the **Specify Flow Rate and Injection Time** alternative in the surface preparation wizard.

In determining whether a ligand immobilization is satisfactory, remember that the level of immobilized ligand does not necessarily reflect the functional capacity of the surface. The important criterion is the analyte binding capacity (which can be estimated using the surface performance test wizard).

- *Immobilization not performed: ligand binding too fast*
  This situation arises if the shortest injection time for ligand is expected to overshoot the target level.

  Try using a lower concentration of ligand. If this does not help, increase the pH of the immobilization buffer by 0.5 - 1 units. (If the pH used is below optimum for electrostatic preconcentration, this measure may actually increase the binding rate. If this behaviour is seen, increase the pH further until the binding rate drops.)

- ***Immobilization not performed: ligand binding too slow***
  This situation arises if ligand binding to the surface could not be detected in a test injection or is too slow to reach the target level within the maximum contact time.

  Check the ligand solution:

  - The pH should be 0.5 - 1 units below the isoelectric point of the ligand (or at a value determined to be optimum for electrostatic preconcentration of ligand on the surface).

  - The ionic strength should be 10 mM or lower (beware of salt introduced by diluting a high ionic strength ligand stock solution into coupling buffer).

  - The ligand concentration should typically be 5-50 µg/ml for high immobilization levels or 2-20 µg/ml for low levels. Higher concentrations may be required in cases where the preconcentration efficiency is low.

- ***Immobilization performed but target level not reached***
  This situation can arise if ligand binds to the surface during injection but is removed by the pulse of ethanolamine (i.e. is not covalently linked), or if the immobilization process is terminated before the target level has been reached.

  Make sure you use fresh NHS and EDC solutions.

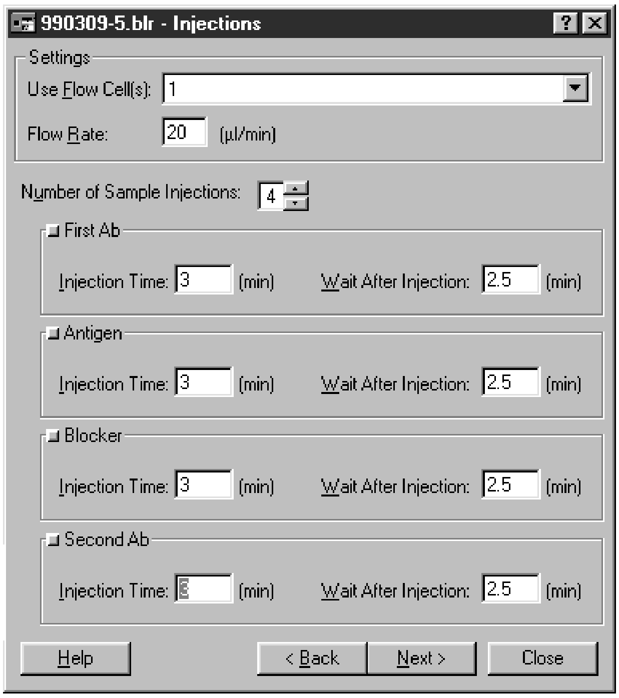
  The amount of ligand solution that can be injected is limited by the capacity of the sample loop. If this is not sufficient to achieve the target level, try optimizing the immobilization pH or increasing the ligand concentration.

  The ligand may bind electrostatically to the surface (Response 1 acceptable) but not be covalently linked (Response 2 low) in spite of fresh NHS and EDC solutions. Alternative immobilization chemistry or capturing may be more appropriate.

- ***Immobilization performed but target level exceeded***
  This situation can arise if immobilization of ligand proceeds significantly faster than the electrostatic binding used to estimate the required injection time. In a case like this, **Aim for Immobilized Level** is not suitable: use **Specify Flow Rate and Injection Time** instead.

5.4 The binding analysis wizard

The binding analysis wizard provides support for determining the extent of binding of up to four consecutive analytes to an immobilized ligand, in multiple cycles that can use the same or different analytes. The results are presented as bar charts of binding levels. This wizard is intended in the first place for yes/no binding tests and rough assessment of the extent of binding (which may provide ranking information for binding affinities or information on relative concentrations, depending on how the analyte samples are prepared).



990309-5.blr - Injections

Settings

Use Flow Cell(s): 1

Flow Rate: 20 (µl/min)

Number of Sample Injections: 4

First Ab

Injection Time: 3 (min) Wait After Injection: 2.5 (min)

Antigen

Injection Time: 3 (min) Wait After Injection: 2.5 (min)

Blocker

Injection Time: 3 (min) Wait After Injection: 2.5 (min)

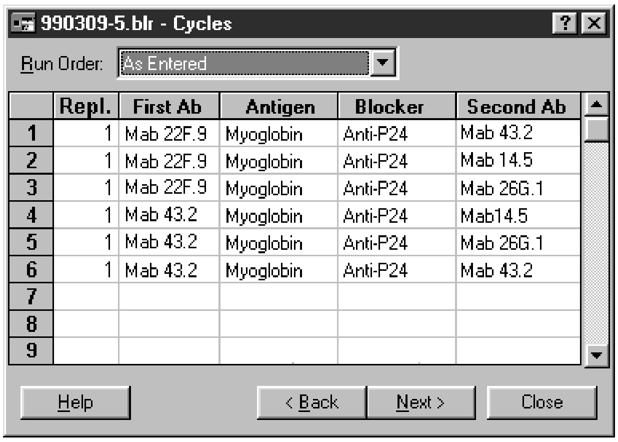
Second Ab

Injection Time: 3 (min) Wait After Injection: 2.5 (min)

Choose the number of injections and the conditions for each injection in the **Injections** dialogue box. You can set the name of the analyte(s) by clicking on the small square at the top left of the panel for each analyte (the names in the example above are set to **First Ab**, **Antigen**, **Blocker** and **Second Ab**, as might be used for example in an epitope mapping experiment).

Analyte injections are performed in direct succession, with no regeneration between samples. In practice, there is a minimum delay of about 2.5 minutes between injections, corresponding to the autosampler preparation time for INJECT commands (see Section 4.4.6). The **Wait After Injection** box allows you to specify delay periods longer than 2.5 minutes.

The injection conditions specified in the **Injections** dialogue box apply to all cycles in the binding analysis run. Details of the analytes for each cycle are specified in the next dialogue box:



990309-5.blr - Cycles

Run Order: As Entered

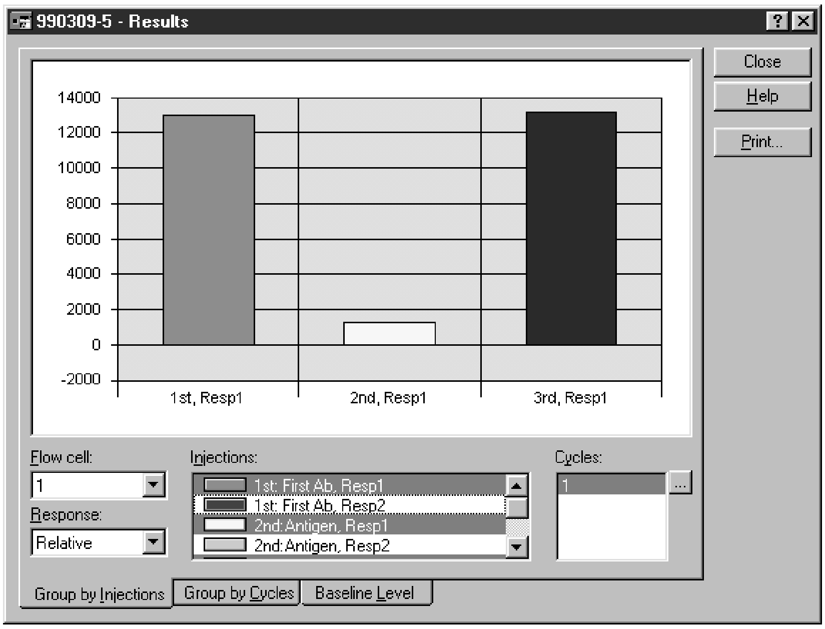
	Repl.	First Ab	Antigen	Blocker	Second Ab
1	1	Mab 22F.9	Myoglobin	Anti-P24	Mab 43.2
2	1	Mab 22F.9	Myoglobin	Anti-P24	Mab 14.5
3	1	Mab 22F.9	Myoglobin	Anti-P24	Mab 26G.1
4	1	Mab 43.2	Myoglobin	Anti-P24	Mab14.5
5	1	Mab 43.2	Myoglobin	Anti-P24	Mab 26G.1
6	1	Mab 43.2	Myoglobin	Anti-P24	Mab 43.2
7					
8					
9					

Each line in the **Cycles** table represents one analysis cycle (with the number of replicates specified in the **Repl** column). Different cycles may use the same or different analytes.

Regeneration is performed after the last injection in each cycle.

Presentation of results

The results of binding analysis are presented as bar charts of response levels, grouped either by analyte or by cycle.



990309-5 - Results

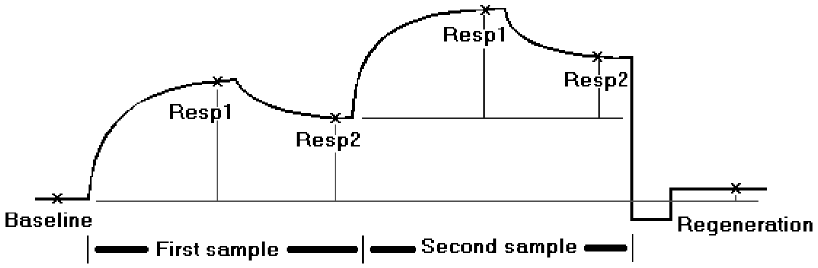
Flow cell: 1

Response: Relative

Injections:
- 1st: First Ab, Resp1
- 1st: First Ab, Resp2
- 2nd:Antigen, Resp1
- 2nd:Antigen, Resp2

Cycles: 1

Response levels are recorded at the following preset points in each injection (shown for two injections in the illustration):



- **Baseline**: shortly after the beginning of the cycle, before the first injection

- **Resp1**: shortly before the end of the injection (representing the amount of analyte bound during injection)

- **Resp2** - shortly before the end of the wait period (representing the amount of analyte that remains bound after the wait period).

- **Regeneration**: shortly after the regeneration procedure (representing the efficiency of regeneration).

The response at **Resp2** for one injection is used as the baseline for the next injection. The **Regeneration** response is calculated relative to **Baseline**.

Note: For curves which are not reference-subtracted, the response at **Resp2** will include the bulk refractive index contribution from the injected sample.

You select which response levels to display in the **Cycles** and **Injections** dialogues. Injections are identified as 1st, 2nd, 3rd and 4th, with the sample name in the **Injections** list for additional identification. Holding the mouse pointer over a histogram bar shows a tool tip which gives full details of the injection for that sample.

You can display results grouped by injections (for ease of comparing corresponding injections between cycles) or by cycles (for ease of comparing corresponding cycles for the same injection) by selecting the appropriate tab in the display window. The **Baseline stability** tab shows trends in the baseline response (at the beginning of each cycle) as a function of cycle number.

5.5 The kinetic analysis wizard

5.5.1 Kinetic analysis wizard functions

The kinetic analysis wizard provides support for:

- Analysis using covalently immobilized or non-covalently captured ligand.
- Determination of kinetic parameters from analyte concentration series. Apparent kinetic constants based on a simple 1:1 binding model are calculated and presented automatically.
- Control experiment to reveal mass transfer limitations based on dependence of binding on flow rate.
- Control experiment to reveal linked reactions or kinetic heterogeneity based on the dependence of dissociation rate on incubation time at steady state.

A flow diagram for the kinetic analysis wizard is shown in Figure 2.

The kinetic analysis wizard may be used with any type of sensor chip that supports direct immobilization or capturing of ligand. The sensor chip type is not specified in the wizard settings.

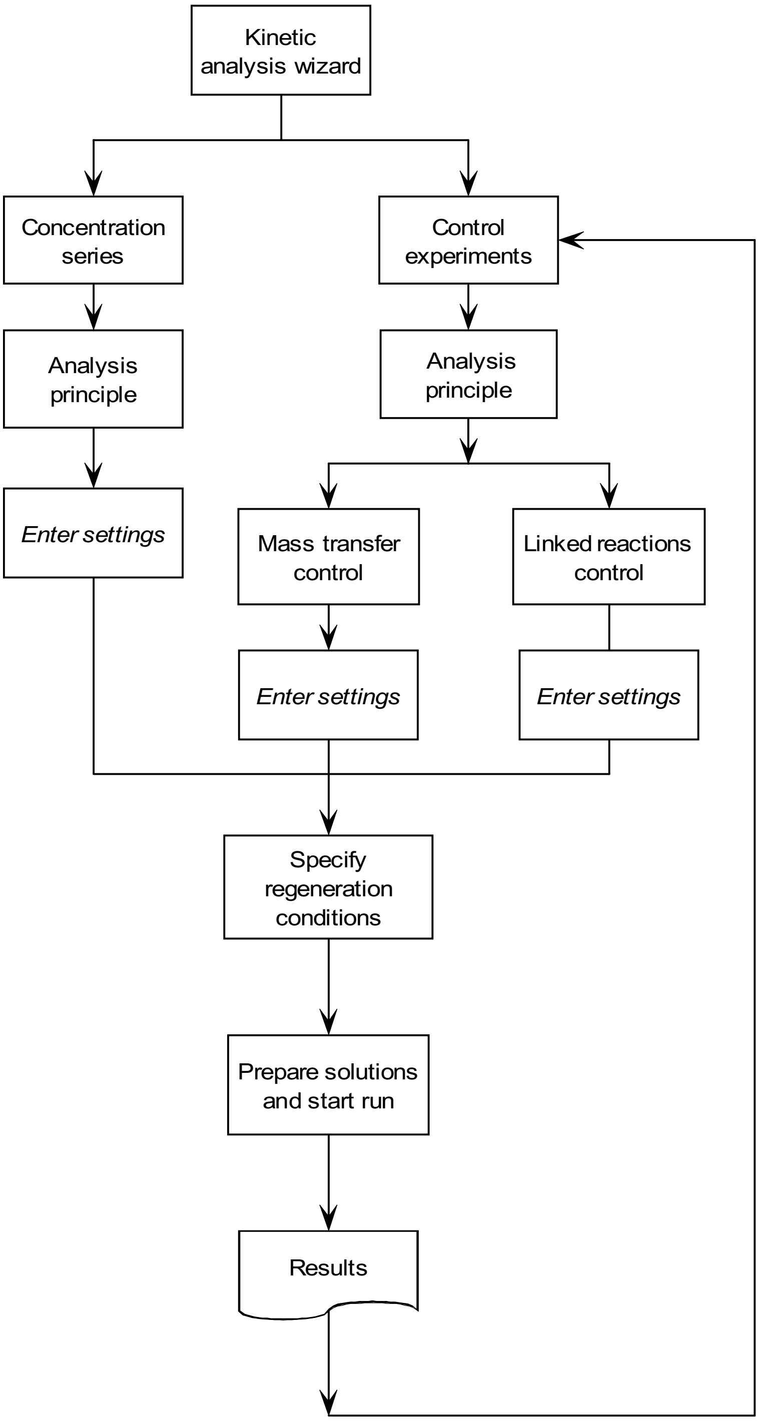
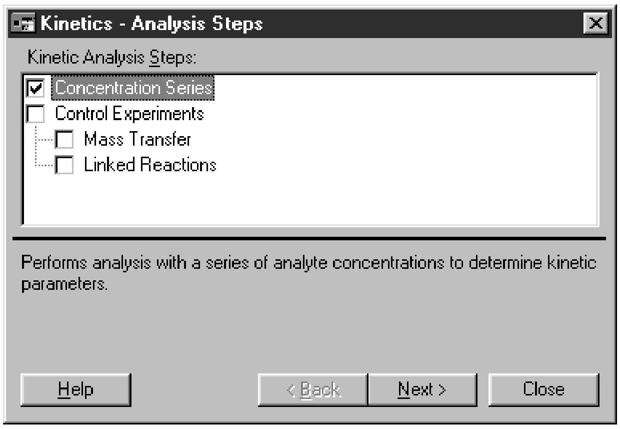


Figure 2. *Flow chart for the kinetic analysis wizard.*

When you start the kinetic analysis wizard, you can choose to run any or all of the concentration series and control experiments in a single "batch" run. Check the experiments you wish to perform.



Performs analysis with a series of analyte concentrations to determine kinetic parameters.

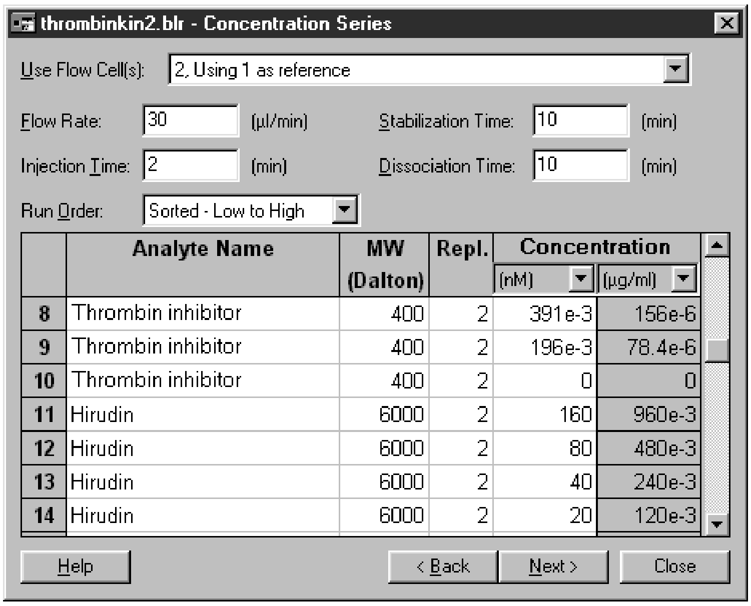
As a recommendation, run the concentration series first without checking any control experiments. When the analysis is finished, click **Next** in the result display (see below) without closing the wizard, and run the control experiments. If you work this way, the application wizard will use the results of the concentration series to suggest analyte concentrations for the control experiments.

If you choose to run both the concentration series and control experiments in a single batch, or if you close the wizard between completing the concentration series and running the control experiments, the wizard will not be able to suggest analyte concentrations for the controls and you must provide these values yourself. Note that if you have not analysed the interaction previously you may not be able to choose suitable concentrations for the control experiments. Concentrations should be chosen as follows:

- for the mass transfer control experiment, the rate of binding should be such that a readily measurable response is obtained with an injection time of 2 minutes. If the binding is too fast or too slow, the results may be difficult to interpret. As a rule of thumb, the response reached after 2 minutes should be less than 50% of the maximum analyte binding capacity of the surface.

- for the linked reactions control, the binding should ideally approach steady state within 1-2 minutes. Frequently this means using the highest analyte concentration available. If the binding does not reach steady state, comparison of the dissociation phases can be misleading.

Running several analytes in one run
You can enter several different analytes in the sample table for the kinetic analysis wizard:



Use Flow Cell(s): 2, Using 1 as reference

Flow Rate: 30 (µl/min)

Stabilization Time: 10 (min)

Injection Time: 2 (min)

Dissociation Time: 10 (min)

Run Order: Sorted - Low to High

	Analyte Name	MW	Repl.	Concentration	
		(Dalton)		(nM)	(µg/ml)
8	Thrombin inhibitor	400	2	391e-3	156e-6
9	Thrombin inhibitor	400	2	196e-3	78.4e-6
10	Thrombin inhibitor	400	2	0	0
11	Hirudin	6000	2	160	960e-3
12	Hirudin	6000	2	80	480e-3
13	Hirudin	6000	2	40	240e-3
14	Hirudin	6000	2	20	120e-3

Note however that the same regeneration conditions will be used for all cycles in the same run. If you want to use different regeneration conditions for different analytes, run the analytes in separate sessions.

5.5.2 Notes on the kinetic analysis wizard

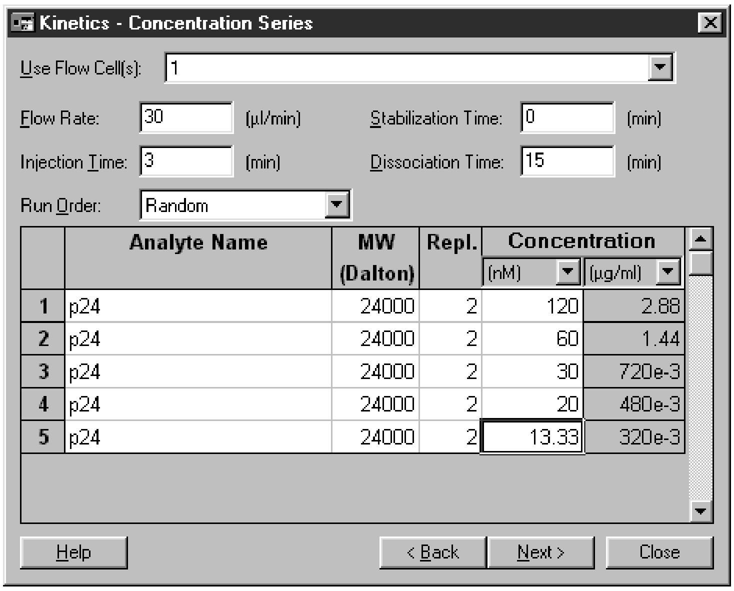
Choice of analysis principle
The kinetic analysis wizard supports analysis using either directly immobilized or captured ligand. This choice is usually dictated by the requirements of the analysis.

Choice of flow cells
You should if possible run kinetic analysis using on-line subtraction of a reference flow cell. This will make interpretation of the results easier by eliminating response contributions such as bulk refractive index in the samples. Ideally, the reference flow cell should be prepared by immobilizing an irrelevant (non-binding) protein to the same level as the ligand, to compensate for any adverse effects that might arise from the presence of protein on the surface. These effects are small if any, but can be significant in very careful kinetic analyses using low response levels. If no suitable irrelevant protein is available, you can use a blank reference flow cell.

Concentration series

When entering details of the concentration series, it is most convenient to enter the highest analyte concentration first. The wizard will automatically suggest values for the next cell row when you press **Enter** to complete an entry, in accordance with the descriptions below. You can change any suggested entry simply by typing a new value:

- the analyte name is copied from the row above
- the analyte molecular weight is copied from the row above: for analyte names which have been used recently, the previously used molecular weight is suggested
- the number of replicates is copied from the row above
- the concentration is calculated from the row above, by multiplying by 0.5 for the second row or by the ratio of the two preceding rows for subsequent rows. This makes it easy to enter a regular dilution series. The factor is reset to 0.5 when the analyte name is changed.



Kinetics - Concentration Series

Use Flow Cell(s): 1

Flow Rate: 30 (µl/min) Stabilization Time: 0 (min)

Injection Time: 3 (min) Dissociation Time: 15 (min)

Run Order: Random

	Analyte Name	MW (Dalton)	Repl.	Concentration (nM)	(µg/ml)
1	p24	24000	2	120	2.88
2	p24	24000	2	60	1.44
3	p24	24000	2	30	720e-3
4	p24	24000	2	20	480e-3
5	p24	24000	2	13.33	320e-3

As a recommendation, you should always run at least 5 different analyte concentrations, with at least one of the concentrations in replicate. The software will advise you if these conditions are not fulfilled when you click on **Next** to continue with the wizard.

It is also advisable to include a sample with zero analyte concentration (this means an injection of analyte buffer). Ideally, this should be a straight baseline after subtraction of the reference response: however, inclusion of such a sample in the concentration series can facilitate assessment of baseline deviations in the final data.

Conditions for analysis and control experiments
For control experiments to be valid, it is important that the concentration series and controls are performed under identical conditions (sample preparation, buffer, age of sample etc). The best way to achieve this is to run the controls directly after the concentration series without closing the wizard window: the wizard will then suggest suitable conditions for each control. If the controls are run on a separate occasion, make sure

- that the same sensor chip is used as for the concentration series, and that the ligand activity has not deteriorated

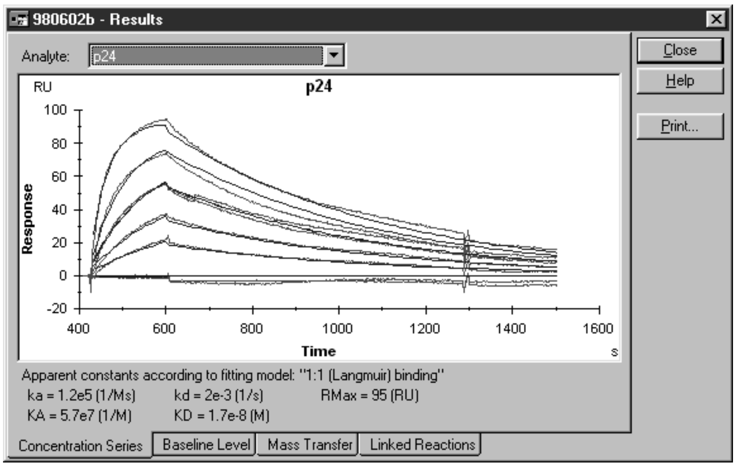
- that the analysis principle and regeneration conditions are the same as those used in the concentration series (the correct values will be suggested if you open the wizard template from the result file, see Section 5.1).

5.5.3 Intepreting results from the kinetic analysis wizard

Results from the kinetic analysis wizard are presented in a tabbed dialogue box with separated tabs for the concentration series and control experiments. If more than one analyte has been run, the analyte can be selected from the pull-down list at the top of the display:

Note: If you run the concentration series and control experiments in separate wizard sessions (closing the wizard dialogue box between runs), the results of each session will be saved in separate files. The tabs in the wizard result presentation represent experiments run within the same wizard session.

Concentration series
The results from the concentration series are shown as an overlay plot with the sensorgrams aligned at the time of injection and the baseline adjusted to zero before injection. The results are evaluated according to a simple 1:1 interaction model; experimental curves are shown in red and fitted curves in blue. Apparent kinetic constants according to the 1:1 fitting are listed, together with the estimated maximum analyte binding capacity (R_{max}).



Analyte: p24

Apparent constants according to fitting model: "1:1 (Langmuir) binding"

ka = 1.2e5 (1/Ms) kd = 2e-3 (1/s) RMax = 95 (RU)

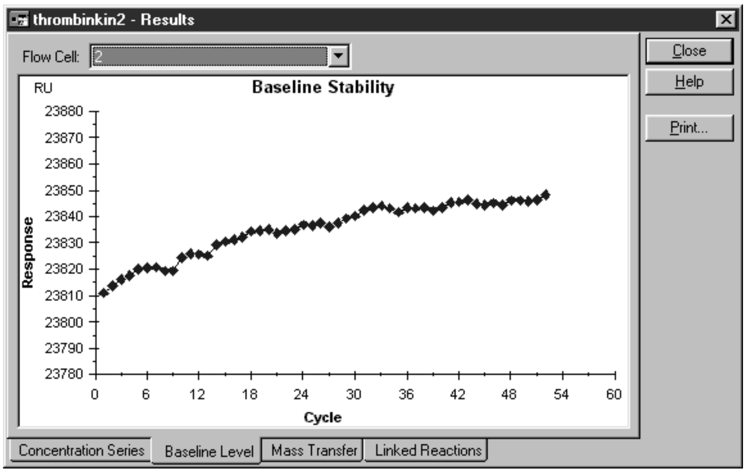
KA = 5.7e7 (1/M) KD = 1.7e-8 (M)

It is important to realise that the kinetic constants reported in the wizard results are derived from the evaluation according to the simplest possible evaluation model, and are consequently apparent constants which may deviate significantly from the true constants for the interaction. More careful evaluation of the kinetic data using BIAevaluation software is always recommended, particularly if the experimental and fitted curves do not coincide in the wizard result window.

The evaluation reported in the wizard results is performed using the actual injection times for start and stop of the association phase, and uses all data in the sensorgrams. If the data is evaluated with BIAevaluation software where parts of the sensorgrams may be excluded from analysis, significant differences in kinetic values may be obtained even if the same interaction model is applied.

Baseline stability

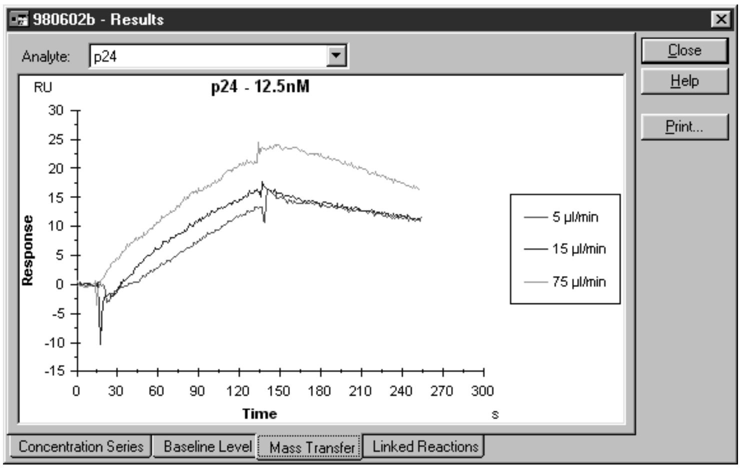
The **Baseline Level** tab displays the baseline response immediately after regeneration as a function of cycle number. Results are presented separately for the flow cells used in the analysis (accessed from the pull-down list), and for the reference-subtracted signal if an in-line reference cell is used. All cycles are presented, including any control experiments: the values are not distinguished according to analyte when the run uses several analytes.



This presentation helps you to assess the regeneration performance during the analysis. Note that the scale of the presentation may give a false visual impression of the baseline stability: in the example here, the baseline changes by about 40 RU over 50 cycles, or less than 1 RU per cycle.

Mass transfer control experiment

The mass transfer control experiment analyses one concentration of analyte at three different flow rates (5, 15 and 75 µl/min), using a 2 minute injection. If the interaction rate is limited by mass transfer, the observed binding rate will vary with flow rate (kinetic interactions are independent of flow rate while mass transfer is proportional to the cube root of the flow rate). Thus if the association phase in the three curves presented in the mass transfer control results show significantly different slopes, mass transfer limitations cannot be ignored. The example below indicates negligible mass transfer limitation. See the on-line help for more examples.



If mass transfer limitations are serious, the apparent kinetic constants reported in the wizard result window bear no relation to the kinetic constants for the interaction itself.

Note that if binding at the analyte concentration used is very fast or very slow, it may be difficult to decide whether the three curves show the same slope or not. If this is the case, repeat the control experiment using a more suitable analyte concentration.

If mass transfer limitations are apparent, either:

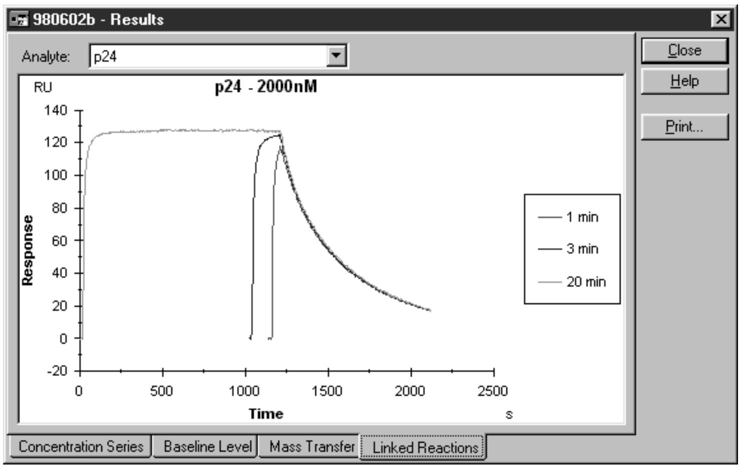
- repeat the analysis using a lower level of immobilized ligand to reduce the significance of mass transfer, or
- evaluate the results using a mass transfer-limited model in BIAevaluation software.

Linked reactions control

The linked reactions control experiment compares the dissociation rates for one analyte concentration after injection times of 1, 3 and 20 minutes. Ideally all three sensorgrams should approach steady state binding. The design of this experiment is based on an interaction model of the type

A + B –> AB –> AB*

where dissociation of AB* proceeds at a significantly different rate from dissociation of AB. Assuming that the rate of conversion of AB to AB* is such that the proportion of AB* changes between 1 and 20 minutes, this reaction scheme will give differently shaped dissociation profiles after different incubation times. The example below does not indicate linked reactions, since the dissociation phases are practically identical. See the on-line help for more examples.



Note that dissociation phases which coincide do not exclude an interaction mechanism of this type:

- if the dissociation rates of AB and AB* are similar, the differences in dissociation profiles will be very small

- if the conversion AB to AB* is essentially complete within 1 minute, all three curves will reflect dissociation of only AB*.

If the interaction does not approach steady state in all three sensorgrams, interpretation of the results is more complicated, and other deviations from the 1:1 interaction model (e.g. mass transfer limitations) can give rise to differences in the dissociation profiles.

6. Customized applications

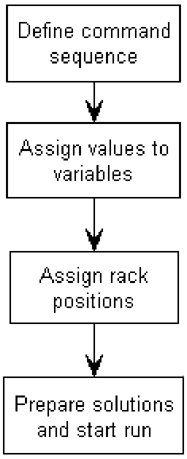
Customized applications can be set up with the help of the Customized Application Wizard. Unlike the other application wizards, this does not provide assistance for running a specific application. Instead, the application is defined in terms of a freely constructed command sequence. The wizard then guides you in setting up samples and reagents in the autosampler rack and in starting the run, using the same kind of dialogue boxes as in the other application wizards (Section 5.1.3).

The customized application wizard will handle most experimental requirements that are not covered by the other application-specific wizards. For complete control of instrument operation, however, the Method Definition Language (MDL) is a programming environment that offers a greater degree of flexibility than the customized application wizard. MDL is documented in Appendices E and F in this manual.

Commands in the customized application wizard provide support for:

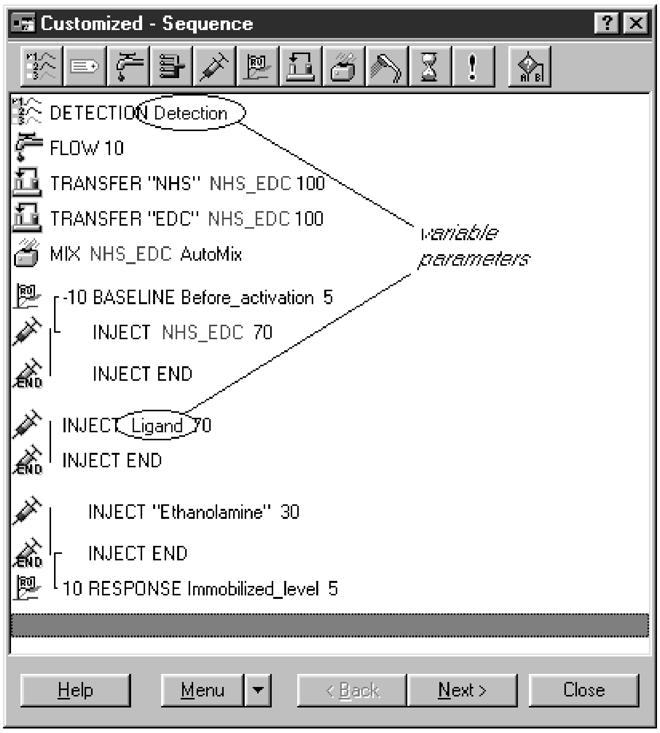
- standard instrument functions such as detection mode, transfer, mix, inject and report point,
- variable command parameters that may be assigned different values for different cycles in the analysis
- conditional ("if-then") commands that control instrument operation on the basis of variables or report point values.
-

6.1 Overview of the customized application wizard



6.1.1 Defining the command sequence

Starting the customized application wizard opens a window where you define the sequence of commands that constitute one cycle in the application:



```
DETECTION Detection
FLOW 10
TRANSFER "NHS" NHS_EDC 100
TRANSFER "EDC" NHS_EDC 100
MIX NHS_EDC AutoMix
┌-10 BASELINE Before_activation 5
└    INJECT NHS_EDC 70
     INJECT END
INJECT Ligand 70
INJECT END
     INJECT "Ethanolamine" 30
     INJECT END
┌
└10 RESPONSE Immobilized_level 5
```

When first opened, the windows is empty except for a FLOW 20 command that sets the buffer flow to 20 µl/min at the beginning of the run. You can edit this command to change the flow rate, but you cannot delete it. The example shown above is a simple amine coupling sequence. The parameters **Detection** and **Ligand** (encircled in the illustration above, shown in blue text in the display) are variables which allow the immobilization sequence to be used for different ligands in different flow cells within a single run.

- To *add commands* to the end of the sequence, click on the toolbar button for the command you want to add. Alternatively you can choose the command from the **Menu** button, or right-click in the window area and choose the command from the context menu. Short-cut key combinations for the commands are listed in the menus. Adding a command opens a dialogue box where you set parameters for the command.

- To *insert a command* in an existing sequence, select the command immediately after the insertion point and follow the procedure for adding a command.

- To ***edit a command***, double-click on the command. This will open the dialogue box for the command parameters.
- To ***delete a command***, select the command and press Del on the keyboard.
- To ***move a command***, select the command and drag it to the new position. Conditional commands cannot be moved, and there are restrictions on how you can move certain other commands (e.g. there must always be a FLOW command as the first instrument operation at the beginning of the sequence).

Command sequence display

Commands in the sequence are presented as an icon and a specification. The specification is based on the MDL syntax (Appendix F), with some adaptations for customized command display. Some command keywords have been adapted to make it easier to understand the command functions: for example report points are identified as BASELINE or RESPONSE depending on whether the report point defines a baseline or not. See the on-line help for details of the display syntax.

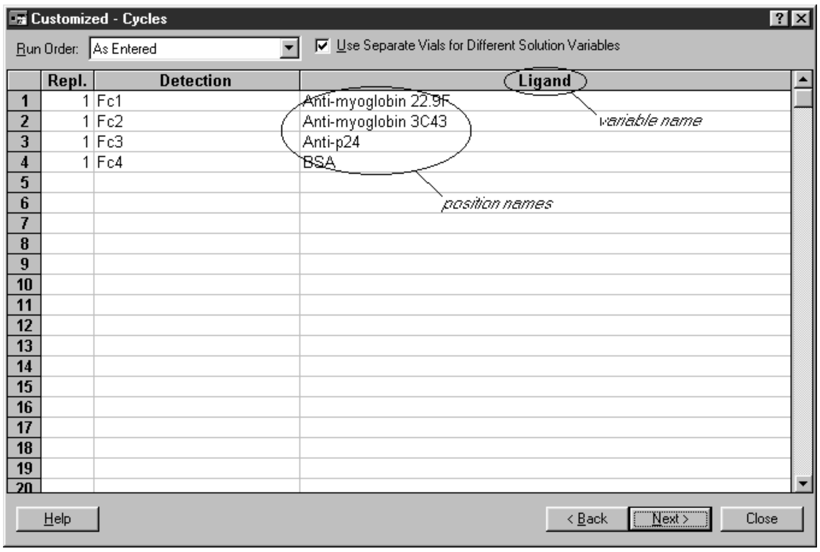
Command parameters may be constant or variable (see Section 6.2). Constant parameters are shown in the window in black text inside quotation marks. Variable parameters are shown in blue text. There is also a special kind of variable representing empty vials for mixing positions, shown in green text.

Report points are set in relation to other commands (at a specified time before or after), and are “tied” in the display to the respective command with a vertical line. Report points move with a command when the command is moved, and are deleted if the command is deleted.

The INJECT command is always displayed together with a corresponding INJECT END. This has no operational function, but allows report points to be set in relation to the end of a sample injection without having to calculate the contact time represented by the injection.

6.1.2 Assigning values to variables

When you have defined the sequence of commands, click on **Next** to go to the **Cycles** dialogue box:



Customized - Cycles

Run Order: As Entered

☑ Use Separate Vials for Different Solution Variables

	Repl.	Detection	Ligand
1	1	Fc1	Anti-myoglobin 22.9F
2	1	Fc2	Anti-myoglobin 3C43
3	1	Fc3	Anti-p24
4	1	Fc4	BSA
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

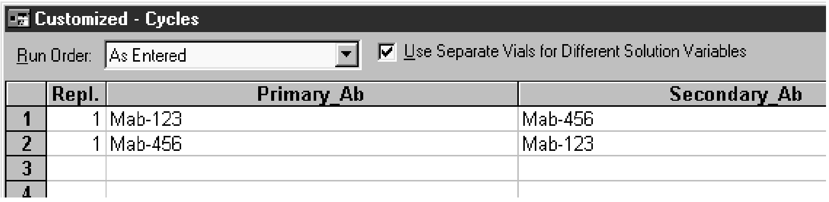
Here you set the number of replicate runs for each cycle and values for any variables used in the command sequence (variables allow you to run multiple cycles using the same sequence of commands). You can also choose whether the cycles will be run in random order or in the order entered in the table.

The table in this dialogue box contains one column for the number of replicates and one for each variable used in the command sequence. The example here defines four cycles, to immobilize four different ligands in flow cells 1-4 respectively.

If you want to change the order in which the columns are displayed, click on the **Menu** button and select **Set column order**. You cannot change the order for the **Repl** and **Detection** columns.

If the option **Use Separate Vials for Different Solution Variables** is checked, entries that have the same values in different variable columns will use separate vials in the autosampler rack (entries with the same value within one column use the same vial). If this option is not checked, entries with the same value in different columns will be mapped to the same vial position.

Check the **Use Separate Vials for Different Solution Variables** box if you want to use position names that might not uniquely identify a sample, for example if you use the sample name as the variable name and a concentration as the position name in the table. Uncheck the box if you use the same solution in different contexts in an analysis, such as epitope mapping where the same antibodies may be used as first or second antibodies, illustrated in the following example. For an application with two injections using variables **Primary_Ab** and **Secondary_Ab**, the table



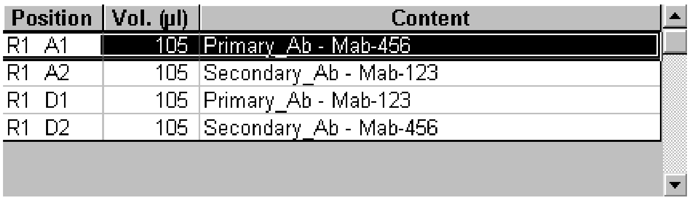
Customized - Cycles

Run Order: As Entered

☑ Use Separate Vials for Different Solution Variables

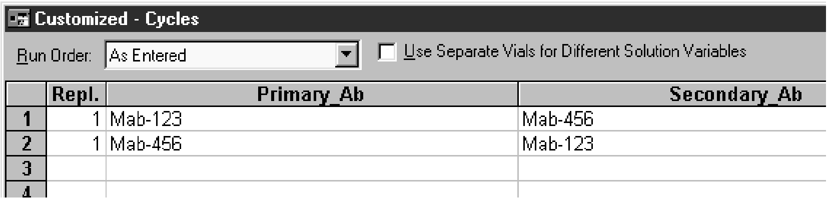
	Repl.	Primary_Ab	Secondary_Ab
1	1	Mab-123	Mab-456
2	1	Mab-456	Mab-123
3			
4			

with **Use Separate Vials for Different Solution Variables** checked will give four samples in the **Rack Positions** dialogue:



Position	Vol. (µl)	Content
R1 A1	105	Primary_Ab - Mab-456
R1 A2	105	Secondary_Ab - Mab-123
R1 D1	105	Primary_Ab - Mab-123
R1 D2	105	Secondary_Ab - Mab-456

However, with **Use Separate Vials for Different Solution Variables** not checked:



Customized - Cycles

Run Order: As Entered

☐ Use Separate Vials for Different Solution Variables

	Repl.	Primary_Ab	Secondary_Ab
1	1	Mab-123	Mab-456
2	1	Mab-456	Mab-123
3			
4			

only two sample positions are generated since the variable names **Mab-123** and **Mab-456** each map to the same position regardless of the variable column in which they are used.

6.1.3 Assigning rack positions

When values have been assigned to all variables, click on **Next** to go to the **Rack Positions** dialogue box. See Section 5.1.3 for a description of this dialogue box.

6.1.4 Preparing the run

Click on **Next** in the **Rack positions** dialogue box to go to the **Prepare run** summary. This dialogue summarizes the solutions and vial positions required in the autosampler (see Section 5.1.3). In some cases (if you have used TRANSFER commands in your sequence) empty vials will be needed in some autosampler positions.

Note: In some circumstances, particularly in sequences involving conditional commands, the required volume of solution cannot be accurately determined in advance. The volumes shown in the **Prepare run** dialogue box are “worst-case” volumes, calculated as though all commands are executed in all cycles.

6.2 Constant and variable command parameters

The customized application wizard allows you to use either constant or variable parameters for most of the commands in the sequence:

- Use a constant parameter if the value is the same in every cycle of an analysis. For example, if you are going to inject the same volume of sample in each cycle, you would use a constant value for the sample volume.

  Most constant parameters are entered as straightforward values (e.g. “20” for a sample volume of 20 µl). Values that correspond to positions in the autosampler rack, however (such as sample identification for an INJECT command) are entered as a name, which is associated with a rack position in the **Rack Positions** dialogue box later in the wizard.

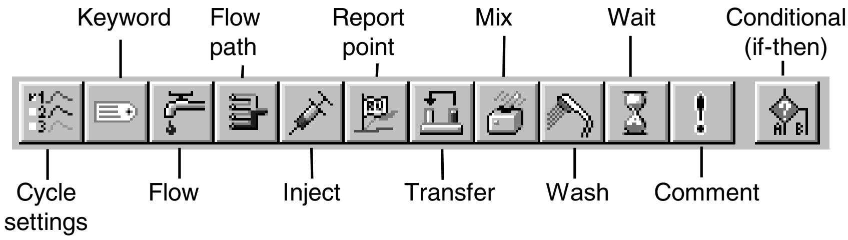
- Use a variable parameter (check **Vary by Cycle**) if you want to vary the value between cycles. For example, if you are running a screening application and want to inject a different sample in each cycle, use a variable for the sample identification. You set the number of cycles and assign a value to the variables for each cycle in the **Cycles** dialogue box later in the wizard.

  In general, use variable parameters as sparingly as possible. Since variables are assigned a value for each cycle in the analysis, you will create a lot of data entry work if you use variables unnecessarily.

Note: The **To** position for TRANSFER is represented by a special kind of variable which creates a new empty vial for each cycle in a run (the primary use of TRANSFER is to prepare mixed samples for injection, and this feature prevents you from preparing a fresh mixture in a used sample vial). The empty vial variable does not appear in the **Cycles** table, since positions are assigned automatically. You do not need to check **Vary by Cycle** in dialogue boxes for commands like INJECT in order to address this kind of variable.

6.3 Customized application commands

Commands supported in the customized application wizard are all accessible from the toolbar. Most of the commands have clear counterparts in the manual operation menus (see Chapters 4 and 9):



Cycle settings
Sets the temperature at the flow cell and the detection mode for the current cycle.

Note: Setting the detection mode also sets the flow path. See Section 6.4 for hints on using the detection mode to best advantage.

Keywords
Defines keywords that will appear in the report point table for the run. You set the keyword value as a variable: the value of the variable for each cycle will appear in the keyword column in the report point table (see Section 6.5).

Flow
Sets the flow rate.

Note: When you create a new sequence, a single FLOW command that sets the flow rate to 20 µl/min is automatically inserted at the beginning of the sequence. You can change the parameters for this command, but you cannot delete it and you cannot place other commands except cycle settings and KEYWORD before the first FLOW. This ensures that the flow rate is set before any other commands are executed in an analysis.

Flow path
Sets the flow path.

Inject
Injects sample over the sensor surface

Report point
Sets report points at a fixed time relative to a chosen command.

Transfer
Transfers solution from one autosampler vial to another, or from the buffer reservoir to a vial.

Mix
Mixes the contents of a specified vial.

Wash
Washes either the autosampler needle or the IFC channels.

Wait
Inserts a delay period between commands.

Comment
Inserts a comment in the command sequence, for documentary purposes. Comments do not affect execution of the commands in the sequence.

Conditional commands (if-then)
Allows conditional execution of commands or termination of the cycle or run, depending on the value of a report point or variable.

The sections below offer some hints on how to construct applications using the commands in the customized application wizard. For more detailed descriptions of the commands themselves, see the on-line help.

6.4 Using detection mode and flow path

6.4.1 Detection mode

The DETECTION command in **Cycle settings** sets both the detection mode and the flow path. The detection mode can be set to a constant value in terms of flow cells and reference subtraction options, chosen from the pull-down list or to a variable named **Detection**. (This variable name is predefined and cannot be changed.)

Setting the detection mode to a variable allows you to change detection mode and corresponding flow path between cycles in a run. The example illustrated in Section 6.1.1 shows how this can be used to address a different flow cell in each cycle. Note that the detection mode determines both the initial flow path and the number of sensorgram curves displayed.

If your customized command sequence does not include a DETECTION command, you will be asked to set the detection mode at the beginning of the run. You can only choose a detection mode that is compatible with the commands in the command sequence. For example, if your sequence includes a FLOW PATH 3 command, you can only choose a detection mode that includes Fc3.

6.4.2 Flow path

The FLOW PATH command allows you to change the flow path within the limitations set by the detection mode. For example, if the detection mode is set to Fc1-2, you can only choose flow cell 1 or 2 in the FLOW PATH command.

You cannot use a variable parameter with the FLOW PATH command, which means that you cannot use this command to change the flow path between cycles. Use FLOW PATH only if you want to change the flow path ***within*** a cycle. Use a variable detection mode if you want to change the flow path between cycles.

6.4.3 Flow cell names

Apart from the DETECTION command, you can refer to flow cells in FLOW PATH commands to direct the flow and in IF-THEN commands to set a condition based on the response in a specified flow cell.

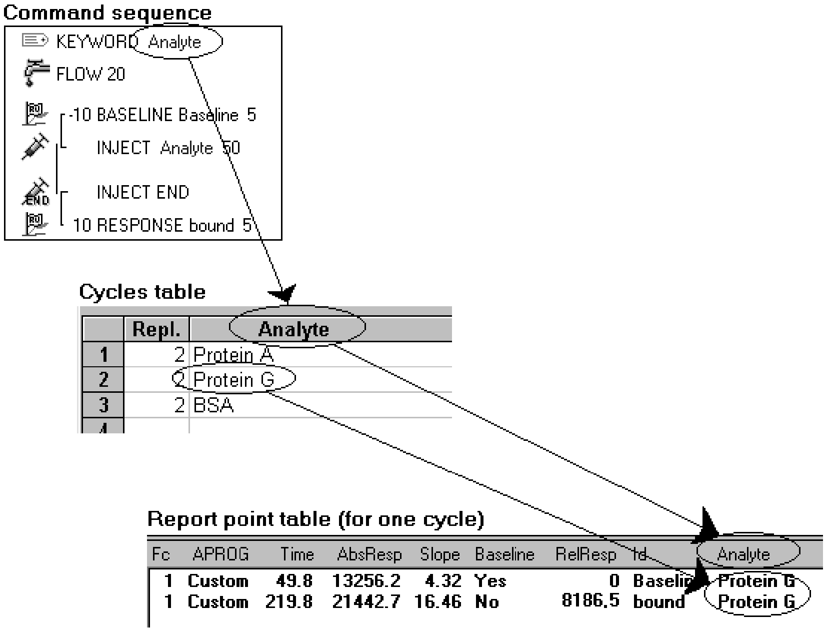
If you have used a one- or two-channel detection mode, or if the detection mode is not specified, you can choose the logical flow cell names **Sample**, **Reference**, and **Subtracted** for FLOW PATH and IF-THEN commands. The table below lists the correspondence between logical flow cell names and actual flow cells for the different detection modes.

Detection mode	**Sample**	**Reference**	**Subtracted**
Fc1	Fc1	n/a	n/a
Fc2	Fc2	n/a	n/a
Fc3	Fc3	n/a	n/a
Fc4	Fc4	n/a	n/a
Fc1,2 Ref 2-1	Fc2	Fc1	Fc2-1
Fc3,4 Ref 4-3*	Fc4	Fc3	Fc3-4
Fc1,2	Fc1	n/a	n/a
Fc1,2,3,4	Fc1	n/a	n/a
Fc1,2,3,4 Ref 2-1, 4-3	Fc2	Fc1	Fc2-1
Fc1,2,3,4 Ref 2-1,3-1,4-1	Fc2	Fc1	Fc2-1

This option is particularly useful for IF-THEN commands, and allows you to set a condition based on a response without knowing in advance which flow cell is in use (see Section 6.11 for examples).

6.5 Using keywords

Keywords generate user-defined columns in the report point table for the run, with the keyword name as column heading. The name is defined as a variable in the customized sequence. The column content is the variable value for each cycle as entered in the **Cycles** table.



Command sequence

```
KEYWORD Analyte
FLOW 20
-10 BASELINE Baseline 5
    INJECT Analyte 50
    INJECT END
10 RESPONSE bound 5
```

Cycles table

	Repl.	Analyte
1	2	Protein A
2	2	Protein G
3	2	BSA
4		

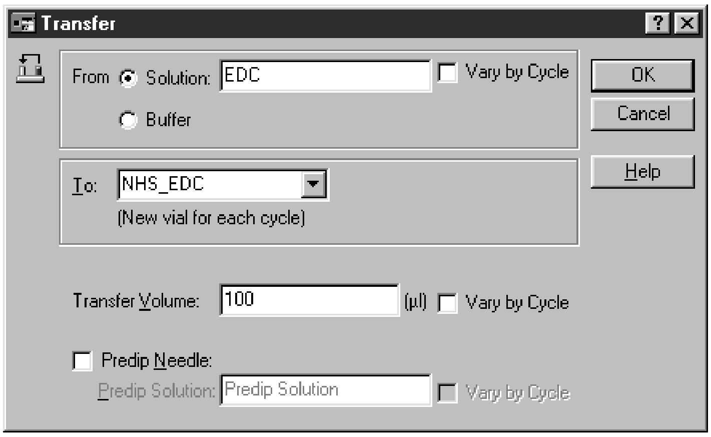
Report point table (for one cycle)

Fc	APROG	Time	AbsResp	Slope	Baseline	RelResp	Id	Analyte
1	Custom	49.8	13256.2	4.32	Yes	0	Baselin	Protein G
1	Custom	219.8	21442.7	16.46	No	8186.5	bound	Protein G

If you add keywords after entering the other commands in the sequence, the **Keyword value** field presents a pull-down list of all variables used in the commands, making it easy to select the appropriate variable name. The KEYWORD command itself is however always placed at the beginning of the sequence.

6.6 Using empty mixing vials

Applications that use the TRANSFER and MIX commands to prepare solutions in the autosampler vials generally require a new vial for every cycle, to avoid carry-over effects between cycles. The dialogue box for TRANSFER automatically handles this requirement:



Transfer

From ● Solution: EDC □ Vary by Cycle

○ Buffer

To: NHS_EDC

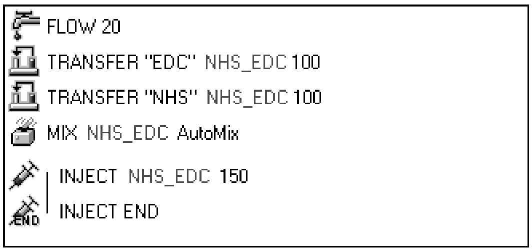
(New vial for each cycle)

Transfer Volume: 100 (µl) □ Vary by Cycle

□ Predip Needle:

Predip Solution: Predip Solution □ Vary by Cycle

The **To** position generates a new empty vial for each cycle in the run (in this example for preparing fresh NHS/EDC for each cycle). The position name for the **To** position can be addressed for mixing, injecting etc but does not appear in the **Cycles** table. Empty vial positions are assigned automatically for each cycle and are shown in green in the command sequence. The sequence below shows how TRANSFER, MIX and INJECT commands may be used to prepare and inject NHS/EDC:

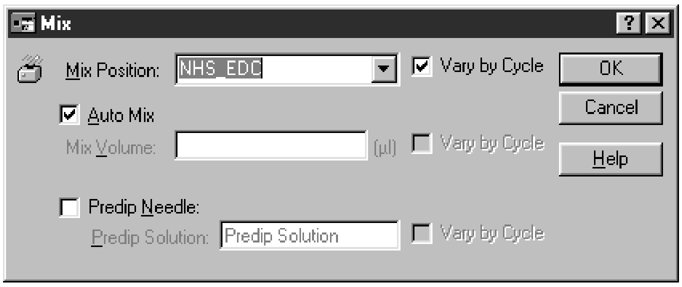


```
FLOW 20
TRANSFER "EDC" NHS_EDC 100
TRANSFER "NHS" NHS_EDC 100
MIX NHS_EDC AutoMix
  INJECT NHS_EDC 150
  INJECT END
```

Note: The first occurrence of the position name for a **To** position in the sequence must be in a TRANSFER command, since the position represents an initially empty vial. The wizard will not accept usage in any earlier command in the sequence.

6.7 Using MIX

The MIX command mixes the solution in a specified rack position (see Section 9.7).


Mix

Mix Position: NHS_EDC ☑ Vary by Cycle

☑ Auto Mix

Mix Volume: (µl) □ Vary by Cycle

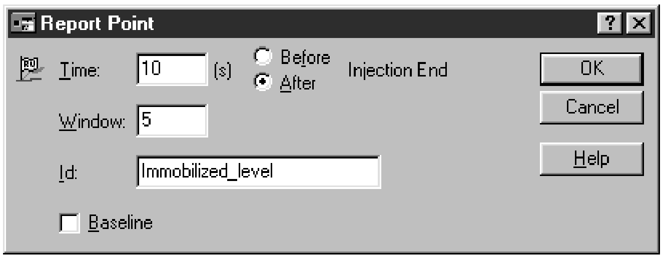
□ Predip Needle:

Predip Solution: Predip Solution □ Vary by Cycle

If you check the **Automix** option, the wizard will determine the optimum mixing volume according to the volume of liquid currently in the vial. With this option unchecked, you specify the mixing volume in µl (as a general guideline, this should be 75% of the volume of liquid in the vial).

6.8 Using report points

Rerport points are set at a fixed time before or after other commands. To set a report point, select the command to which the report point refers, then click on the **Report Point** button on the toolbar.


Report Point

Time: 10 (s) ○ Before ● After Injection End

Window: 5

Id: Immobilized_level

□ Baseline

You can also set report points as isolated commands within an IF-THEN construction (see Section 6.9). To do this, enter an IF-THEN command using the command block alternative, select the END IF-THEN command and insert a report point.

6.9 Using conditional commands

Conditional commands allow you to control the sequence of commands that are actually executed depending on the outcome of the analysis. The basic structure of a conditional command is as follows:

```
IF condition THEN
    consequence
END IF
commands
```

If the condition is fulfilled, the commands represented by "consequence" are executed, followed by the commands after the END IF line. If the condition is not fulfilled, execution continues directly with the commands that follow END IF.

The condition is one of the following:

RelResp Relative response for a specified report point on a specified sensorgram curve. The report point must be a RESPONSE set before the conditional command is inserted, and must be preceded by a BASELINE command from which to calculate a relative response.

AbsResp Absolute response for a specified report point on a specified sensorgram curve. The report point must be set before the conditional command is inserted.

Slope Slope for a specified report point on a specified sensorgram curve. The report point must be set before the conditional command is inserted.

Variable The value for a specified variable (**Detection** and solution variables cannot be used in conditional commands). The variable must be defined before the conditional command is inserted.

If required, an additional condition can be added, with the logical operator **AND** (both conditions must be fulfilled) or **OR** (only one condition needs to be fulfilled).

The consequence may be one of the following:

Exit cycle End the current cycle and continue with the next cycle.

Exit method End the run, ignoring any cycles that may not have been performed.

Command block A user-defined sequence of commands.

Command block
ELSE
Command block Two command sequences separated by ELSE. In this construction, the first sequence is executed if the condition is fulfilled and the second if the condition is not fulfilled. In either case, execution then continues after the END IF line.

Editing, deleting and moving conditional commands
Because of the logical complexity that can be introduced by the use of conditional commands, there are some restrictions on how these commands can be handled in the customized command sequence:

- You cannot add conditional commands so that already existing commands are directly included in the conditional block. To place existing commands in a conditional block, add the conditional command first and then drag the other commands into the block.

- You cannot move a conditional command. You can however drag commands out of a conditional command. One way to effect a move is to enter a new conditional command in the required new position and drag any commands from the old condition to the new one.

- You cannot delete a conditional command without deleting the command block(s) included in it. To delete a conditional command but retain the included command(s), first drag the commands outside the condition and then delete the condition.

- Once you have entered a condition, you cannot change the type of consequence. If you want to change the consequence, enter a new conditional statement and delete the old one. Remember to drag any commands you want to keep out of the old condition before you delete it.

6.10 Using WAIT

The WAIT command inserts a delay period between commands in the sequence. The delay starts when the command preceding the WAIT (including setting any report points associated with the command) is completed, and ends when the next command (***excluding*** any report points that are set before the command) is initiated. For INJECT operations, the start of the command is counted as the time when the operation actually begins, excluding any preparation time that might be needed. Note that WAIT command immediately preceding an INJECT command will have no effect if the delay period is shorter than the autosampler preparation time (see Section 4.4.6).

This is illustrated in the diagram below:

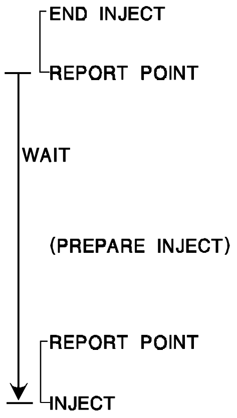


Diagram showing the time relationships of WAIT to other commands, using a WAIT between the end of one injection and the beginning of the next for illustration purposes. The arrow at the left indicates the duration of the delay specified in the WAIT command.

6.11 Customized application examples

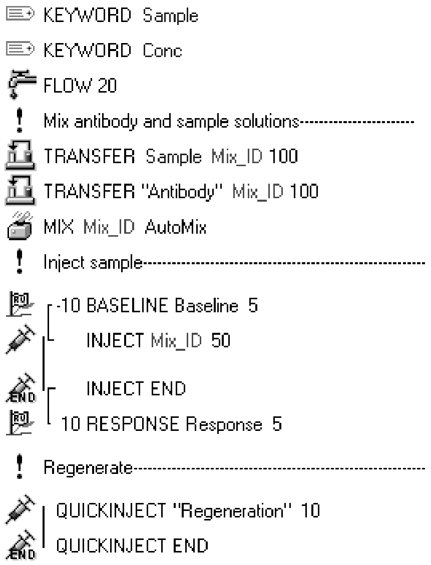
This section shows some simple examples of how applications can appear in the customized application wizard. The examples should be seen as illustrations of the principles of customized sequences rather than as working experimental templates.

6.11.1 Inhibition assay

This example illustrates an inhibition assay for concentration determination of a low molecular weight analyte. The sequence mixes sample with a known amount of antibody and injects the mixture over a surface on which purified analyte is immobilized. The response measures the amount of free antibody remaining in the sample, and is inversely related to the amount of analyte.

The sequence does not include a DETECTION command, so that the detection mode and flow path are set at the beginning of each run. This allows the sequence to be used with a free choice of flow cell.

The keywords **Sample** and **Conc** ensure that responses recorded in the report point table are adequately identified for later evaluation. The variable **Sample** in the KEYWORD command is the same as that in the INJECT command: the variable **Conc** is used only as a keyword, to record concentrations in the report point table.

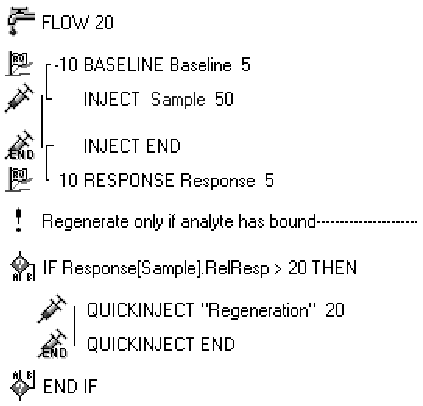


```
KEYWORD Sample
KEYWORD Conc
FLOW 20
! Mix antibody and sample solutions-----------------------
TRANSFER Sample Mix_ID 100
TRANSFER "Antibody" Mix_ID 100
MIX Mix_ID AutoMix
! Inject sample---------------------------------------------------------
-10 BASELINE Baseline 5
    INJECT Mix_ID 50

    INJECT END
10 RESPONSE Response 5
! Regenerate-----------------------------------------------------------
QUICKINJECT "Regeneration" 10
QUICKINJECT END
```

6.11.2 Conditional regeneration

This example illustrates the use of an IF-THEN command to regenerate the surface only if analyte has bound. This kind of construction can considerably reduce the total time required for a series of analyses.

Note that the sequence does not contain a DETECTION command, allowing free choice of flow cell at the beginning of the run. Correspondingly, the IF-THEN command refers to the relative response in flow cell **Sample**, which will map to the appropriate flow cell according to the detection mode used in any given run (see Section 6.4).

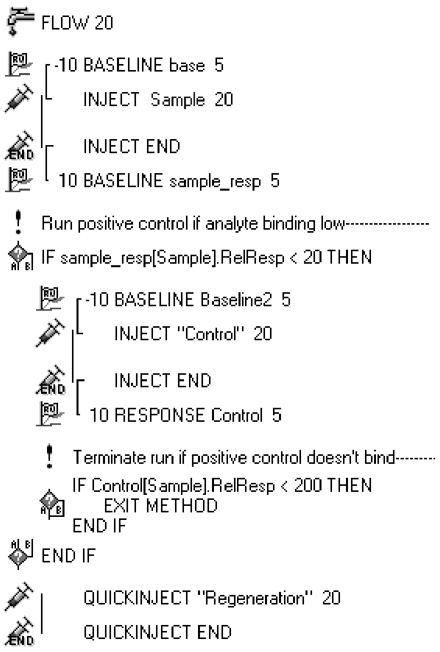


```
FLOW 20
┌-10 BASELINE Baseline 5
└    INJECT  Sample  50
┌    INJECT END
└ 10 RESPONSE Response 5
! Regenerate only if analyte has bound--------------------
IF Response[Sample].RelResp > 20 THEN
    QUICKINJECT "Regeneration" 20
    QUICKINJECT END
END IF
```

6.11.3 Conditional control experiments

This example illustrates how IF-THEN commands can be used to check the activity of a surface if no analyte binding is observed. The sequence uses one IF-THEN command to check analyte binding in the sample injection: if the response is less than 20 RU, a positive control sample is injected. A second IF-THEN command (nested within the first) terminates the run if the positive control does not show any binding, since in that case the surface is inactive and there is no point in continuing with the analysis.

The positive control (and the subsequent conditional termination of the run) will be performed only if analyte binding from the sample is less than 20 RU.



```
FLOW 20
┌-10 BASELINE base 5
└    INJECT  Sample  20
┌    INJECT END
└ 10 BASELINE sample_resp 5
! Run positive control if analyte binding low-----------------
IF sample_resp[Sample].RelResp < 20 THEN
    ┌-10 BASELINE Baseline2 5
    └    INJECT "Control" 20
    ┌    INJECT END
    └ 10 RESPONSE Control 5
    ! Terminate run if positive control doesn't bind--------
    IF Control[Sample].RelResp < 200 THEN
        EXIT METHOD
    END IF
END IF
     QUICKINJECT "Regeneration" 20
     QUICKINJECT END
```

7. Maintenance

For your own safety:

- Always wear gloves and protective clothing during operation and maintenance.
- Use required safety equipment when handling hazardous substances.
- If the instrument is contaminated with biohazards, please contact your local Biacore Service Support for further information about decontamination procedures.

Regular maintenance procedures are described in this chapter. If more extensive maintenance or service of the instrument is required, please contact your local Biacore Service Support.

7.1 Regular maintenance

Regular maintenance of BIACORE 2000 is essential for reliable results. It is important to keep the instrument free from contamination such as microbial growth and adsorbed proteins in the liquid handling system. The BIAmaintenance Kit recommended for BIACORE 2000 includes all solutions (except 70% ethanol) required for maintenance and system tests.

Solutions in BIAmaintenance Kit

BIAdesorb solution 1	0.5% (w/v) SDS in water
BIAdesorb solution 2	50 mM glycine-NaOH pH 9.5
BIApreservation solution	0.25% (w/v) diazolidinyl urea, 0.005% Surfactant P20 in water
BIAdisinfectant solution	1% sodium hypochlorite
BIAnormalizing solution	70% (w/w) glycerol in deionized water
BIAtest solution	15% (w/w) sucrose in HBS-EP buffer
HBS-EP buffer	

All solutions are filtered through a 0.22 µm filter.

Note: If you prepare your own solutions for maintenance, be particularly careful about the concentrations of glycerol and sucrose in the solutions for normalization and system test respectively. Correct concentrations of these solutions are critical for the maintenance procedures.

7.1.1 At the end of the day

At the end of a day's work, run the working tool **Standby** to keep buffer flowing at a low rate through the IFC. Do not leave the instrument overnight or longer with stationary buffer in the flow cells. The **Standby** tool may be appended to a method for automatic operation when the method finishes.

If you intend to run **Standby** for an extended period, make sure that there is sufficient buffer in the buffer pack. **Standby** will run for up to 4 days, which requires about 80 ml buffer. If the system is to be left for more than 4 days, run **Close** or **Shutdown** instead of **Standby** (see Section 7.2.1).

Replace the running buffer with water if your buffer tends to precipitate on standing. In this context, remember that the ambient temperature may drop during the night.

7.1.2 System cleaning

Removing adsorbed proteins
Inclusion of 0.005% Surfactant P20 to sample and running buffer is recommended to minimize non-specific adsorption of material to the pump tubing and IFC channels. Run the working tool **Desorb** to clean the tubing and IFC channels at least once a week, or more often if you do not use surfactant in sample and running buffer.

For some applications, a coating of protein on the tubing and IFC channel walls helps to give reproducible results, and the first few runs after **Desorb** may be unreliable. If you observe this behaviour, run **Desorb** when you change your protein system rather than once a week.

The recommended cleaning agent for **Desorb** is BIAdesorb Solution (0.5% SDS) from BIAmaintenance Kit. If you wish to use other cleaning agents, consult the chemical resistance tables in Appendix C.

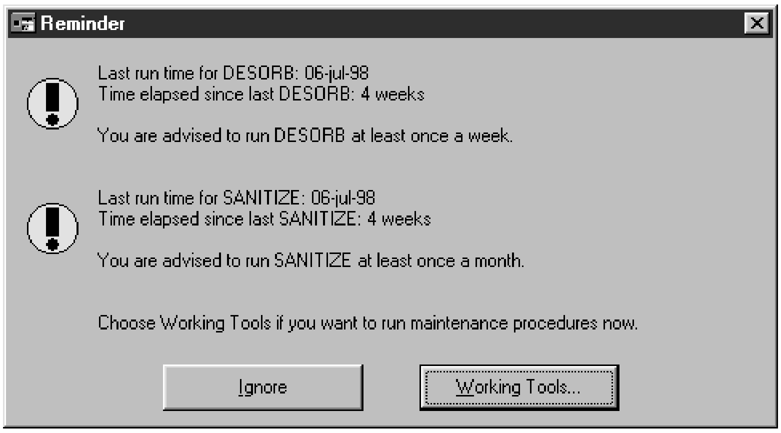
Removing salt residues
Many buffers, including HBS, use physiological salt concentrations (typically 150 mM NaCl). To prevent salt deposits accumulating in the liquid handling system, wash the recovery block and the needle with deionized water once a week. (Wipe the outside of the needle with a wet tissue and run **Command:Wash** with selection **Needle** to wash the inside.)

Preventing microbial growth
To prevent microbial growth in the instrument, keep a high standard of hygiene in the solutions used. Store buffers frozen or in a refrigerator and filter through a 0.22 µm filter before use.

Run the working tool **Sanitize** at least once a month to remove microbial contaminants. For best results, run **Desorb** before **Sanitize**.

Note: The sensor chip surface may be damaged by the **Desorb** and **Sanitize** procedures. You may keep a special chip for running these tools, but you should not use it for any other purpose.

A scheduler will remind you that it is time to run the working tools **Sanitize** and/or **Desorb** when more than the recommended interval has elapsed since the respective maintenance tool was last used. You may choose to run the maintenance tool or ignore the reminder and run the tool at a later date. If you ignore the reminder, it will be repeated after one week.



Last run time for DESORB: 06-jul-98
Time elapsed since last DESORB: 4 weeks

You are advised to run DESORB at least once a week.

Last run time for SANITIZE: 06-jul-98
Time elapsed since last SANITIZE: 4 weeks

You are advised to run SANITIZE at least once a month.

Choose Working Tools if you want to run maintenance procedures now.

7.1.3 Other routines

Inspect all visible liquid connections, pump syringes and valves, injection port and connector block for leaks each day. If any leaks are discovered, clean and tighten the connections or if necessary replace seals.

If deterioration in performance is suspected, run **Test Tools:System check** to check the system performance as outlined in Section 7.2.2.

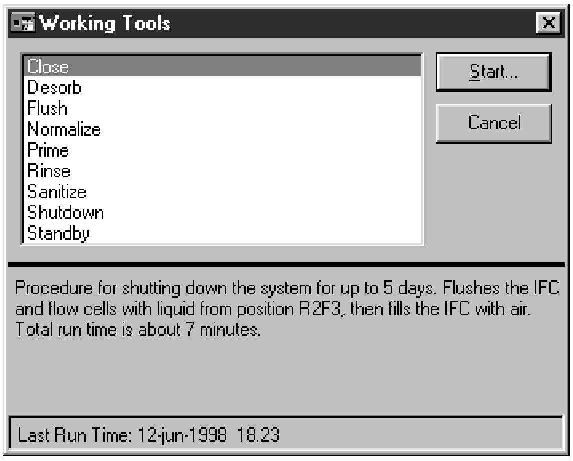
7.2 User tools

All user tools are available from the **Tools** menu. They are organized under three commands, for working tools, test tools and service tools respectively. To run a user tool, choose the appropriate command, select the user tool and follow the instructions on the screen.

The tool choice dialogue box includes short information to identify the highlighted tool. The date on which the selected tool was last run is shown at the bottom of the dialogue box.

7.2.1 Working tools

The working tools are a set of routines for starting up and shutting down the system and for special cleaning procedures.



Procedure for shutting down the system for up to 5 days. Flushes the IFC and flow cells with liquid from position R2F3, then fills the IFC with air. Total run time is about 7 minutes.

Last Run Time: 12-jun-1998 18.23

Close

A procedure for shutting down the system for up to 5 days. The procedure flushes the IFC and flow cells with liquid from a fixed autosampler position (r2f3), then empties the IFC of liquid. Approximately 2.5 ml liquid is required. Deionized or distilled water is recommended as flushing liquid. Total run time is about 11 minutes.

Required solutions: Deionized water.

Desorb

Removes adsorbed proteins from the autosampler and the IFC by flushing with SDS. As a general recommendation, run this procedure once a week. For some applications, a coating of protein on the tubing and IFC channel walls helps to give reproducible results, and the first few runs after **Desorb** may be unreliable. If you observe this behaviour, run **Desorb** when you change your protein system rather than once a week. Total run time is about 22 minutes.

A scheduler will issue a reminder at the beginning of the week if more than one week has elapsed since **Desorb** was last run. If the system is busy or switched off at the scheduled reminder time, the reminder will be displayed at the next possible opportunity (when the system is connected to the PC and is not busy).

Notes: The sensor chip surface may be damaged by the **Desorb** procedure. You may keep a special chip for running **Desorb**, but you should not use it for any other purpose. Rinse the chip after each usage and make sure it is clean before re-inserting it (see BIAtechnology Note 104).

BIAdesorb solutions are available from Biacore AB as part of the BIAmaintenance Kit. Do not run the procedure below 20°C, since this solution contains SDS which may precipitate at lower temperatures.

Required solutions: BIAdesorb solution 1.
BIAdesorb solution 2.

Flush

Flushes the IFC and flow cells with buffer. Use this procedure for fast cleaning of the IFC and flow cells. Total run time is about 2.5 minutes. **Flush** may be run between cycles in a method (method command `FLUSH`).

Required solutions: Running buffer.

Normalize

Normalizes the resonance signal in all flow cells to compensate for small differences between individual sensor chips. Use this procedure a) when the opto-interface is changed, b) when the temperature is changed, and c) when a new sensor chip is docked and maximum sensitivity is required. Total run time is about 4 minutes. **Normalize** may be run between cycles in a method (method command `NORMALIZE` with a parameter for the normalization solution rack position).

Note: If **Normalize** is included in a method and the normalization fails, the method will terminate.

The concentration of glycerol used for normalization is critical. BIAnormalizing solution is available from Biacore AB as part of the BIAmaintenance Kit. If you cannot perform a successful normalization, the opto-interface or sensor chip may be defective.

Required solutions: BIAnormalizing solution.

Prime

Primes the liquid system by flushing pumps, IFC and autosampler with buffer. Use this procedure at start up and when the buffer is changed, and also to remove small air bubbles which may accumulate during operation at low flow rates for long periods. Total run time is about 6 minutes. **Prime** may be run between cycles in a method (method command `PRIME`).

Required solutions: Running buffer.

Rinse

Washes the injection port thoroughly, then rinses the IFC and flow cells with buffer. Use this procedure to clean the IFC, flow cells and recovery cup. Total run time is about 3 minutes. **Rinse** may be run between cycles in a method (method command `RINSE`).

Required solutions: Running buffer.

Sanitize

Cleans pumps, IFC and autosampler from micro-organisms. Run this procedure once a month. For best results, run **Desorb** beforehand. Total run time is about 45 minutes.

A scheduler will issue a reminder at the beginning of the week if more than four weeks have elapsed since **Sanitize** was last run. If the system is busy or switched off at the scheduled reminder time, the reminder will be displayed at the next possible opportunity (when the system is connected to the PC and is not busy).

Note: The sensor chip surface may be damaged by the **Sanitize** process. You may keep a special chip for running **Sanitize**, but you should not use it for any other purpose. Rinse the chip after each usage and make sure it is clean before re-inserting it (see BIAtechnology Note 104).

Required solutions: BIAdisinfectant solution.
BIApreservation solution.
(or 0.005% Surfactant P20 in deionized water).

Shutdown

A procedure for shutting down the system for more than 5 days. The procedure flushes the IFC and flow cells with water, empties the IFC of liquid and then flushes the pumps and tubing with 70% ethanol. Total run time is about 10 minutes.

> *IMPORTANT*
> Follow the instructions on the screen carefully. The IFC does not withstand exposure to 70% ethanol, and must be disconnected before the pumps are flushed with ethanol.

Required solutions: Deionized water.
70% (v/v) ethanol.

Standby

This procedure maintains a low flow of buffer through the IFC and flow cells for up to 4 days. Buffer consumption is a pproximately 20 ml/day. The procedure uses both continuous flow and autosampler pumps to ensure a clean system. **Standby** may be appended to a method (method command `APPEND STANDBY`).

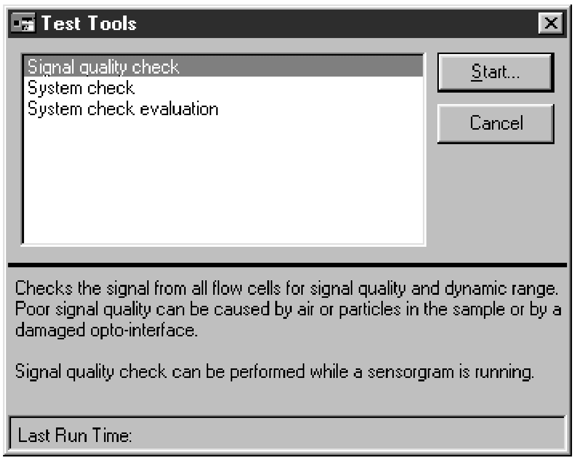
Required solutions: Running buffer.

Note: There are three tools for flushing the system with buffer. The essential differences between these are summarized below.

Tool	*Description*	*Time*
Flush	Flushes the continuous flow pump and IFC briefly at a moderate flow rate. Use **Flush** if you want to clean the IFC rapidly and do not have problems with air bubbles.	2.5 min
Rinse	Flushes pumps and IFC channels at a high flow rate and washes the recovery cup. The high flow rate helps to remove air bubbles that may be trapped in the system. Use **Rinse** regularly if you run at a low flow rate or elevated temperature where air bubbles may be a problem.	3 min
Prime	Washes the pumps, autosampler tubing and IFC with buffer. Use **Prime** when you want to change the buffer in the pumps.	6 min

7.2.2 Test tools

The test tools are a set of routines for testing the system performance.



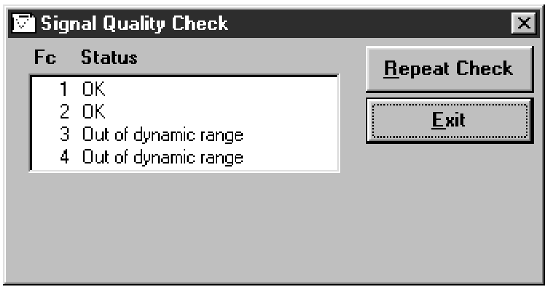
Checks the signal from all flow cells for signal quality and dynamic range. Poor signal quality can be caused by air or particles in the sample or by a damaged opto-interface.

Signal quality check can be performed while a sensorgram is running.

Last Run Time:

Signal quality check

This procedure checks the signal from the flow cells. The test finishes by displaying the flow cell status.



Fc	Status
1	OK
2	OK
3	Out of dynamic range
4	Out of dynamic range

If signal quality is acceptable, the status is “OK”. If the status is “Out of dynamic range” or “Poor signal quality” there may be air, particles or a high refractive index solution (e.g. glycerol) in the IFC channels, or the opto-interface may be bad.

If the signal quality is not acceptable, try running **Working Tools:Prime** once, to run buffer through the IFC. If this doesn’t help, run **Normalize** once. If signal quality still hasn’t improved, contact Biacore local service support.

System check

This procedure performs a comprehensive check of system performance, using a standard sucrose solution which is provided in the BIAmaintenance kit from Biacore AB. Total run time is about 40 minutes.

You should run the procedure at 25°C with a Sensor Chip CM5 reserved for this purpose, on which no ligand has been immobilized. For maximum reliability we recommend that you use a new Sensor Chip CM5.

Required solutions:	BIAtest solution
	HBS buffer, BIA *Certified*

The sensorgram resulting from system check is displayed in an untitled sensorgram window. You may evaluate the unsaved sensorgram directly and/or save the sensorgram with **File:Save** for later inspection.

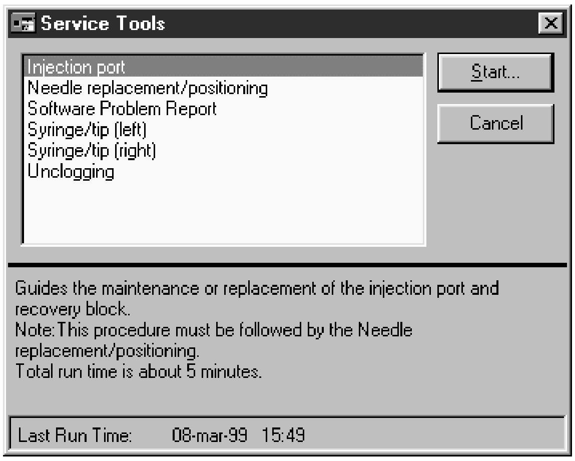
System check evaluation

This procedure generates a report on the saved results of system check. Choose the system check result file in the **Open** dialogue box. The evaluation protocol (2 pages) is displayed on the screen with scrolling facilities, and can be printed with the **Print** button. The protocol includes instructions for assessing the system check results.

Note: Unsaved system check results can be evaluated with the same procedure by choosing **Evaluate** when the system check procedure is complete.

7.2.3 Service tools

The service tools are a set of routines for system maintenance.



Guides the maintenance or replacement of the injection port and recovery block.
Note: This procedure must be followed by the Needle replacement/positioning.
Total run time is about 5 minutes.

Last Run Time: 08-mar-99 15:49

Injection port
A guide to the maintenance or replacement of the injection port and connector block.

Note: This procedure must be followed by a needle positioning test (**Needle replacement/positioning**).

Needle replacement/positioning
A guide to replacement of the autosampler needle and needle positioning test. Needle positioning can be checked without needle replacement.

Software problem report
A utility to help in formulating a problem report in the event of malfunction. If possible, run this utility on the PC controlling the system and allow the utility to collect computer system information automatically. The report generated by this utility may be submitted by e-mail or printed out and faxed to Biacore.

Syringe/tip (left)
A guide to replacing the syringe tip on the left-hand pump.

Syringe/tip (right)
A guide to replacing the syringe tip on the right-hand pump.

Unclogging
Clears partially blocked IFC flow cells and loops by flushing with buffer at a high flow rate. Total run time is about 4 minutes. **Unclogging** may be run between cycles in a method (method command `UNCLOG`).

Required solutions: Running buffer.

8. Troubleshooting

If you have problems with BIACORE 2000 or with your results:

As a first measure	Run **Test Tools:System check** to verify system performance. Remember to dock an unused sensor chip before running this procedure. If the system check results do not indicate any instrument malfunction, check your method and reagents carefully for possible sources of error.
If you obtain unexpected or unacceptable sensorgram results	Read Section 8.1.
If you suspect a specific instrument malfunction	Read the relevant part(s) of Section 8.2.
If you have problems with BIACORE Control Software	Read Section 8.3.
If you cannot identify or correct problems with the help of the information here	Call your Biacore service representative.

8.1 Sensorgram problems

8.1.1 Signal noise

Normal signal noise levels are <0.3 RU RMS (< 1 RU peak-to-peak) at data collection rate 1 Hz. Excessive noise may be caused by:

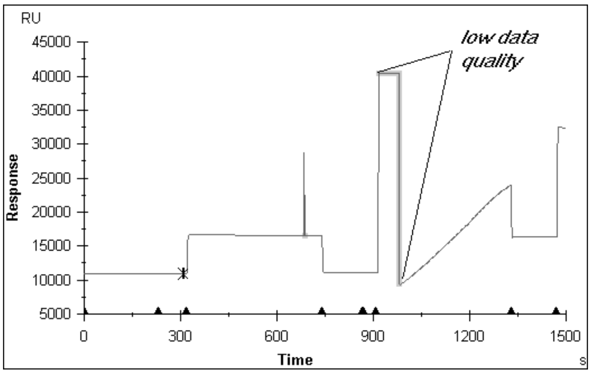
- air or particles in the sample or buffer. Degas and filter solutions. Run **Working Tools:Prime** to remove air which may accumulate in the flow cell channels.

- mechanical vibration. The instrument should be placed on a firm bench. Do not place heavily vibrating equipment (e.g. an unbalanced bench centrifuge) on the same bench as BIACORE 2000.

- electrical noise. Do not place the instrument in an area subject to strong electrical or magnetic fields.

8.1.2 Low data quality

Low data quality is indicated by a thick line on the sensorgram and an entry in the extended event log (choose **View:Extended Event Log**).



Low data quality is normal if the signal rises above about 40,000 RU (corresponding to a bulk refractive index of >1.36). This may be caused by injection of air which shifts the baseline so that the signal is outside the dynamic range of the instrument. Low data quality below 80,000 RU may be caused by aggregation of material on the sensor chip surface. If low data quality is reported in other circumstances, dock a new sensor chip and run **Working Tools:Normalize**. If the problem persists, call Biacore service.

Note: The marking for low data quality on the sensorgram extends from the first low quality point to the first normal quality point. If the response curve is very steep, this may sometimes give the impression that the data quality is low even at normal response levels (see the illustration above). The event log records start of low data quality and the return to normal data quality correctly.

8.1.3 Spikes

Spikes in the sensorgram may be caused by:

- small air bubbles in the flow cell. These may appear anywhere in the sensorgram.
- transient matrix effects (see Section B.1) or conformational changes in the immobilized ligand. These appear at the beginning and/or end of sample injection.

Note: As a result of the way data is displayed on the screen, spikes that are represented by only one data point may disappear at certain zoom factors. The same effect may occasionally be seen if the sensorgram is printed at draft quality.

8.1.4 Drift

Signal drift may be caused by sample dispersion, carry-over of sample with a different bulk refractive index, matrix effects or conformational changes in the immobilized ligand. To minimize matrix effects, sample composition should be as close as possible to running buffer. Long-term drift may also result from unstable temperature at the flow cell.

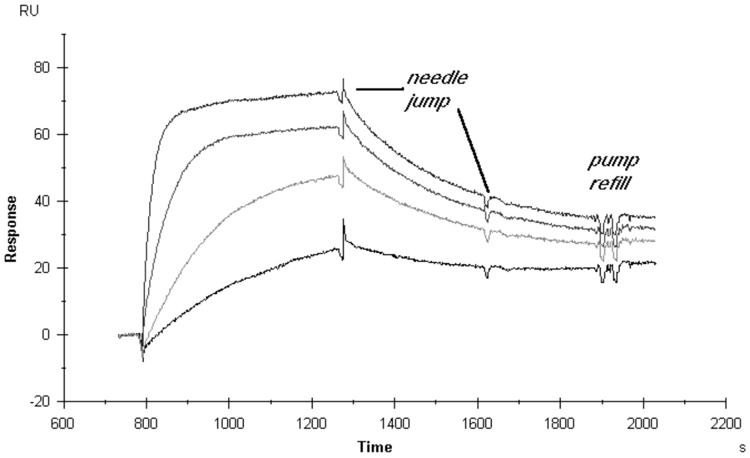
Drift may also be caused by small air bubbles developing in the flow cell and growing in size. If the air bubble breaks free during buffer or sample flow, the drift will suddenly cease (see ***Shifts*** below).

Release of compounds that have permeated the silicone walls of the IFC during a previous injection (e.g. regeneration with formic acid) can also cause drift effects. If you suspect that this is happening, wash the IFC and run **Working Tools:Prime**.

Use an in-line reference cell where possible to correct for systematic drift effects.

8.1.5 Reproducible disturbances

Flow system events such as releasing the needle from the injection port, refilling the pump, etc can cause small disturbances (≈5 RU or less) in sensorgrams. Typically, these events affect all four sensorgrams at the same time in multichannel analyses. These disturbances may be noticeable in very high performance analyses, and may often be corrected by subtracting the signal from an in-line reference cell.



Small disturbances of this kind which are reproducibly associated with system events should not be regarded as a malfunction: they are rather a reflection of the very high sensitivity of the instrument.

8.1.6 Shifts

Shifts in signal baseline may be caused by small air bubbles developing in the flow cell and then breaking free into the buffer flow. See Section 8.2.2 for air bubble considerations.

Mechanical instability can also cause shifts in baseline.

8.1.7 No response on injection

Check the detector by injection of 15% sucrose or glycerol in water in multichannel mode. Choose **View:Scale** and set the response scale (y-axis) to **Auto**. If there is no response, contact Biacore service.

Note: If you obtain no response on injection in a method in single channel mode, check that you have not used a DETECTION command in MAIN and a FLOWPATH command in APROG with different flow cell numbers. See Section E.4.1.

8.1.8 Poor reproducibility

If repeated injections of the same sample do not give the same response:

- If the variability is systematic with increasing response in later injections, use septa-covered vials to eliminate sample evaporation. Small changes in concentration as a result of evaporation from open vials can give noticeable variations in response.

- If the variability is systematic with progressively decreasing response in later injections, there may be carry-over of running buffer to the sample vial in the autosampler. Pre-dip the needle in a vial with sample or sample buffer to minimize carry-over effects (see Appendix B). Alternatively, if the samples are cooled, condensation on the inside of the vials may dilute the samples.

- Variations between the first few injections in a series followed by consistent results in the rest of the series may indicate carry-over of material or memory effects from earlier runs. Make sure that the liquid handling system is properly cleaned between runs (see Appendix B) and is equilibrated with the running buffer for the present run (run **Working Tools:Prime** and allow buffer flow for 5-10 minutes before starting the first cycle). If this does not help, run a few dummy cycles with sample injection and regeneration to condition the surface.

- If the variability is random, there may be a fault in the liquid handling system (see Section 8.2.2). Repeated injections performed under manual control may show more variation than repeated method cycles, unless care is taken to use exactly the same conditions for each injection.

8.1.9 Sample dispersion

See Appendix B for a discussion of sample dispersion at the beginning and end of injections. Normal dispersion levels are represented by a response change of 99% within about 0.5-2 µl after the start and end of the injection depending on the number of flow cells in the flow path. If dispersion is excessive, check the air segments separating sample from buffer in the autosampler tubing.

Undue dispersion in method-controlled runs may be caused by early preparation of a timed injection. See Appendix B.

8.1.10 Other application problems

To avoid problems with your applications (i.e. the system functions correctly but your application “doesn’t work”), read the BIAapplications Handbook for general aspects and Appendix B of this manual for aspects specific to BIACORE 2000.

8.2 Instrument problems

8.2.1 General failure

If the red ERROR indicator lights on the front panel of the instrument, switch the instrument and PC off, wait 1 minute, then restart the system. If the problem persists, call Biacore service.

8.2.2 Liquid delivery

Pump functions

Faulty liquid handling may cause a variety of problems including incorrect injection volumes, delayed injections, broken air segments during injection, incorrect dilutions and transferred volumes etc. Test liquid handling with injections of sucrose or glycerol solutions which give a well-defined refractive index pulse. Test autosampler functions (dilute, transfer and mix) by preparing defined mixtures of sucrose or glycerol solution and buffer. The plateau response during injection should be directly proportional to the concentration.

Faulty liquid handling can affect **Dilute**, **Transfer**, **Mix**, **Dipneedle**, **Inject** and other injection commands), **Fractions** and **Recover**, and may be caused by:

- incorrect volumes in command. Check the values in the manual dialogue box or the method.

- problems with air segments (see below).

- incorrect needle positioning (see Section 8.2.3).

- leakage in tubing connections. For correct injections, there should be a delay between the start of the injection as recorded in the event log and the beginning of the response in the first flow cell corresponding to not more than 0.5 µl liquid (e.g. 1 second at 30 µl/min). The delay between responses in multichannel mode should correspond to about 1 µl liquid. Longer delays probably indicate leakage.

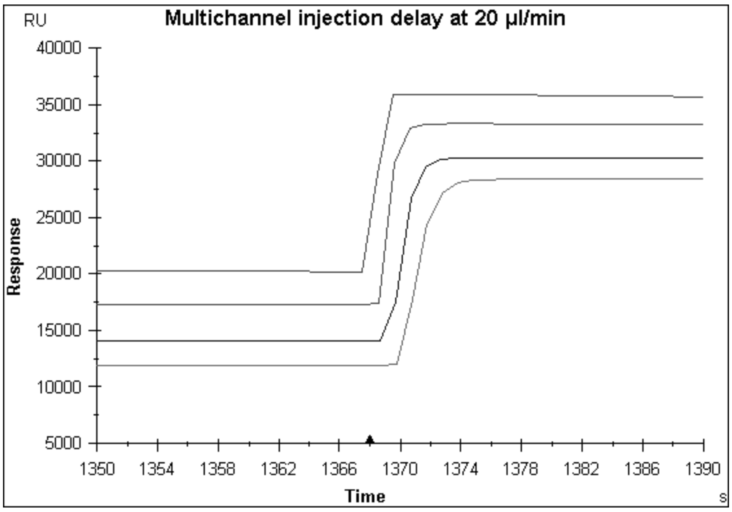


Illustration of the normal delay between sensorgrams in multichannel mode. Note the enlarged time axis.

- pump failure. Contact Biacore service if the pumps do not operate.

- formation of air bubbles in solutions.

- flow obstruction. Check that the tubing is not sharply bent or trapped so as to obstruct liquid flow. Replace any damaged tubing.

- air in pump syringe. Examine the pump for air bubbles in the syringe. To remove air bubbles, tap the syringe gently during a **Prime** operation.

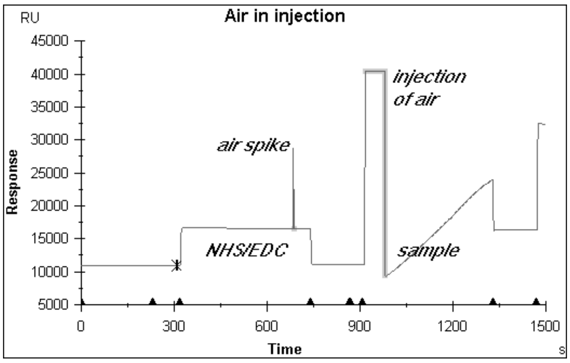
Air in sample or buffer
Small air bubbles developing in sample or buffer can give rise to a noisy signal or signal spikes. Air bubbles which form in the flow cell and grow in size cause the signal to drift. Air bubble formation can be particularly apparent during long runs (>5 hours), at elevated temperatures and at low flow rates (<10 µl/min). To minimize problems with air bubble formation:

- degas buffer thoroughly at temperatures above the running temperature or open a new pack of room-tempered HBS buffer *BIA Certified*.
- for analyses above ambient temperature, degas the buffer at a temperature higher than the analysis temperature. Do not degas for long periods at elevated temperatures, since this may cause significant evaporation of water from the buffer.
- check fittings and tubing connections for leakage.
- run **Working Tools:Prime** at regular intervals (between each cycle in a method).
- increase the flow rate.

For air bubbles injected together with the sample, see Section 8.2.3.

8.2.3 Autosampler

Irreproducible air segments in liquid handling
If the air segments separating sample from buffer in the autosampler tubing are too large or appear to break up, air may be injected over the sensor surface giving a high signal outside the dynamic range of the instrument. In other autosampler operations such as **Dilute**, **Fractions**, **Recover**) faulty air segments can lead to delivery of incorrect volumes.



Sensorgram from amine coupling with air spike in the NHS/EDC injection and air bubble at the beginning of the sample injection.

If this happens:

- check that there is sufficient sample in the vial and that the autosampler needle is correctly positioned during sample aspiration.
- check that the autosampler tubing is clean and not damaged. Replace if necessary.
- bear in mind that air bubbles can stick to the walls of plastic vials and can be aspirated together with sample.

Note: Some reagents, including NHS/EDC (amine coupling reagent) and cyanoborohydride (used in aldehyde coupling) generate gas in small quantities. Gas bubbles may often form during injection of such reagents.

Bad needle positioning

If the needle positioning is unsatisfactory for the injection port, wash station, recovery cup or rack positions (i.e. too high, too low or displaced to the side):

1. Check that the autosampler rack is correctly seated on the rack base.
2. Run **Service Tools:Needle replacement/positioning**.

If the needle moves to an invalid rack position, check that the rack used in the autosampler is correctly defined in the control software (**Command:Define Rack Base**).

Samples are warm

The temperature in the autosampler compartment should not be more than 5-6°C above ambient. If lower temperatures are required for sample stability, connect coolant to the rack base using the tubing connectors on the side of the instrument.

The temperature in the autosampler compartment may rise if there is insufficient air circulation around the instrument. Do not place the instrument directly under a shelf. If air circulation is adequate and the temperature in the autosampler compartment is more than 5-6°C above ambient, contact Biacore service.

Note: The temperature in the autosampler compartment does not affect the temperature at the sensor chip surface. Samples equilibrate to the detector temperature during passage through the IFC.

8.2.4 Connector block and IFC

Leakage at the injection port
Clean or replace the injection port on the connector block (**Service Tools:Injection Port**).

IFC valve malfunction
The IFC valves may stick if they are left closed for long periods of time. Run **Service Tools:Unclogging** followed by **Working Tools:Prime** if you suspect that valves are not functioning properly.

The valves are left open in the normal stand-by state (when no sensorgram is running). The working tool **Standby** switches valves at intervals to prevent sticking.

8.2.5 Sensor chip

If you cannot dock or undock a sensor chip, switch the instrument off and re-start. If the problem persists, contact Biacore service.

Remember that the commands **Dock** and **Undock** are not available during a run (including during operation of **Standby**).

8.2.6 Temperature control

Adequate circulation of air around the instrument is important for correct temperature control. There should be at least 20 cm clearance behind and above the instrument. Make sure that the fan inlet (underneath the instrument at the front left-hand side) and outlet (on the back panel at the left-hand side) are not obstructed. Do not put papers under the instrument, particularly at the left-hand side. Paper is easily sucked against the fan inlet, blocking the air intake. Make sure that the space between the bottom of the instrument and the bench is not obstructed by laboratory equipment, notebooks etc.

Unable to reach set temperature
The cooling unit will not maintain temperatures more than 20°C below ambient. With this restriction, if the set temperature is not reached within a reasonable time (heating approximately 1 hour/5°C, cooling somewhat longer), call Biacore service.

Unstable temperature
If the temperature is not stable during a run (the temperature LED on the front panel flashes), check first that the ambient temperature is not subject to large variations (e.g. open windows, direct sunlight etc). The temperature control in BIACORE 2000 may not compensate for

sudden changes in ambient temperature (> 5°C over less than about 15 minutes).

An increase in temperature will give a decrease in absolute response.

To check the temperature during a run:

- check the temperature at the beginning and end of each cycle as recorded in the event log.

- use the `CURVE temp` command in the method to record the temperature throughout the run.

If a constant temperature is not maintained in spite of reasonably constant ambient conditions, the thermostatting unit may be defective. Contact Biacore service.

8.3 Software problems

No contact with instrument
The instance of BIACORE Control Software which has contact with the instrument is identified as **BIACORE 2000 on COM1:** in the title bar of the window. If you cannot establish contact with the instrument:

- Check that the cable connecting the PC to the instrument is properly seated. Check also that the cable is connected to the right port on the PC and the instrument (see Chapter 3).

- Check that you do not have another instance of BIACORE Control Software running. Close all instances of the control software and restart it from the Windows **Start** menu (do not start by double-clicking on a BIA file in Windows Explorer: control software started in this way will not connect to the instrument).

- Close BIACORE Control Software and shut down the computer. Turn off the power to BIACORE 2000 and then restart the system.

File changes not saved
With two instances of BIACORE Control Software open, you can open the same file in both instances. If you make changes to the file (e.g. edit a method or add report points to a sensorgram) in one instance and save the changes, then save the file again from the second control software instance, the changes will be lost. This is because the control software works with a copy of the file contents in memory, and changes in one instance do not affect the working copy in another.

Other problems

For problems relating to the computer hardware or Windows operation, refer to the respective product documentation.

For problems with BIACORE Control Software which do not have a readily apparent solution:

1. If possible, perform tests to determine whether the problem is reproducible.

2. Run the **Software Problem Report** in **Service Tools** (Section 7.2.3) to create a report including system-specific information. In the report, the nature of the problem and how you can reproduce it (or alternatively what you were doing when it arose) *in as much detail as possible*, and submit the report by e-mail or fax to Biacore (the address is provided in the **Software Problem Report** utility).

   If the software malfunction prevents you from running the **Software Problem Report** from **Service Tools**, insert the Biacore Software CD in your CD drive and right-click on the drive icon in Windows Explorer. The software problem report can be accessed from the context menu for the CD.

Errors which arise during a run are recorded in the event log for the run.

9. BIACORE Control Software reference

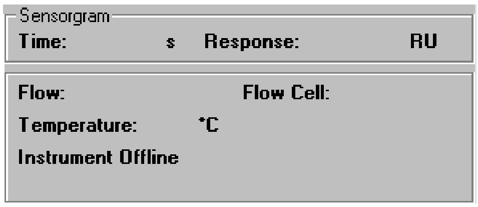
This chapter provides a full reference to the BIACORE Control Software user interface for BIACORE 2000, organized in accordance with the software menu structure. See Chapter 4 for an operational description of how to use the software.

9.1 Starting BIACORE Control Software

9.1.1 Starting from the Windows Start menu

A shortcut to the BIACORE Control Software is installed under BIA **Programs** on the **Start** menu.

It does not matter whether you switch on the instrument or start the software first. If you start the software first, the status window will display the message **Instrument Offline**. Connection will be established when the instrument is switched on.



Sensorgram

Time: s Response: RU

Flow: Flow Cell:

Temperature: °C

Instrument Offline

Depending on the current instrument status when the software is started, you will be prompted for the appropriate set-up action (e.g. dock a sensor chip, prime the system). See Section 4.2 for further details of these actions.

9.1.2 Starting from Windows Explorer

BIACORE Control Software is associated in Windows Explorer with file extensions .blr, .blm and .blw, so that double-clicking on files of these types in Explorer will start the Control Software. BIACORE Control Software started in this way however will not be connected to the instrument, and can only be used for writing and editing methods and examining result files.

9.1.3 Multiple software instances

You can start several instances of BIACORE Control Software, but only the first instance can be connected to the instrument. If you start multiple instances, you will be warned that the software is already running.

The control software instance that is connected to the instrument is identified as **BIACORE Control Software on COM1** in the window title bar (and in the icon title if the window is reduced to an icon on the desktop). Other instances are identified simply as **BIACORE Control Software**.

Note that only one instance of the software can be connected to the instrument.

The number of instances you can open depends on the Windows system resources available in your computer. The following error messages appearing at start-up may indicate that system resources are insufficient:

- **Out of memory**
- **ConnectToBIACORE, unexpected error=20100**
- **Could not start communication server**

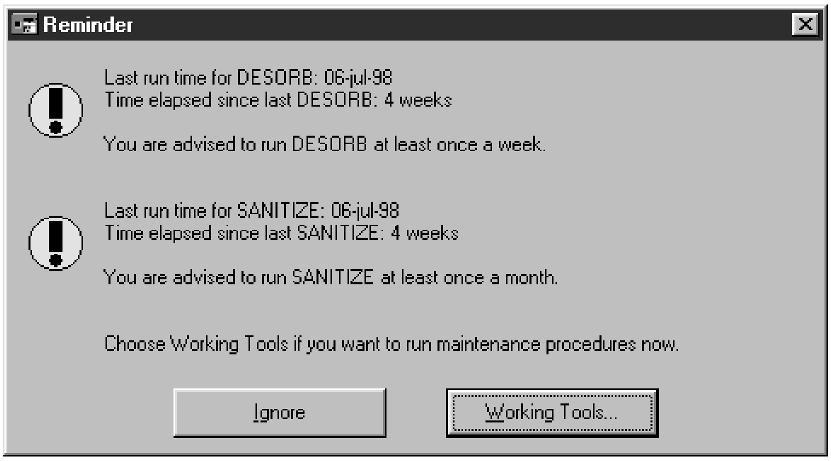
If you receive one of these messages, close some other applications and try again.

9.1.4 Recovering lost data

A message **Lost data found** displayed when you start the control software indicates that a previous run was not correctly saved (this can happed for example if there is a power failure during a run). If you choose to recover the lost data, the data will appear in a sensorgram window and can be saved as a normal result file. If you choose not to recover the data when the control software starts, the lost data will be deleted permanently and cannot be recovered at a later date.

9.1.5 Start-up routines

At start-up, the system may display a reminder that it is time to run the working tools **Sanitize** and/or **Desorb** (see Section 7.1.2). This reminder appears when more than the recommended interval has elapsed since the respective tool was last used. You may choose to run the maintenance tool or ignore the reminder and run the tool at a later date. If you ignore the reminder, it will be repeated after one week.



Last run time for DESORB: 06-jul-98
Time elapsed since last DESORB: 4 weeks

You are advised to run DESORB at least once a week.

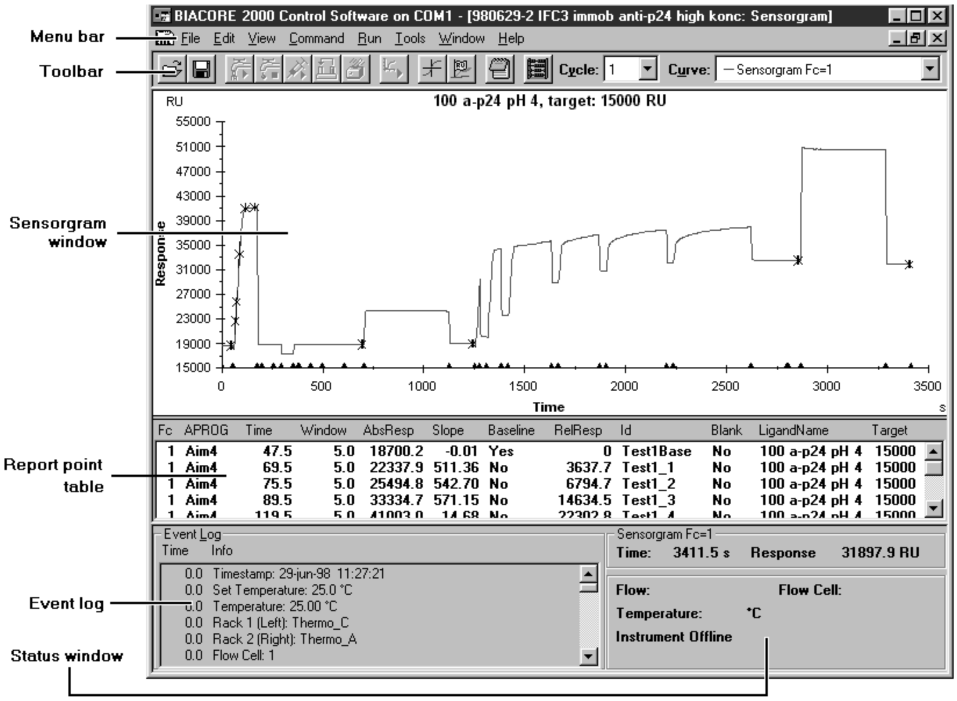
Last run time for SANITIZE: 06-jul-98
Time elapsed since last SANITIZE: 4 weeks

You are advised to run SANITIZE at least once a month.

Choose Working Tools if you want to run maintenance procedures now.

9.2 Screen display

The BIACORE 2000 screen is divided into six main areas:



Fc	APROG	Time	Window	AbsResp	Slope	Baseline	RelResp	Id	Blank	LigandName	Target
1	Aim4	47.5	5.0	18700.2	-0.01	Yes	0	Test1Base	No	100 a-p24 pH 4	15000
1	Aim4	69.5	5.0	22337.9	511.36	No	3637.7	Test1_1	No	100 a-p24 pH 4	15000
1	Aim4	75.5	5.0	25494.8	542.70	No	6794.7	Test1_2	No	100 a-p24 pH 4	15000
1	Aim4	89.5	5.0	33334.7	571.15	No	14634.5	Test1_3	No	100 a-p24 pH 4	15000
1	Aim4	119.5	5.0	41003.0	14.68	No	22302.8	Test1_4	No	100 a-p24 pH 4	15000

Event Log

```
Time   Info
  0.0  Timestamp: 29-jun-98  11:27:21
  0.0  Set Temperature: 25.0 °C
  0.0  Temperature: 25.00 °C
  0.0  Rack 1 (Left): Thermo_C
  0.0  Rack 2 (Right): Thermo_A
  0.0  Flow Cell: 1
```

Sensorgram Fc=1

Time: 3411.5 s Response 31897.9 RU

Flow: Flow Cell:

Temperature: °C

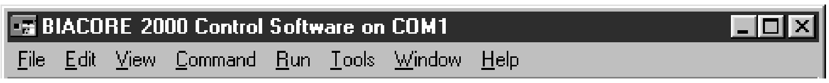
Instrument Offline

- The **menu bar** provides access to BIACORE 2000 menu commands. Many of the commands can also be accessed through shortcut keys (see Section 9.3).

- Buttons on the **toolbar** duplicate frequently used menu commands for more convenient access.

- The **sensorgram window** displays the progress of the interaction as a sensorgram. This area also displays the **method editor window**.
- The **report point table** records sensorgram information at each report point.
- The **event log window** records important parameters at the start and end of each run and at events during the run.
- The **status window** displays the current instrument status, including current response, continuous flow rate, temperature, data collection rate. System status messages are displayed at the bottom of this window.

9.2.1 Menu bar


BIACORE 2000 Control Software on COM1
File Edit View Command Run Tools Window Help

Menu commands follow Windows standard usage. Briefly:

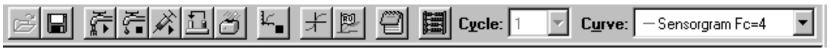
- To select a menu, click on the menu name in the menu bar or press **Alt** followed by the underlined letter in the name.
- To close a menu, click anywhere outside the menu or press **Esc**.
- To choose a command, select the menu then click on the command or press the underlined letter in the command. Alternatively, press the shortcut key or key combination (see Section 9.3). You do not need to select the menu in order to use the shortcut key.

Commands which are dimmed cannot be used in the current circumstances (e.g. you cannot start a sensorgram unless you have docked a sensor chip).

Some commands change according to the current circumstances (e.g. **Run Sensorgram** changes to **Stop Sensorgram** when a sensorgram is started).

9.2.2 Toolbar

The toolbar contains buttons for commonly used commands. Click on a toolbar button to choose the corresponding command. All commands are also available from the menus.



Cycle: 1

Curve: — Sensorgram Fc=4

Buttons which are dimmed cannot be used in the current circumstances (e.g. you cannot start a sensorgram unless you have docked a sensor chip).

The **Cycle** list box shows the current cycle during a method run and allows you to select which cycle to display. (Each cycle corresponds to an APROG in the method (see Section 6.3) and contains one, two or four sensorgrams depending on the detection mode.)

The **Curve** list box identifies and allows you to select the "active" curve in multichannel mode. Coordinates for the active curve are displayed in the status window. Changing the active curve does not affect data collection.

9.2.3 Sensorgram window

The sensorgram window displays the progress of the current analysis.

The display is automatically scaled by default to accommodate the sensorgram(s). Scaling can be set manually with **View:Scale**, or adjusted to fit the current sensorgram(s) with **View:Adjust Scale**. See the command descriptions in Section 9.6 for more details.

To zoom the display, drag the mouse marker over the area to be enlarged. To return the standard display, choose **View:Unzoom** or double-click with the left mouse button in the sensorgram window.

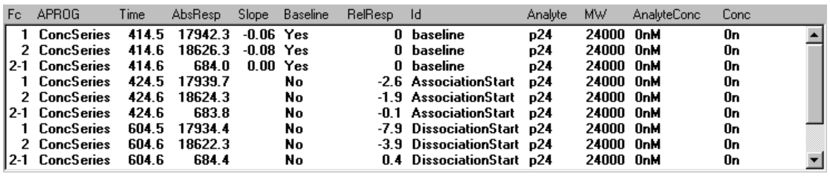
The sensorgram(s) may show event markers (triangles on x-axis), report points (crosses on sensorgram with the report window indicated) and baseline and/or reference line (horizontal and vertical lines respectively). See Section 9.6 for more details.

Fixed colours are used to distinguish the flow cell for each sensorgram:

Fc 1	red
Fc 2	green
Fc 3	blue
Fc 4	magenta
reference-subtracted	black

For best display of the sensorgram window, make sure that the window background colour is set to white (this is the normal Windows setting).

9.2.4 Report point table



Fc	APROG	Time	AbsResp	Slope	Baseline	RelResp	Id	Analyte	MW	AnalyteConc	Conc
1	ConcSeries	414.5	17942.3	-0.06	Yes	0	baseline	p24	24000	0nM	0n
2	ConcSeries	414.6	18626.3	-0.08	Yes	0	baseline	p24	24000	0nM	0n
2-1	ConcSeries	414.6	684.0	0.00	Yes	0	baseline	p24	24000	0nM	0n
1	ConcSeries	424.5	17939.7		No	-2.6	AssociationStart	p24	24000	0nM	0n
2	ConcSeries	424.6	18624.3		No	-1.9	AssociationStart	p24	24000	0nM	0n
2-1	ConcSeries	424.6	683.8		No	-0.1	AssociationStart	p24	24000	0nM	0n
1	ConcSeries	604.5	17934.4		No	-7.9	DissociationStart	p24	24000	0nM	0n
2	ConcSeries	604.6	18622.3		No	-3.9	DissociationStart	p24	24000	0nM	0n
2-1	ConcSeries	604.6	684.4		No	0.4	DissociationStart	p24	24000	0nM	0n

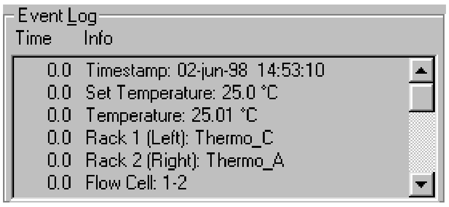
The report table records sensorgram information at each report point. For details on the type of information recorded, see Section 9.5.

9.2.5 Event log window

The event log window records all events in the current cycle. “Events” may be functional breakpoints such as injections or changes in the flow rate or status information such as time stamp and detector temperature. Note that events outside cycles in methods (i.e. in the `MAIN` block) will not be recorded in the event log.

The event log may be displayed in brief or extended mode. Choose **View:Extended Event Log** to toggle between these modes. In brief mode, only essential events and information are displayed. Additional information displayed in extended mode can be useful in troubleshooting situations. Messages which can appear in the event log are documented in Appendix D.

The event log starts when a sensorgram is started. The first entries in the log always record the date and time, the flow cell temperature at the start of the run, the initial flow path, flow rate and the rate of data collection.



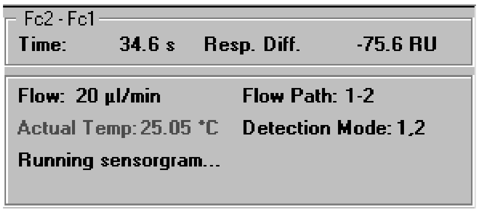
Time	Info
0.0	Timestamp: 02-jun-98 14:53:10
0.0	Set Temperature: 25.0 °C
0.0	Temperature: 25.01 °C
0.0	Rack 1 (Left): Thermo_C
0.0	Rack 2 (Right): Thermo_A
0.0	Flow Cell: 1-2

Use the scroll bar to scroll through the log.

If the reference line is displayed on the sensorgram display, you can click on a line in the event log to highlight the event and move the reference line to the corresponding event marker (provided that the event is within the current sensorgram display window).

9.2.6 Status window

The status window shows the current instrument status:



Fc2 - Fc1
Time: 34.6 s Resp. Diff. -75.6 RU

Flow: 20 µl/min Flow Path: 1-2
Actual Temp: 25.05 °C Detection Mode: 1,2
Running sensorgram...

- the elapsed time and response for the active sensorgram when a sensorgram(s) is running.

- the current flow rate and flow path.

- the flow cell temperature. Before the temperature at the flow cell is stable, the temperature is displayed in red text and toggles between preset and actual temperatures in red text. Once stable, the temperature at the flow cell is displayed in black text.

- the detection mode (during manual runs).

- system messages (documented in Appendix D).

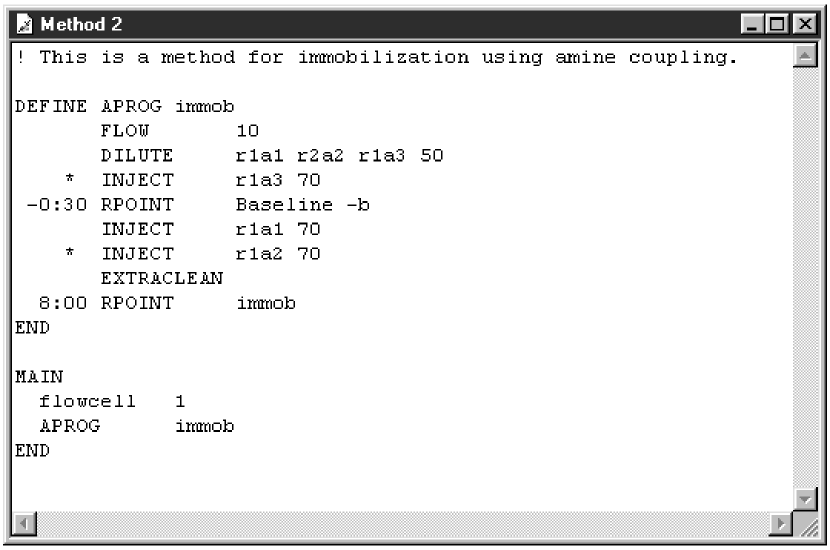
In addition:

- If the reference line is displayed, the elapsed time and absolute response at the reference line are shown for the active sensorgram (see **Curve** list on tool bar).

- If both reference line and baseline are displayed, the elapsed time and relative response (baseline = 0) at the reference line are shown for the active sensorgram.

9.2.7 Method editor window

The method editor is used for writing and editing methods. Open the method editor window with either **File:New Method** for a new method or **File:Open** for a saved method file. Multiple method editor windows can be open simultaneously.



```
! This is a method for immobilization using amine coupling.

DEFINE APROG immob
       FLOW       10
       DILUTE     r1a1 r2a2 r1a3 50
    *  INJECT     r1a3 70
 -0:30 RPOINT     Baseline -b
       INJECT     r1a1 70
    *  INJECT     r1a2 70
       EXTRACLEAN
  8:00 RPOINT     immob
END

MAIN
  flowcell   1
  APROG      immob
END
```

Note: The method editor cannot be used if a sensorgram is running. Start another instance of the control software if you want to edit methods while a sensorgram is running.

For a description of how to write methods with the method editor, refer to Appendix E.

9.3 Menus and buttons

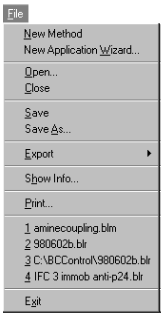
Several commands in BIACORE 2000 have corresponding buttons in the toolbar for quick access. All commands are available from the menus: many have in addition shortcut keys as an alternative (shortcut keys are shown alongside the commands in the menus where appropriate). The table on the next page lists toolbar buttons and shortcut keys for menu commands.

Toolbar button icons are displayed in some dialogue boxes (e.g. **Inject**) to help to identify the dialogue box. A special icon in some boxes (e.g. **Dock**) indicates that a manual operation is required:

🧍 Indicates manual operation.

Menu/Command	Toolbar button	Shortcut key
File		
Open	📂	Ctrl-O
Save	💾	Ctrl-S
Print		Ctrl-P
Edit (sensorgram)		
Copy Graph		Ctrl-H
Add Report Point	🏁	
Edit (method)		
Undo		Ctrl-Z
Cut		Ctrl-X
Copy		Ctrl-C
Paste		Ctrl-V
Find		Ctrl-F
Replace		Ctrl-R
View		
Unzoom		Ctrl-U
Reference Line	┼	F8
Baseline		F9
Notebook	📓	
Wizard Results		Ctrl-R
Command		
Inject/Stop Inject	▶ ■	
Transfer	⬚	
Mix	⬚	
Flow/Stop Flow	▶ ■	
Run		
Run/Stop Sensorgram	▶ ■	
Stop Sensorgram		
Tools		
Evaluation	▤	Ctrl-E

9.4 File menu



New Method

Opens a new, empty method editor window.

See Appendix E for a description of how to write methods in the method editor.

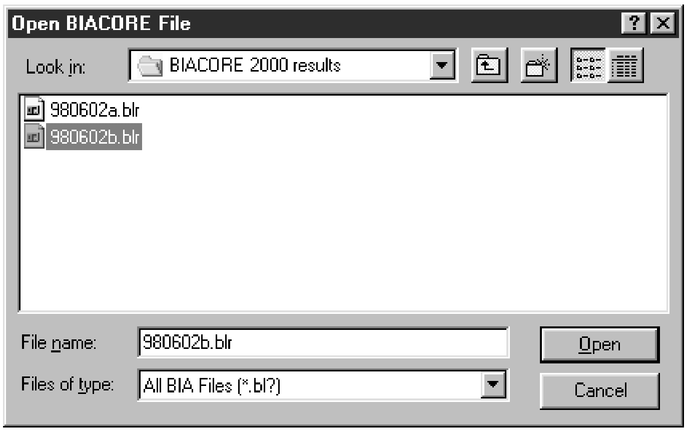
File:New Method is not available while a sensorgram is running.

New Application Wizard

Opens the Application Wizard dialogue (see Chapter 5) from which you can define a new wizard template. Use this option to define wizard templates when you do not have a connection to the instrument (e.g. in a second instance of the control software).

Open (ctrl-O)

Opens an existing method or result file.


Open BIACORE File

Look in: BIACORE 2000 results

File name: 980602b.blr

Files of type: All BIA Files (*.bl?)

Change the folder if necessary in the **Look in** field. Select the file to open or enter the file name in the **File Name** field.

Opening a result file (default extension .BLR) opens a sensorgram window containing the results. Only one sensorgram window can be open at once.

Opening a method file (default extension .BLM) opens a method editor window containing the method. Several method editor windows can be opened simultaneously (the maximum number depends on the available Windows system resources).

File:Open is not available while a sensorgram is running.

Close

Closes the active window (sensorgram or method editor). If the current contents of the window have not been saved, you will be given an opportunity to save the results or method.

File:Close is not available while a sensorgram is running.

Save (ctrl-S)

Saves the contents of the currently active window as a file. This option is equivalent to **File:Save As** when a file is saved for the first time.

The file is saved as a result or method file according to whether the current window is a sensorgram or method editor window respectively. A saved result file includes sensorgram data, report point table, event log, method definition and notebook. A saved method file contains a text representation of the method, including any pre-run error messages (see **Run:Prerun Method**).

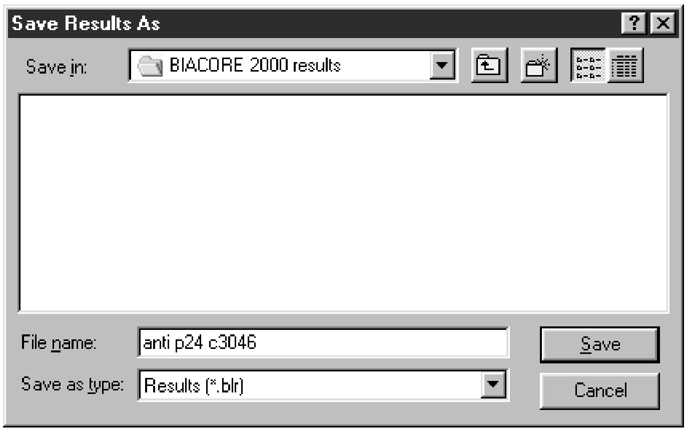
You can save a result file at any time during a run without interrupting the run. The results are saved at the point when you choose the command. If **Autosave** is active under **Tools:Preferences**, results will be saved automatically at preset intervals.

Result files are saved in BIACORE 2000 format. Use **File:Export** in the sensorgram window to export sensorgram data to a text file readable by other programs.

For a file which has already been saved, the **Save** command saves the data with the original file name, overwriting the original copy. Use **Save As** if you want to save the data with a different file name (e.g. if you have made changes to the notebook contents and want to keep the original result file).

Save As

Saves the contents of the currently active window in a new file. For a file which has not previously been saved, **Save As** is equivalent to **Save**.



Save Results As

Save in: BIACORE 2000 results

File name: anti p24 c3046

Save as type: Results (*.blr)

Export

You can export either sensorgram data or report point table to tab-separated text files for transfer to other applications. Choose which type of data to export from the submenu.

Sensorgram data are exported as tab-separated table of time and response values. In multichannel mode, one or more sensorgrams are exported according to the current display mode (see **View:Overlay** and **View:Plot Curve Classes**). The whole time range of the sensorgram data is exported, regardless of the range currently displayed in the sensorgram window.

Report point data are exported as a tab-separated table. Report points for cycles and all sensorgrams are exported regardless of the current display mode.

Exported data may be imported directly into most spreadsheet programs (e.g. Microsoft Excel®) or curve-fitting programs (e.g. Origin) by choosing tab-separated text format for the import operation. Note that BIAevaluation software can open .BLR files directly and does not require data to be exported.

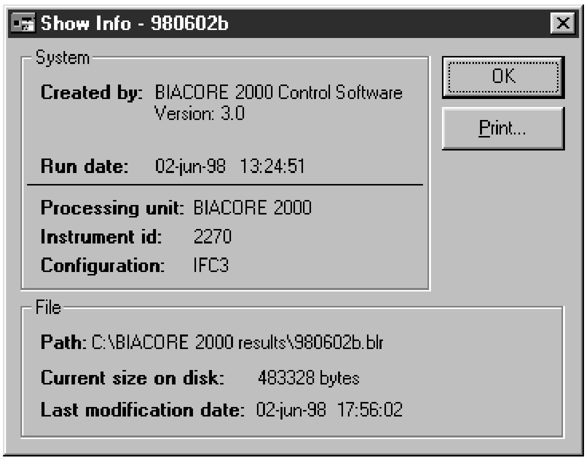
The event log cannot be exported. Notebook contents cannot be exported, but can be copied to other applications using the Windows clipboard.

File:Export is only available when a sensorgram window is active. You can export sensorgram data at any time during a run.

Show Info

Displays information about the current sensorgram. This command is only available when a sensorgram window is active.

The information displayed includes the date and time when the sensorgram was run, the BIACORE Control Software version, instrument Id and configuration, and file details for the result file where applicable (file details are replaced by the text **#N/A** for sensorgrams that have not been saved).



Show Info - 980602b

System

Created by: BIACORE 2000 Control Software Version: 3.0

Run date: 02-jun-98 13:24:51

Processing unit: BIACORE 2000
Instrument id: 2270
Configuration: IFC3

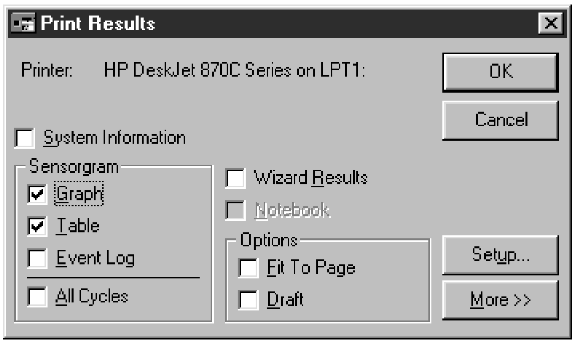
File

Path: C:\BIACORE 2000 results\980602b.blr
Current size on disk: 483328 bytes
Last modification date: 02-jun-98 17:56:02

Click on **Print** in the dialogue box to print the sensorgram information.

Print (ctrl-P)

Prints the contents of the active window.



Print Results

Printer: HP DeskJet 870C Series on LPT1:

☐ System Information

Sensorgram
☑ Graph
☑ Table
☐ Event Log
☐ All Cycles

☐ Wizard Results
☐ Notebook

Options
☐ Fit To Page
☐ Draft

The following options are available for the sensorgram window:

System Information

Check this box to print the sensorgram information (see **File:Show Info**).

Graph

Check this box to print the contents of the sensorgram window. Sensorgrams are printed as currently displayed with respect to overlay mode, markers, reference line/baseline and zoom settings. If **All Cycles** is checked, data for all cycles in the run will be printed. Otherwise, data for the current cycle only is printed.

Table

Check this box to print the report point table. The keywords in the printed report point table are those displayed on the screen (see **View:Keywords**).

Event Log

Check this box to print the event log. The normal or extended event log is printed according to the current event log mode.

All Cycles

Check this box to print data from all cycles in a multi-cycle run. If this box is not checked, only the currently displayed cycle will be printed.

Wizard results

Check this box to print the wizard results. Only available of the sensorgram was obtained by running an application wizard.

Fit To Page

Check this box to print each component on a separate page.

Draft

Checking this box reduces the number of points printed on the sensorgram. Use this feature if memory shortage in your printer causes difficulty in printing sensorgrams with large numbers of data points, or if printing takes an unduly long time.

Note: Printing in draft mode can occasionally hide single-point spikes in the sensorgram by skipping the outlier point. Do not use draft mode unless you have problems printing in normal mode.

Click on **Setup** to set the current printer properties (printer driver, page orientation, paper tray etc).

Click on the **More>>** button to set the margins and to select header and footer components for the printout.

Controlling sensorgram print size

Sensorgrams are printed at the current width of the page (between the margins). To print at a smaller width, increase the margin settings.

If **Fit to Page** is not checked, sensorgrams are printed with the same *proportions* as the current sensorgram window on the screen. Thus if the window is tall and narrow, the printout will also be tall and narrow. Drag a corner of the sensorgram window to adjust the window proportions.

If **Fit to Page** is checked, sensorgrams will be scaled to fit within the margins of the printed page, regardless of the proportions of the sensorgram window on the screen. To change the printed proportions in this mode, change the margin setting accordingly.

Exit

Closes the BIACORE 2000 application. You can also close BIACORE 2000 by pressing Alt-F4 or using the window control menu accessed from the button at the top left-hand corner of the window (these are standard Windows alternatives for closing an application). You are given an opportunity to save the data in any unsaved windows before closing the application.

You will be asked if you wish to shut down the instrument before closing the application. For more information on closing down the system, refer to Section 4.7.

You cannot close the application while a sensorgram is running.

If several instances of the control software are running, this option will only exit from the active instance and only prompt to shut down the system if the instance is connected to the system.

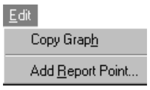
Note: BIAevaluation and BIAsimulation applications are not closed by exiting the control software, even if they were started from the control software **Tools** menu.

9.5 Edit menu

This menu is only displayed when a sensorgram or method editor window is active. The menu options relate to the type of active window.

9.5.1 Sensorgram window

The options that follow relate to the sensorgram window.



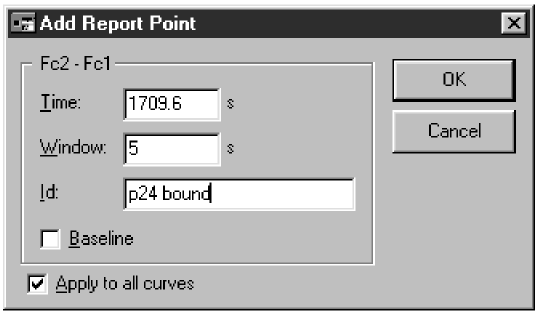
Copy Graph (ctrl-H)

Copies the current sensorgram display to the Windows clipboard. The copied display can then be pasted into another Windows application, e.g. Word or Write.

This option does not copy the report point table or the event log.

Add Report Point

Opens a dialogue box for adding a new report point to the current sensorgram window.



Fc2 - Fc1

Time: 1709.6 s

Window: 5 s

Id: p24 bound

□ Baseline

☑ Apply to all curves

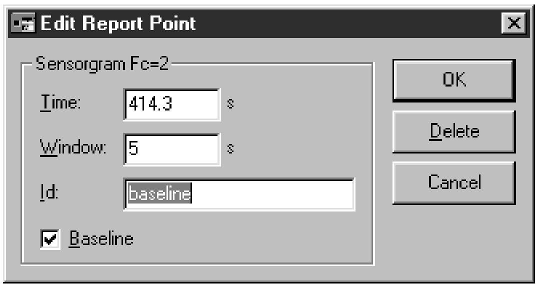
Enter values for **Time** and **Window** and a report point identifier for **Id.** If the reference line is active, the time at the reference line is suggested as the time for the report point. Response values recorded in the report point table will be the average response over the time window centered on the report point time.

Check **Baseline** if this report point is to be used as a baseline for subsequent points. Check **Apply to all curves in window** if you want to record a report point for all sensorgrams in the window in multi-channel mode. If this box is not checked, the report point will be recorded for the active sensorgram only (as indicated by the **Curve** list box in the toolbar).

The information displayed in the report point table is determined by the settings under **View:Keywords**. These settings can be changed at any time.

Note: You can also add report points to additional curves created with the CURVE command in a method, allowing report points to record e.g. temperature or buffer flow at specified times during the analysis.

To edit or delete a report point, double-click on the corresponding line in the report point table.



Edit Report Point

Sensorgram Fc=2

Time: 414.3 s

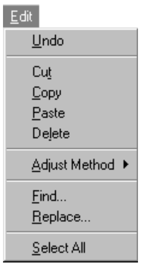
Window: 5 s

Id: baseline

☑ Baseline

Click on **Delete** to remove the report point, or change the settings and click on **OK** to edit the report point definition.

9.5.2 Method editor



Undo (ctrl-Z)

Reverses the last editing action.

Cut (ctrl-X)

Removes selected text and stores it in the Windows Clipboard, from where it can be pasted to a new location in the method or to another method definition. The removed text replaces any previous contents of the Clipboard.

To select text, drag with the left mouse button or hold down Shift and move the cursor with the arrow keys.

Copy (ctrl-C)

Copies selected text and stores it in the Windows Clipboard, from where it can be pasted to a new location in the method or to another method definition. The copied text replaces any previous contents of the Clipboard.

To select text, drag with the left mouse button or hold down Shift and move the cursor with the arrow keys.

Paste (ctrl-V)

Copies the contents of the Windows Clipboard into the current method file at the location of the cursor. A copy of the pasted text remains in the Clipboard until a new selection is cut or copied or the method file is closed.

Delete

Deletes marked text.

To select text, drag with the left mouse button or hold down Shift and move the cursor with the arrow keys.

Adjust Method

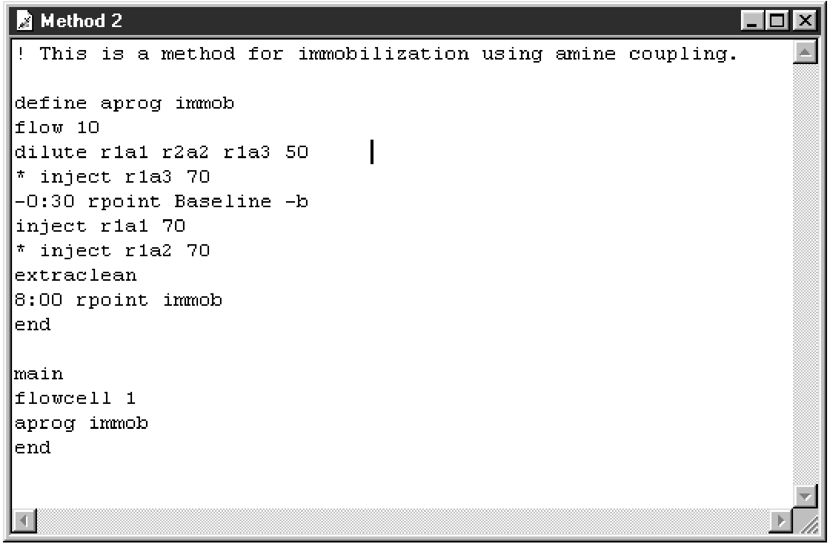
Adjusts the method definition to a chosen time base and a standardized text layout. Select the time base (seconds or minutes) from the **Adjust Method** submenu. Changing the time base alters the way times are displayed without changing the actual times (e.g. 2 minutes 30 seconds is displayed as 2:30 in minutes or 150 in seconds).

Note: Time specifications entered as numbers with no punctuation marks are read as seconds. Time specifications which contain a colon are read as minutes in the format MM:SS. The table below illustrates the results of adjusting to seconds or minutes base.

Entered as	*Adjusted to seconds*	*Adjusted to minutes*
30	30	0:30
75	75	1:15
0:30	30	0:30
1:15	75	1:15
1:5	65	1:05
1:10:00	4200	70:00
1:75	error (invalid entry format)	

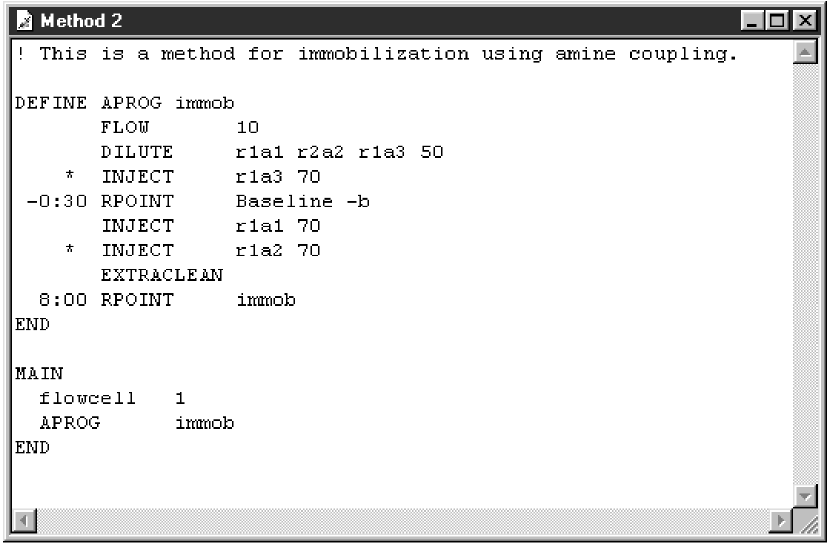
The standardized text layout indents method command lines to make method structure clear and converts all keywords to capitals. Parameters in LOOP constructions are aligned vertically. An example is shown below.

Before **Adjust Method**



```
! This is a method for immobilization using amine coupling.

define aprog immob
flow 10
dilute r1a1 r2a2 r1a3 50
* inject r1a3 70
-0:30 rpoint Baseline -b
inject r1a1 70
* inject r1a2 70
extraclean
8:00 rpoint immob
end

main
flowcell 1
aprog immob
end
```

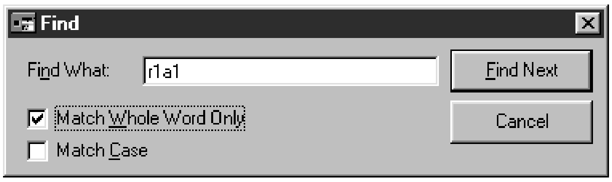
After **Adjust Method:Minutes**



```
! This is a method for immobilization using amine coupling.

DEFINE APROG immob
       FLOW       10
       DILUTE     r1a1 r2a2 r1a3 50
    *  INJECT     r1a3 70
 -0:30 RPOINT     Baseline -b
       INJECT     r1a1 70
    *  INJECT     r1a2 70
       EXTRACLEAN
  8:00 RPOINT     immob
END

MAIN
  flowcell   1
  APROG      immob
END
```

Find (Ctrl-F)

Search for a specified text. The search can be specified to match whole words only and/or match case.



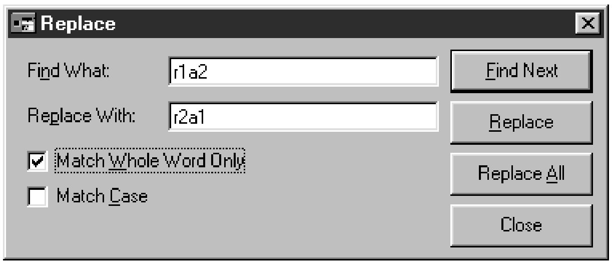
Find What: r1a1

☑ Match Whole Word Only

□ Match Case

Replace (Ctrl-R)

Replace a specified text string with entered text. The search can be specified to match whole words only and/or match case.



Find What: r1a2

Replace With: r2a1

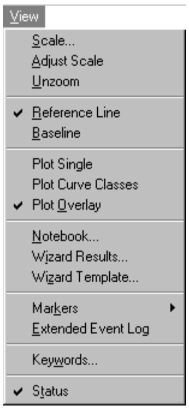
☑ Match Whole Word Only

□ Match Case

Select All

Selects all the text in the file (for **Copy**, **Cut** and **Delete** operations).

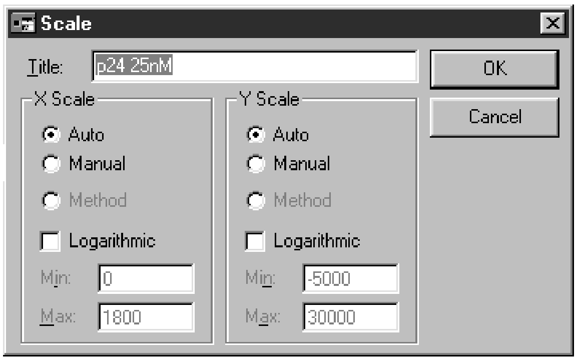
9.6 View menu



Commands in this menu relate primarily to sensorgram window display.

Scale

Sets the scale of the sensorgram display. You can also enter a title for the sensorgram. The title is displayed in the sensorgram window (and is independent of file name).



Title: p24 25nM

X Scale
● Auto
○ Manual
○ Method
□ Logarithmic
Min: 0
Max: 1800

Y Scale
● Auto
○ Manual
○ Method
□ Logarithmic
Min: -5000
Max: 30000

Scaling options may be set independently for the x-axis and the left and right y-axes (options for the right y-axis are displayed only for method controlled runs which use the `CURVE -o` command, see Appendix F).

Auto Scales the display automatically. For a sensorgram window displaying completed results (i.e. when the sensorgram is not running), auto-scaling adjusts the scale to accommodate the entire sensorgram. During a run, automatic scaling applies only to data collected after the setting is changed to **Auto**. To accommodate the entire sensorgram during a run, use **View:Adjust Scale**.

Method Scales the display according to the `SCALE` command in the method `APROG` by which the sensorgram was generated (not available if the `APROG` does not include a `SCALE` command).

Manual Scales the display according to maximum and minimum values entered in the dialogue box.

Logarithmic Sets a logarithmic instead of a linear scale.

Notes: With logarithmic scaling, the maximum and minimum values entered for **Manual** refer to the absolute response in RU, not to the logarithms of these values.

Zero or negative values cannot be plotted correctly on a logarithmic scale.

The default setting is **Auto** except for method-controlled runs where the `SCALE` command is used. Changing the scale of the sensorgram display does not affect the number of data points collected for the sensorgram(s).

The **Scale** dialogue box can also be displayed by double-clicking on any of the sensorgram axes.

Adjust Scale

Adjusts the scale of the sensorgram window in both x- and y-directions to display the entire sensorgram.

Choosing **Adjust Scale** is equivalent to setting **Scale** to **Auto** for both axes when a sensorgram is not running. Choosing **Adjust Scale** while a sensorgram is running fits the entire collected data in the display (compare automatic scaling, which adjusts the scale as necessary for incoming data).

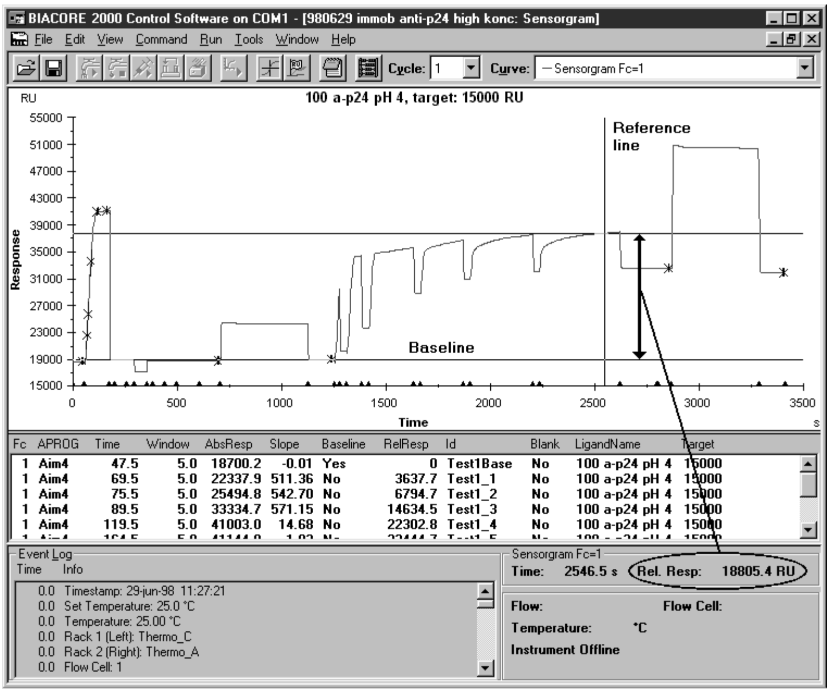
Unzoom (ctrl-U)

Restores a zoomed sensorgram display to the previous zoom factor. Double-clicking with the left mouse button is a short-cut to **Unzoom**.

The display is “zoomed” by dragging the cursor over the area to be enlarged. **Unzoom** will restore up to 10 previous zoomed sizes. When the original size has been restored, the **Unzoom** command is grey.

Reference Line (F8)

Toggles display of the reference line in the sensorgram window. The reference line allows exact coordinates (RU and elapsed time) to be read from the sensorgram. The coordinates are shown in the status window when the reference line is displayed. The response is shown as absolute RU if the baseline is not displayed and as RU relative to the baseline if the baseline is displayed (see **View:Baseline**).


BIACORE 2000 Control Software on COM1 - [980629 immob anti-p24 high konc: Sensorgram]
File Edit View Command Run Tools Window Help
Cycle: 1
Curve: — Sensorgram Fc=1
RU
100 a-p24 pH 4, target: 15000 RU
55000
51000
47000
43000
39000
35000
31000
27000
23000
19000
15000
Response
Reference line
Baseline
0
500
1000
1500
2000
2500
3000
3500
Time
s

Fc	APROG	Time	Window	AbsResp	Slope	Baseline	RelResp	Id	Blank	LigandName	Target
1	Aim4	47.5	5.0	18700.2	-0.01	Yes	0	Test1Base	No	100 a-p24 pH 4	15000
1	Aim4	69.5	5.0	22337.9	511.36	No	3637.7	Test1_1	No	100 a-p24 pH 4	15000
1	Aim4	75.5	5.0	25494.8	542.70	No	6794.7	Test1_2	No	100 a-p24 pH 4	15000
1	Aim4	89.5	5.0	33334.7	571.15	No	14634.5	Test1_3	No	100 a-p24 pH 4	15000
1	Aim4	119.5	5.0	41003.0	14.68	No	22302.8	Test1_4	No	100 a-p24 pH 4	15000

```
Event Log
Time     Info
  0.0  Timestamp: 29-jun-98  11:27:21
  0.0  Set Temperature: 25.0 °C
  0.0  Temperature: 25.00 °C
  0.0  Rack 1 (Left): Thermo_C
  0.0  Rack 2 (Right): Thermo_A
  0.0  Flow Cell: 1
```

```
Sensorgram Fc=1
Time:    2546.5 s    Rel. Resp:    18805.4 RU

Flow:                          Flow Cell:
Temperature:          °C
Instrument Offline
```

Move the reference line in one or more of the following ways:

- drag the reference line left or right with the cursor.
- press the Home or End keys to move to the beginning or end of the sensorgram window respectively.
- press the PageUp and PageDown keys to move rapidly left or right.
- press the left and right arrow keys for fine placement (pressing an arrow key once moves the reference line by one data point).
- click on a line in the report point table or event log to move to the corresponding marker (provided that the marker is within the current display). This feature can be used even if the report point or event markers are turned off.

The reference line can be used to interactively set the position of report points added to a sensorgram. Place the reference line in the position where the report point is required and choose **Edit:Add Report Point**. The position of the reference line will be suggested as the time for the report point.

Baseline (F9)

Toggles display of the baseline in the sensorgram window. The option is only available when the reference line is displayed. If the baseline is displayed, the status window shows the relative response at the reference line.

To obtain the relative response at a given point:

1. Display the reference line.
2. Move the reference line to a point to be used as baseline.
3. Display the baseline.
4. Move the reference line to the point to be measured and read the relative response.

Note: The response values used as baseline and displayed for the reference line represent single data points. Responses measured in this way are thus subject to variations arising from signal noise and are not recommended for determination of binding levels. To obtain more reliable values based on the average signal over a short time period, use report points with a suitable value for the report point window (see **Edit:Add Report Point**).

Plot Single

Displays only the currently active curve in the sensorgram window. Use the **Curve** list to change the current curve.

Plot Curve Classes

Plots either original sensorgrams (from individual flow cells) or reference-subtracted curves according to the type of curve that is currently active. Selecting a different type of curve from the **Curve** list will change the curve class that is plotted.

Note: Additional curves created by the CURVE command in a method form additional classes for **Plot Curve Classes**.

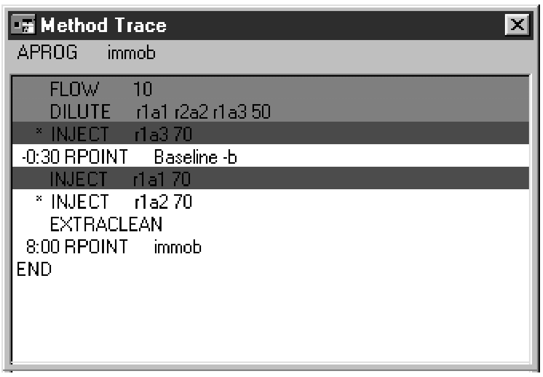
Plot Overlay

Toggles between display of one and several curves in the sensorgram window. The report point table lists report points for the curves that are displayed.

In single sensorgram display mode, choose the sensorgram to display in the **Curve** list on the toolbar. If the keyboard focus is on the curve list, you can switch between curves using the arrow keys.

Method Trace

Displays a method trace window where the progress of the method can be followed. See Section 4.6.4 for details.



```
APROG    immob
     FLOW       10
     DILUTE     r1a1 r2a2 r1a3 50
   * INJECT     r1a3 70
-0:30 RPOINT    Baseline -b
     INJECT     r1a1 70
   * INJECT     r1a2 70
     EXTRACLEAN
 8:00 RPOINT    immob
END
```

Method

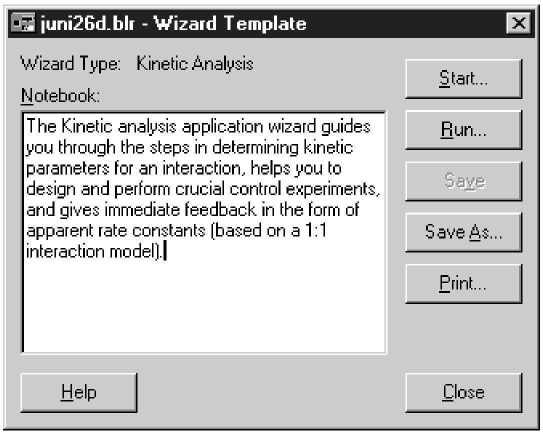
Displays the method associated with the current sensorgram window. This command is only available if the analysis was run from a method.

Wizard results

Displays the application wizard results. This command is only available if the analysis was run from an application wizard.

Wizard template

Provides options for running and/or printing the wizard settings used to create the result file. This command is only available if the analysis was run from an application wizard. See Section 5.1 for more details.



Wizard Type: Kinetic Analysis

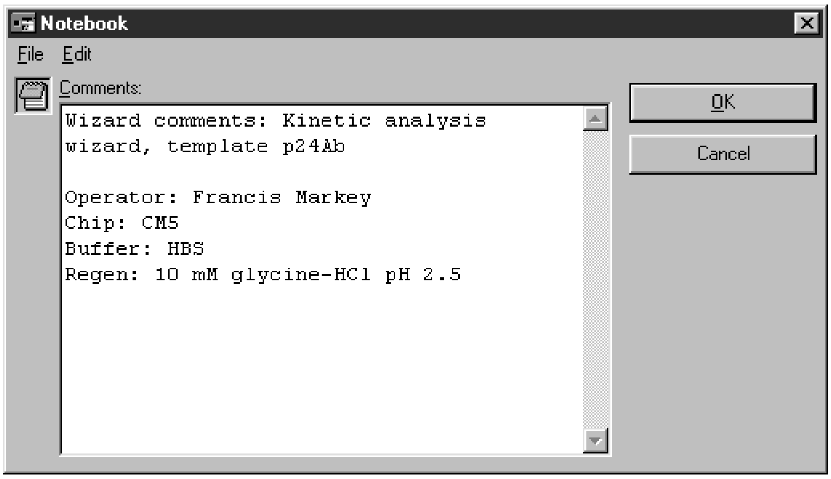
Notebook:

The Kinetic analysis application wizard guides you through the steps in determining kinetic parameters for an interaction, helps you to design and perform crucial control experiments, and gives immediate feedback in the form of apparent rate constants (based on a 1:1 interaction model).

Note: It is not necessary to save the wizard settings explicitly as a template in order to use this command. The settings are saved in the result file and may be accessed with **View:Wizard Template** even if no separate template was saved.

Notebook

Opens the notebook for the current sensorgram.



Comments:

```
Wizard comments: Kinetic analysis
wizard, template p24Ab

Operator: Francis Markey
Chip: CM5
Buffer: HBS
Regen: 10 mM glycine-HCl pH 2.5
```

If the notebook has not previously been opened for this sensorgram, you may choose to open the notebook template (see below) or open a blank notebook. In the notebook, you can move quickly to user-defined “fields” using the tab (forwards) or Shift-tab (backwards) keys. A field is defined by the first occurrence of a colon (:) on a line.

The notebook window has a menu bar with **File** and **Edit** menus. The **File** menu contains options for saving the current notebook as a template and printing the notebook. The **Edit** menu contains standard

Windows text editing commands (**Undo**, **Cut**, **Copy**, **Paste**, **Delete**, **Select All**).

The same notebook template is available for all runs.

If you want to change the notebook template:

1. Start a manual sensorgram.
2. Open the notebook and make the required changes to the template.
3. Choose **File:Save As Template** from the menu in the notebook dialogue box.
4. Stop the method and discard the sensorgram.

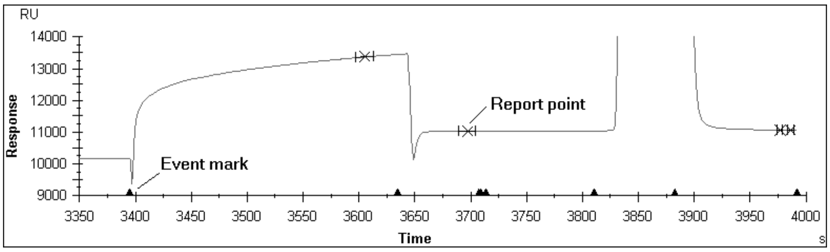
Changes you make to the template do not affect the notebook for runs that have previously been saved.

Markers

Toggles the display of report points and event markers in the sensorgram window.

A *report point* is displayed as an x on the sensorgram with an indication of the report point window. For further information on report points, refer to Section 9.5.1.

An *event marker* is displayed as a black triangle on the x-axis. Event markers are added to the display at functional breakpoints (such as injections or changes in the flow rate). A description of the event is recorded in the event log.

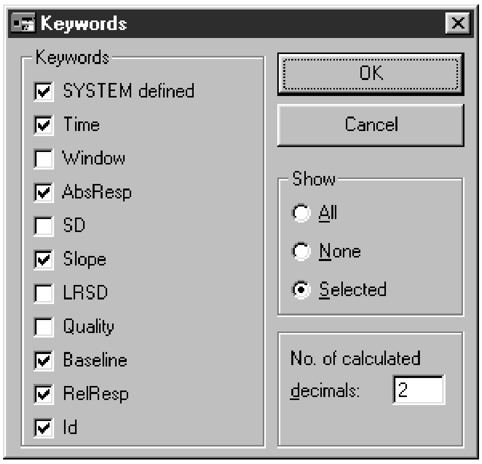


Extended Event Log

Toggles the event log display between standard and extended. The extended event log includes additional events which may be useful in troubleshooting and service contexts.

Keywords

Selects the keywords to include as headers in the report point table and the number of decimals displayed for report point values.



Keywords

☑ SYSTEM defined
☑ Time
□ Window
☑ AbsResp
□ SD
☑ Slope
□ LRSD
□ Quality
☑ Baseline
☑ RelResp
☑ Id

Show

○ All
○ None
● Selected

No. of calculated decimals: 2

Choose to show all, none or selected keywords. If you choose **None**, the report point table is hidden.

The keywords are as follows:

SYSTEM defined **Fc** (flow cell) for all report points and **APROG** (name of the `APROG`) for report points generated by the `RPOINT` command in a method.

Time Time for the report point, measured from the start of the sensorgram.

Window The time window for calculation of response values. The window extends equally on either side of the report point time. Setting the window to 0 selects a single data point.

AbsResp Absolute response in RU, calculated as the mean value over the time window.

SD Standard deviation of data points in the time window, calculated as

$$SD = \sqrt{\frac{1}{(n-1)}\sum\left(y_i - \bar{y}\right)^2}$$

where n = number of points
and y = response in RU

Slope Slope during time window in RU·s^{-1}, calculated as

$$\text{slope} = \frac{\sum\left((y_i - \bar{y})^2 (x_i - \bar{x})\right)}{\sum (x_i - \bar{x})^2}$$

LRSD Alignment of slope to a straight line (regression coefficient), calculated as

$$\text{LRSD} = \sqrt{\frac{Q_0}{(n-2)}}$$

where

$$Q_0 = \sum (y_i - \bar{y})^2 - \frac{\left(\sum (y_i - \bar{y})(x_i - \bar{x})\right)^2}{\sum (x_i - \bar{x})^2}$$

Quality **OK** for normal signal quality, **Bad** for bad data.

Baseline YES for report points defined as baseline. Otherwise NO.

RelResp Relative response (difference between AbsResp and baseline) in RU. #N/A if no baseline has been set.

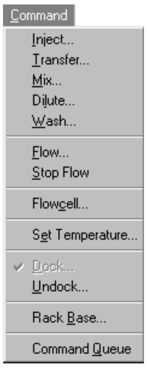
Id Report point id (text string).

APROG keywords For method-controlled runs, additional keywords may be defined in the method.

Status

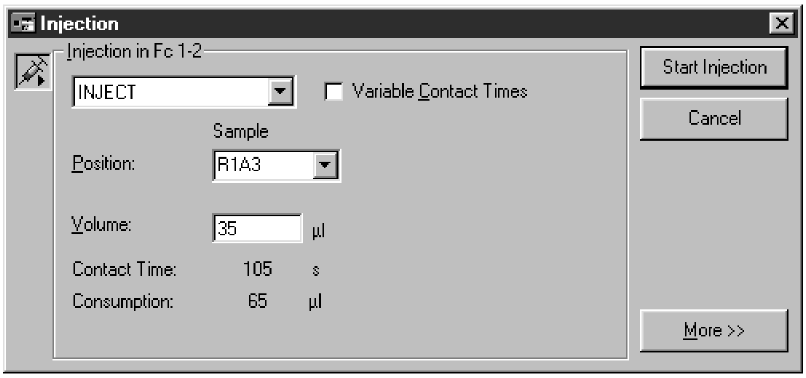
Toggles display of the status window between the full window and a single status line showing time, response and status messages.

9.7 Command menu



Inject

Injects sample over the sensor chip surface.



Injection in Fc 1-2

INJECT

☐ Variable Contact Times

	Sample	
Position:	R1A3	
Volume:	35	µl
Contact Time:	105	s
Consumption:	65	µl

Use the fields in this dialogue box as follows:

Inject command

Choose the appropriate inject command from the list box. A range of inject commands is provided for different application purposes, as summarized in the table below. All inject commands except MANUAL INJECT prepare and perform the injection automatically and wash the IFC at the end of the injection. See Section 4.4.6 for a table of preparation and finishing times for the different inject times.

Command	***Comments***	***Injected volume***	***Sample consumption***	***Max flow rate***
INJECT	General purpose inject.	5-325 µl	Injected vol +30 µl	100 µl/min
KINJECT	Low dispersion inject for kinetics and for samples sensitive to dilution with running buffer. Controlled dissociation time after sample injection.	10-250 µl	Injected vol +40 µl	100 µl/min
QUICKINJECT	Low sample consumption. Less preparation time than INJECT but more sample dispersion.	5-325 µl	Injected vol +10 µl	100 µl/min
COINJECT	Injection of second sample immediately following the first (e.g. for dissociation in the presence of ligand). Variable contact times for first sample only. Recovery of either first or second sample, but not both.	Sample 1: 10-100 µl Sample 2: 15-250 µl	Sample 1: injected vol +40 µl Sample 2: injected vol +40 µl	100 µl/min
BIGINJECT	Large volume inject. Direct inject up to 325 µl, sample loop between 326 and 425 µl, direct between 426 and 750 µl.	326-750 µl	Injected vol +52 µl	50 µl/min*
MANUAL INJECT	Manually controlled inject. See Section 4.4.8.	1-100 µl / loading	Loaded vol +30 µl	100 µl/min

*BIGINJECT cannot be performed at higher flow rates because the autosampler tubing must be refilled during injection from the sample loop for samples larger than 425 µl.

Variable Contact Times

Check this box for variable contact time injections in multichannel mode. The box is only available if all four flow cells are included in the detection. Volume fields appear for the flow cells in the current flow path (e.g. if you have set flow path 1-2-3-4 there will be four volume fields for an INJECT command). Each volume must be less than or equal to the volume in the preceding flow cell. For example, with four flow cells:

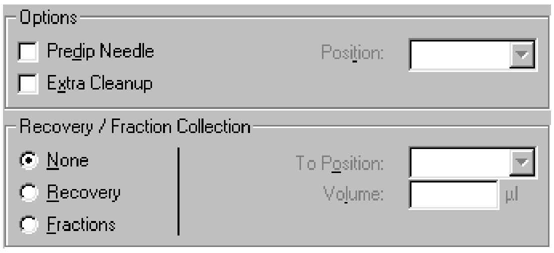
Permitted:	40	30	20	10
	40	40	20	20
	40	25	25	10
Not permitted:	40	20	***30***	10
	40	30	20	***30***

Position
Enter the rack position for the sample to be injected, or select the position from the pull-down list.

Volume
Enter the volume of sample to be injected. The dialogue box also displays contact time of the sample with the sensor surface (contact time = injected volume/flow rate) and the sample consumption for the injection (see table above).

The fields in the dialogue box are adjusted to correspond to the characteristics of the currently selected inject command (e.g. COINJECT requires two sample positions, with variable contact times permitted only for the first injection).

The **More>>** button opens additional options for the injection:



Options
□ Predip Needle   Position:
□ Extra Cleanup

Recovery / Fraction Collection
● None   To Position:
○ Recovery   Volume: µl
○ Fractions

Check **Predip Needle** and enter a rack position to dip the needle in the specified position before aspirating sample. This operation also aspirates 10 µl from the predip position into the autosampler tubing to prevent the sample from coming into contact with running buffer. Use this feature to minimize carry-over between sample vials, e.g. when solutions of widely different composition are used in succession, and also when it is important that the sample in the vial does not become contaminated with running buffer. Use of **Predip Needle** in an extra vial of sample can also help to minimize sample dispersion in the injection by improving the separation between injected sample and running buffer.

Check **Extra Cleanup** to perform an additional wash of the needle and IFC after the injection. Use this feature to minimize carry-over when the next injection uses sample of widely different composition. (The IFC is routinely washed as part of the standard injection sequence. **Extra Cleanup** should be used if particular care is necessary, e.g. after injection of regeneration solution.)

Check **Recovery** or **Fractions** and enter values in the appropriate fields to activate sample recovery or fraction collection for the injection.

> *IMPORTANT*
> Before starting an injection with recovery or fraction collection, perform a **Wash** with **Recovery** option to wash the recovery cup and IFC recovery channel and empty the recovery channel. If you do not do this the recovered sample may be diluted with up to 15-20 µl buffer from the recovery channel.

Recovery **To Position** is the rack position to which the recovered sample will be dispensed. **Volume** is the volume of liquid that will be recovered from the recovery cup (range 5-350 µl, recommended >25 µl for best precision).

Fractions **Start Position** is the rack position for the first fraction. **Volume** is the volume of each fraction (range 5-325 µl, recommended >25 µl for best precision). **No of fractions** is the number of fractions that will be collected (range 1-96). If the end of a rack is reached before the specified number of fractions has been collected, fraction collection is terminated and the effluent goes to waste.

Note: Fractions are marked with event markers on the sensorgram and in the event log. Each marker is placed at the correct time on the sensorgram, but does not appear until the fraction is actually collected from the recovery cup. The delay represents the IFC channel volume between the flow cell and the recovery cup (about 15 µl). For example, at a flow rate of 30 µl/min, the event marker for a fraction collected at 10.0 minutes will not appear until 10.5 minutes but will be placed at the correct time of 10.0 minutes on the sensorgram.

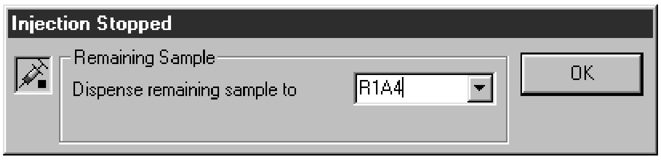
Manually controlled injections

The inject command MANUAL INJECT starts an interactive injection mode, where the injection can be paused and resumed at any time. See Section 4.4.8 for a full description of manual injection.

The **Inject** command is only available while a manually controlled sensorgram is running.

Stop Inject

Stops the current injection immediately. You can allow the excess sample which has not yet been injected to go to waste or recover it in a selected rack position:



Injection Stopped

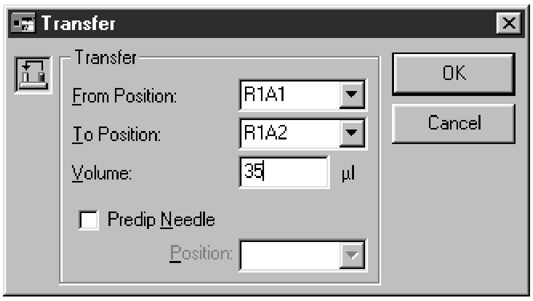
Remaining Sample

Dispense remaining sample to: R1A4

The **Stop Inject** command is not available during MANUAL INJECT. To end a manual injection, click on the **Exit** button in the MANUAL INJECT dialogue box (see Section 4.4.8).

Transfer

Transfers sample from one rack position to another.



Transfer

From Position: R1A1

To Position: R1A2

Volume: 35 µl

□ Predip Needle

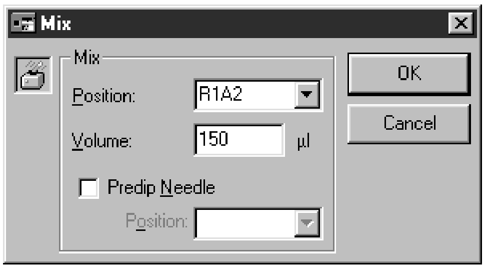
Position:

Enter rack positions and volume to be transferred in the appropriate fields.

Check **Predip Needle** and enter a rack position to dip the needle in the specified position before the transfer operation. This function also aspirates 10 µl from the predip position into the autosampler tubing to prevent the sample from coming into contact with running buffer. Use this feature to minimize carry-over between sample vials, e.g. when solutions of widely different composition are used in succession, and also when it is important that the sample in the vial does not become contaminated with running buffer.

Mix

Mixes sample in one position. Use this command for example after transferring two different solutions to a common vial to prepare a homogeneous mixture.



Mix

Position: R1A2

Volume: 150 µl

□ Predip Needle

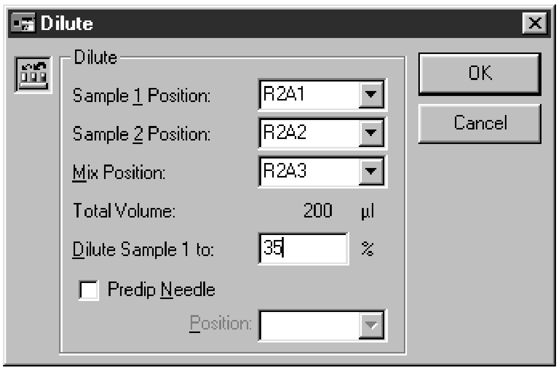
Position:

Enter the required rack position and appropriate volume to be mixed (maximum 450 µl), then click on **OK**. As a general guideline, mix using 75% of the total volume in the vial. The mixing procedure aspirates the specified volume of sample and then dispenses it to the same vial. This is performed three times.

Check **Predip Needle** and enter a rack position to dip the needle in the specified position before the mix operation. This function also aspirates 10 µl from the predip position into the autosampler tubing to prevent the sample from coming into contact with running buffer. Use this feature to minimize carry-over between sample vials, e.g. when solutions of widely different composition are used in succession, and also when it is important that the sample in the vial does not become contaminated with running buffer.

Dilute

This command performs the equivalent of **Transfer** followed by **Mix** with a fixed final volume of 200 µl, and may be used for diluting samples with buffer or for preparing a defined mixture of two samples.



Dilute

Sample 1 Position: R2A1

Sample 2 Position: R2A2

Mix Position: R2A3

Total Volume: 200 µl

Dilute Sample 1 to: 35 %

□ Predip Needle

Position:

Sample 1 Position Rack position for the sample to be diluted.

Sample 2 Position Rack position for the solution to dilute with.

Mix Position Rack position in which the diluted sample will be prepared.

Total volume The final total volume of diluted sample is fixed at 200 µl.

Dilute Sample 1 to Final concentration of sample 1 expressed as a percentage.

The example in the illustration above will prepare a mixture consisting of 70 µl sample from position r2a1 + 130 µl solution from position r2a2. The mixture will be prepared in position r2a3.

Check **Predip Needle** and enter a rack position to dip the needle in the specified position before the dilute operation. This function also aspirates 10 µl from the predip position into the autosampler tubing to prevent the sample from coming into contact with running buffer. Use this feature to minimize carry-over between sample vials, e.g. when solutions of widely different composition are used in succession, and also when it is important that the sample in the vial does not become contaminated with running buffer.

Wash

Extra wash of autosampler needle, IFC or recovery cup.

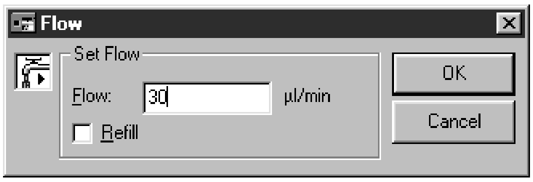
For needle wash, running buffer is pumped through the needle to waste.

For recovery wash, the recovery channel in the IFC and the recovery cup are rinsed with liquid from the specified position. Choose the position **BUFFER** to rinse with running buffer.

Note: The liquid handling system is washed automatically as required for normal operation. Use the **Wash** command when additional security against cross-contamination is required.

Flow

Sets the flow rate for running buffer.



Flow: 30 µl/min

□ Refill

Enter the required flow rate in µl/min (permitted range 0-100 µl/min in steps of 1 µl). To stop the flow, either choose **Command:Stop Flow** or set the flow to 0 µl/min with **Command:Flow**.

If the **Refill** box is checked, the pump will be refilled before the new flow rate is set. The refill operation takes about 7 seconds.

This command is not available unless a sensorgram is running.

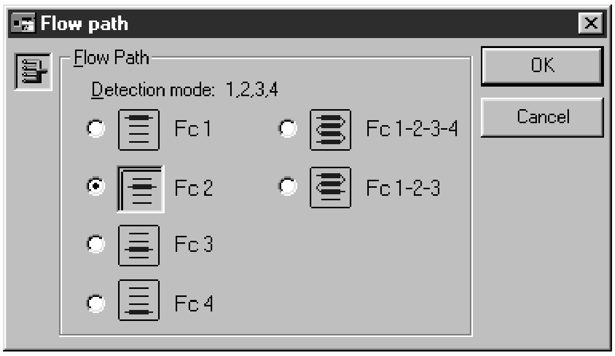
Stop Flow

Stops the flow of running buffer. This command is equivalent to setting the flow to 0 µl/min.

This command is only available when the flow is set to a non-zero value. The flow is stopped automatically at the end of a manual sensorgram or method-controlled run. The flow cannot be stopped during an injection (except MANUAL INJECT).

Flow path

Changes the flow path during a sensorgram.



Detection mode: 1,2,3,4

○ Fc 1 ○ Fc 1-2-3-4

● Fc 2 ○ Fc 1-2-3

○ Fc 3

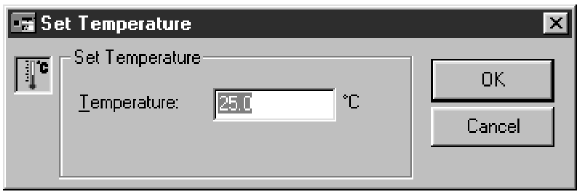
○ Fc 4

Choose the required flow path as indicated by the icons. You can only change the flow path to a subset of the originally selected flow cells (e.g. if you start with Fc1-2, you can change to Fc1 or Fc2 but to any of the other options). Changing the flow path does not change the detection mode.

This command is only available when a sensorgram is running.

Set Temperature

Sets the required temperature at the flow cell.



Set Temperature

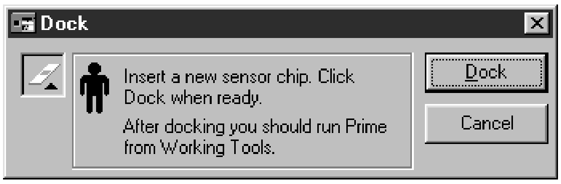
Temperature: 25.0 °C

The temperature display in the status window toggles between the set temperature and the actual temperature until the measured temperature is stable. A temperature change of 5°C takes about 60 minutes.

Wait until the sensor chip surface temperature is stable before performing an analysis (as shown by the temperature LED on the processing unit and the temperature display in the BIACORE Control Software status window).

Dock

Docks the sensor chip.



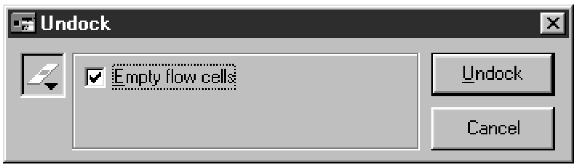
Insert a new sensor chip. Click Dock when ready.

After docking you should run Prime from Working Tools.

The docking operation takes a few seconds to complete. After docking, the **Working Tools** dialogue box appears automatically with **Prime** highlighted as a reminder that you should run **Prime** after docking a new chip. For best results, you should also run **Working Tools:Normalize** after docking a new chip.

Undock

Undocks the sensor chip.



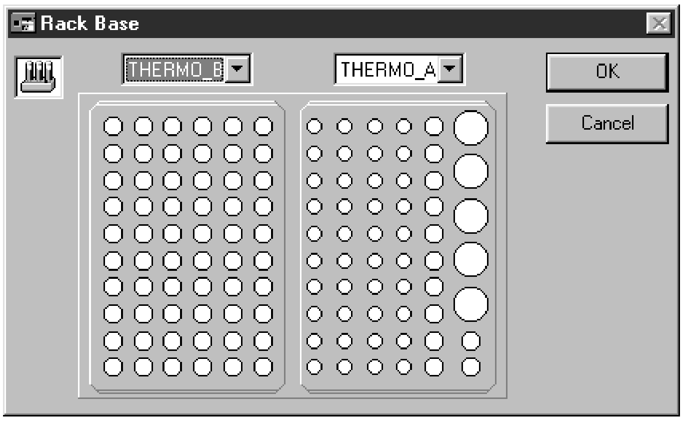
If the **Empty flow cells** box is checked, the flow cells will be emptied of liquid before the sensor chip is undocked. This option is checked by default, and should always be used unless your sensor chip surface is particularly sensitive to exposure to air (e.g. your ligand is readily oxidized in air).

When the undocking operation is complete, the **Dock** dialogue box is displayed automatically in preparation for docking a new sensor chip. If you do not want to dock a new chip (e.g. you intend to shut down the instrument), simply choose **Cancel** from this dialogue box.

Important
Do not attempt to remove the sensor chip from the instrument until the undocking operation is complete.

Rack Base

Defines the type and placing of racks on the autosampler rack base.



The dialogue box shows the current settings for rack base numbers 1 and 2 (left and right on the autosampler respectively). Choose the settings that correspond to the racks you intend to use.

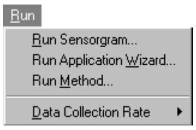
The settings in this dialogue box determine the positions available for manual commands that use the autosampler (e.g. **Inject**, **Dilute**, **Transfer**, **Mix**) and also the valid positions for the corresponding commands in a method that does not include a RACK command. Positions in a method are checked during pre-run. See Section 4.2.4 for details of valid positions in a method.

Note: There is no automatic sensing of rack type in the software. If you set the wrong rack type, the autosampler needle may be directed to the wrong or non-existent positions. Remember to re-define the rack base using this command if you change the type of rack in the autosampler during manual operation.

Command Queue

Starts or stops the command queue according to the current status of the queue. The command is checked if the queue is started. This command can only be used while a sensorgram is running.

9.8 Run menu



Run Sensorgram

Starts a manually controlled sensorgram. The **Detection** and **Flow** dialogue boxes are automatically displayed before the sensorgram starts, to ensure that the settings are correct for the intended run. By default, these dialogue boxes display the most recent settings.

All method editor windows will be closed before the sensorgram starts. If any method editor window contains unsaved changes, you will be prompted to save the method. Changes which are not saved at this stage are lost.

This command changes to **Stop Sensorgram** while a sensorgram is running.

Note: During sensorgram start, the status window transiently displays various settings for flow rate and flow path, regardless of the settings requested for the run. This is a normal part of the sensorgram start-up procedure. The requested settings will be established before data collection starts.

Stop Sensorgram

Stops the current manually controlled sensorgram. Once you have stopped a sensorgram you cannot resume recording except as a new sensorgram.

The liquid handling system is flushed with buffer and the flow rate is set to 0 at the end of the sensorgram.

The command changes to **Run Sensorgram** when a sensorgram is stopped.

Notes: During sensorgram end, the status window transiently displays various settings for flow rate and flow path, regardless of the settings used during the run. This is a normal part of the sensorgram termination procedure.

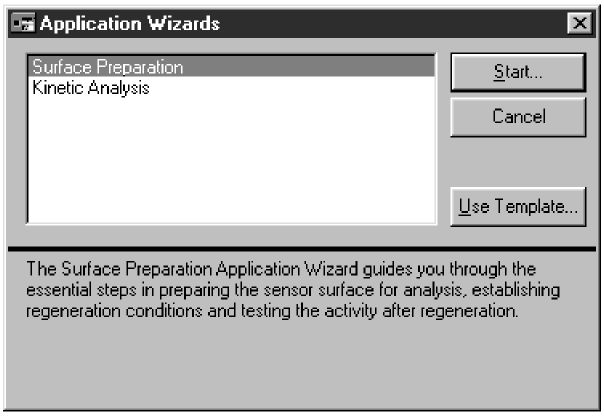
To stop a method, choose **Command:Stop Method**.

Start new cycle

Starts a new cycle in a manually controlled run. This command is only available while a manual sensorgram is running.

Run Application Wizard

Presents a list of application wizards that may be run. See Section 5.1 for more details.



The Surface Preparation Application Wizard guides you through the essential steps in preparing the sensor surface for analysis, establishing regeneration conditions and testing the activity after regeneration.

Run Method

Runs a method from the currently active method editor window or from a file on disk.

If a method editor window is currently active, the **Run Method** command will start the method in that editor window. Otherwise, the command will prompt for a method file.

Pre-run is performed automatically whenever a method run is requested, to prevent syntactically incorrect methods from starting.

Before the method starts, a number of dialogue boxes may appear depending on the current instrument status and the method contents:

- If **Autosave** is set for method runs (see **View:Preferences**), you will be asked to specify a result file name for the autosave feature.
- If the method does not contain a DETECTION command, you will be asked to select the detection mode and flow path. Make sure that you select a detection mode that is compatible with the method purpose. If in doubt, choose **FC1-2-3-4**, since this will ensure that data is recorded from all four channels.
- If the data collection rate is currently set to **Low** or **High**, you will be given an opportunity to reset the rate to **Medium**.
- You will be warned if there is limited space available on the computer hard disk. You should make more space available before starting the run, particularly if **Autosave** is set to on. If the computer runs out of hard disk space during an autosave operation, sensorgram data will probably be lost.
- If a sensorgram window is open and the contents have not been saved, you will be given an opportunity to save the results. (Note that this prompt refers to results currently displayed on the screen, not to the results which will be generated by the method being started.) Any open sensorgram window will be closed before the method starts.
- If any method editor windows are open and the contents have not been saved, you will be given an opportunity to save the method(s). All method editor windows will be closed before the method starts. However, if the method is being started from an active method editor window, this window will be reopened when the method is completed. Note that unsaved methods are not reopened on completion of a method that is run from a saved file.

Stop Method

Stops a running method. The method will stop at the end of the current cycle (i.e. the end of the current APROG in the method). Note that the method will run to completion if **Stop Method** is requested during the last cycle in the method. **Stop Method** automatically washes the system as for a normal method completion.

To stop a method immediately, use the emergency stop Ctrl-Break (see Section 4.4.14). You will be asked if you want to wash the liquid handling system.

Data Collection Rate

Determines the number of points collected per unit time during the sensorgram. The available settings are **Low**, **Medium** and **High**, corresponding to actual data collection rates as summarized below.

	Detection mode	
Data collection rate	Single	Multichannel
Low	0.1 point/s	0.1 point/s
Medium	1.0 point/s	1.0 point/s
High	10.0 points/s	2.0 points/s

Note: Values in multichannel mode refer to the collection rate per channel.

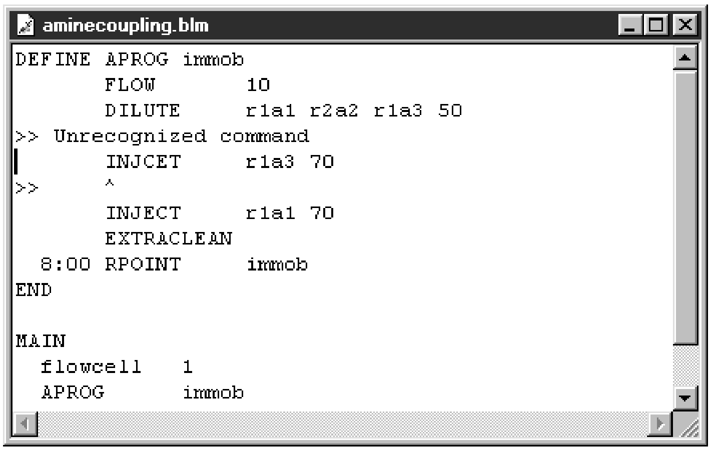
The default setting of **Medium** is adequate for most purposes. Set the rate to **High** for applications requiring very high resolution. Set the rate to **Low** to reduce the size of the sensorgram file for long runs. A reminder is issued if you start a run with **Data Collection Rate** set to **High** or **Low**.

Sensorgram data is recorded internally in segments of maximum 36000 points. Thus at a collection rate of 10 measurements/s, one segment will take 3600 s = 60 minutes. When a segment is full, automatic data reduction occurs whereby pairs of adjacent points are averaged to one point and data collection continues at half the previous rate. A new segment is started whenever the **Data Collection Rate** is set (even if the current setting is not changed). To avoid data reduction, therefore, choose **Data Collection Rate** shortly before the end of a segment.

Data reduction is recorded as an event in the event log.

Prerun Method (ctrl-P)

Checks the syntax of the method in the active method editor window. If syntax errors are detected you will be warned, and an explanatory comment will be included in your method to help you find the error. The illustration below shows a syntax error resulting from an incorrectly spelled keyword. See Section E.5 for further details.



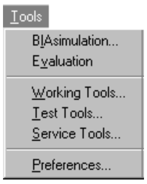
```
DEFINE APROG immob
       FLOW       10
       DILUTE     r1a1 r2a2 r1a3 50
>> Unrecognized command
       INJCET     r1a3 70
>>     ^
       INJECT     r1a1 70
       EXTRACLEAN
  8:00 RPOINT     immob
END

MAIN
  flowcell   1
  APROG      immob
```

Syntax error messages are removed automatically from the method when you correct the error and repeat the pre-run.

Notes: If the method does not contain a RACK command, the autosampler position specifications in the method will be checked against the current rack settings. Position specifications which are not compatible with the current settings are reported as syntax errors.

Pre-run is performed automatically whenever a method run is requested, to prevent syntactically incorrect methods from starting. You can however save a method with both syntax errors and syntax error messages.

9.9 Tools menu



BIAsimulation

Provides access to the BIAsimulation software, which currently provides facilities for simulating:

- kinetic analysis of association and dissociation phases, with optional signal noise and baseline drift for realistic simulation
- kinetic analyses with heterogeneous ligand or analyte
- multiple binding studies
- epitope mapping matrices
- concentration determination.

BIAsimulation software is described in full in the BIAsimulation Software Handbook.

This command is not available if BIAsimulation software has not been installed.

Evaluation (ctrl-E)

Provides access to BIAevaluation software, which currently provides facilities for:

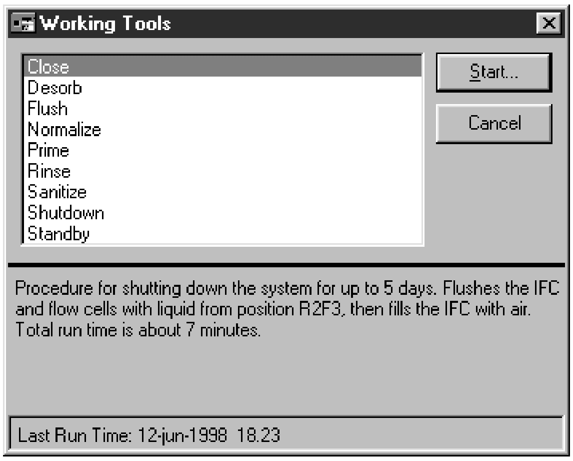
- overlay plots for comparison of sensorgrams
- evaluation of binding rates
- derivation of kinetic constants
- evaluation of report point tables for concentration measurement.

BIAevaluation software is described in the BIAevaluation Software Handbook.

This command is not available if BIAevaluation software has not been installed.

Working Tools

The working tools are a set of routines for starting up and shutting down the system and for special cleaning procedures.



Procedure for shutting down the system for up to 5 days. Flushes the IFC and flow cells with liquid from position R2F3, then fills the IFC with air. Total run time is about 7 minutes.

Last Run Time: 12-jun-1998 18.23

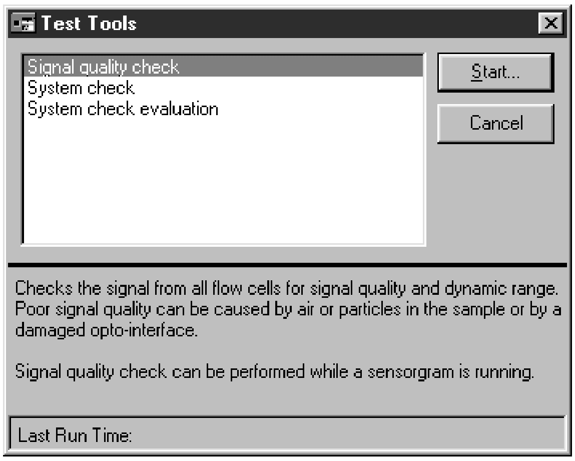
To use a working tool, choose the tool from the list and click on **Start**. Instructions for each step in the procedure are provided on the screen.

The last time the highlighted tool was used is displayed at the bottom of the dialogue box.

See Section 7.2.1 for a description of the working tools.

Test Tools

The test tools are a set of routines for testing the system performance.



Checks the signal from all flow cells for signal quality and dynamic range. Poor signal quality can be caused by air or particles in the sample or by a damaged opto-interface.

Signal quality check can be performed while a sensorgram is running.

Last Run Time:

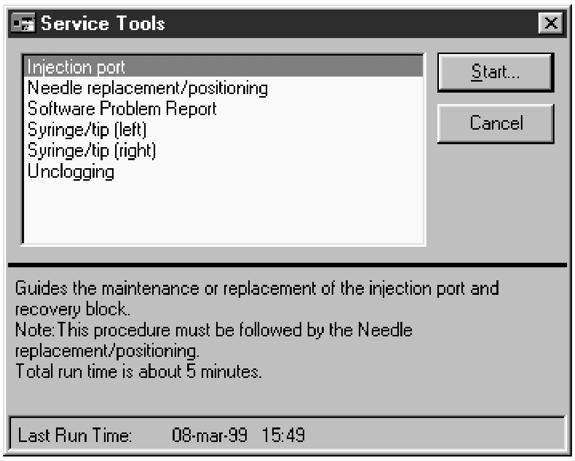
To use a test tool, choose the tool from the list and click on **Start**. Instructions for each step in the procedure are provided on the screen.

The last time the highlighted tool was used is displayed at the bottom of the dialogue box.

See Section 7.2.2 for a description of the test tools.

Service Tools

The service tools are a set of routines for system maintenance.



Injection port
Needle replacement/positioning
Software Problem Report
Syringe/tip (left)
Syringe/tip (right)
Unclogging

Guides the maintenance or replacement of the injection port and recovery block.
Note:This procedure must be followed by the Needle replacement/positioning.
Total run time is about 5 minutes.

Last Run Time: 08-mar-99 15:49

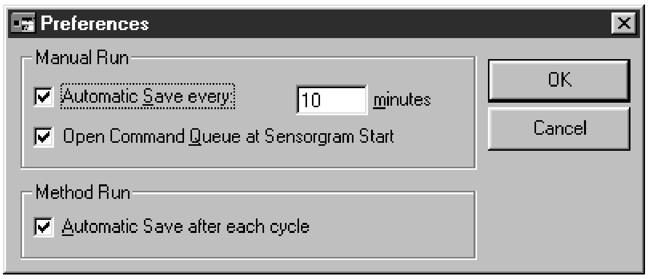
To use a service tool, choose the tool from the list and click on **Start**. Instructions for each step in the procedure are provided on the screen.

The last time the highlighted tool was used is displayed at the bottom of the dialogue box.

See Section 7.2.3 for a description of the service tools.

Preferences

Sets preferences for control software operation. **Preferences** settings apply for all runs until the settings are changed.



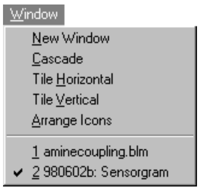
Manual Run
☑ Automatic Save every: 10 minutes
☑ Open Command Queue at Sensorgram Start

Method Run
☑ Automatic Save after each cycle

The **Autosave** function will save results at the specified interval during a manual run or after every cycle in a method-controlled run, providing a safeguard against loss of data in the event of a power failure or computer malfunction. Check the corresponding box and enter a time interval to turn the **Autosave** function on for manual and/or method-controlled runs. Setting **Autosave** to on is recommended, particularly for unattended method-controlled runs.

Note: If **Autosave** is turned on, you will be asked for a filename in which to save data before the run is started.

9.10 Window menu



New window

Opens a new sensorgram window, providing a new view of the current data. You can set display properties (scale, overlay mode, reference line and baseline etc.) independently for separate sensorgram windows. You can also display different cycles from the same result file in separate windows as long as the sensorgram is not running (you can only display the current cycle during a run). You cannot however display different result files in different windows at the same time.

This feature is particularly useful in multichannel detection mode with reference cell subtraction, since the original data and the subtracted curves frequently require different scale settings.

Cascade, Tile Horizontal, Tile vertical

These commands are standard Windows commands that control the way separate windows are arranged. Refer to the Windows 95 on-line help for a description of these functions.

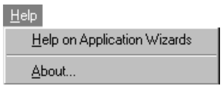
Arrange Icons

Arranges icons for minimized windows in the BIACORE 2000 application window. This is a standard Windows command, and has no effect if windows are not minimized.

<Open window list>

Open or minimized windows are listed at the bottom of the **Window** menu. The active window is checkmarked. Select any inactive window to activate it.

9.11 Help menu

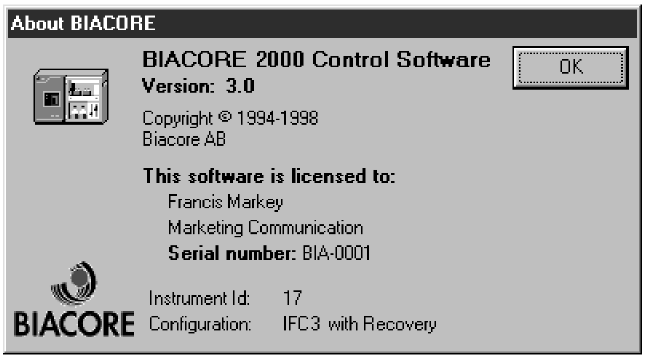


Help on Application Wizards

Opens the on-line help for application wizards.

About

Displays information about the version of BIACORE Control Software currently being used as well as the instrument ID and configuration.


About BIACORE

BIACORE 2000 Control Software
Version: 3.0

Copyright © 1994-1998
Biacore AB

This software is licensed to:
Francis Markey
Marketing Communication
Serial number: BIA-0001

Instrument Id: 17
Configuration: IFC3 with Recovery

A. System specifications

A.1 System performance

Measuring range

The range of BIA measurements in practice is highly dependent on the experimental conditions (ligand size and immobilization level, analyte size and concentration, bulk refractive index effects etc). The figures below are typical values. Ranges may be significantly wider under favourable conditions.

Concentration measurement

For analysis times <15 minutes, precision $\leq$5% CV dose:

High molecular weight analytes (10^4 -10^6 g/mole)

direct assay	typically 10^{-5} - 10^{-9} M
sandwich assay	typically 10^{-3} - 10^{-11} M

Low molecule weight analytes (<5000 g/mole)

inhibition assay	typically 10^{-3} - 10^{-9} M

Affinity measurements at equilibrium

K_A	typically 10^4 - 10^{11} M^{-1}

Kinetic measurements

High molecular weight analytes (10^4 -10^6 g/mole)

k_a	typically 10^3 - 10^6 M^{-1} s^{-1}
k_d	typically 10^{-5} - 10^{-1} s^{-1}

Precision of immobilized protein

Between flow cells on one chip

sequential method	typically < 5% CV
serial method	typically < 2% CV

Between sensor chips	typically < 10% CV

A.2 Technical specifications

Processing Unit

Dimensions (l × w × h)	760 × 350 × 610 mm
Net weight	50 kg
Mains supply (type 1)*	100-120 V~ / 220-240 V~ switched 47-63 Hz, 580 VA max Fuses 2 × T6.25A / 2 × T3.15A
Mains supply (type 2)*	100-230 V~ Autorange 47-63 Hz, 400 VA max Fuses 2 × T5.0A H
Memory protection	lithium backup battery
Ambient temperature range	15-35°C
Ambient humidity	20-85% RH
Safety standard	electronics designed to IEC requirements
Acoustic noise level (LEQ)	< 60 dB(A) measured using standard procedures

Detection Unit

Light source	LED (Light Emitting Diode)
Wavelength	760 nm
Band width	< 5 nm
Refractive index range	1.33-1.36 (0-33,000 RU)
Baseline drift	typically < ±0.3 RU/minute
Data output	
one flow cell	0.1, 1 or 10 measurements/s (0.1-10 Hz)
two flow cells	0.1, 1 or 5 measurements/s (0.1-2 Hz)
four flow cells	0.1, 1 or 2 measurements/s (0.1-5 Hz)
Noise	typically < 0.3 RU RMS (10 Hz) < 0.3 RU RMS (1 Hz)
Temperature range	4 - 40 °C (not lower than 20 °C below ambient temperature)
Temperature stability	Better than ±3 × 10^{-3}°C/min
Temperature accuracy	< ±0.5 °C (entire temp. range) < ±0.1 °C at 25 °C

* Instruments are fitted with one of two power supply types. Refer to the mains input panel to determine the specifications for your instrument.

Temperature equilibration time	
after 5 °C step	typically < 60 minutes
after 10 °C step	typically < 120 minutes

Pumps

Flow rate range	1-5000 µl/min (1-100 µl/min user-controlled)
Increments	Steps of 1 µl/min
Flow rate accuracy	< ±2% (1-100 µl/min, liquid at 25°C)
Flow rate precision	< 1% (1-100 µl/min, liquid at 25°C)
Pressure range	< 0.2 MPa
Tubing	
Inlet (both pumps)	PEEK, i.d. 0.75 mm
Outlet (continuous flow)	PEEK, i.d. 0.5 mm
Outlet (autosampler)	Tefzel, i.d. 0.8 mm

Autosampler

Dispensing volume range	5-500 µl (including air segment)
Increments	Steps of 1 µl
Accuracy	< 1% deviation for volumes ≥50 µl
Precision	
50 µl	< 0.4% CV
5 µl	< 2% CV
Carry over between vials	typically < 1 µl without septum typically < 0.1 µl with septum
Sample capacity per rack:	
Microtitre plate*	96 (0.2 ml/well)
Rack type A	5 vials, diam 16 mm (4 ml) 12 vials, diam. 9 mm (1.5 ml) 40 vials, diam. 7 mm (0.7 ml)
Rack type B	60 vials, diam. 9 mm (1.5 ml)
Rack type C	24 vials, diam. 10.8 mm (Eppendorff-type 1.5-2 ml)
Maximum vial height	55 mm
Optional thermostatting:	¼" connections maximum inlet pressure 0.2 MPa

*Note: Only some brands of microtitre plates are compatible with the autosampler. Call your Biacore technical support representative for more information.

Integrated μ-Fluidic Cartridge (IFC)

Flow cells	
number	4
Valves	Pneumatic membrane valves
Injection volume to flow cell	5 -750 μl, steps of 1 μl
Carry over from previous injection	< 0.1%
Sample loop capacity	120 μl (maximum injection volume 100 μl)

Channel and valve configuration allows access to flow cell 1, 2, 3, 4, 1+2, 1+2+3 or 1+2+3+4 from loop or injection port.

For injection volumes and sample consumption see Section 4.4.6.

Sample recovery

Volume	5 - 350 μl (>25 μl for best precision)
Dilution of recovered sample	see Technical description below
Carry-over	< 0.1 % from previous recovered sample

Fractions

Volume	5 - 325 μl (>25 μl for best precision)
Maximum number	up to 96 (limited by number of vials in one autosampler rack)

A.3 Technical description

This section describes technical aspects of BIACORE 2000. The information here is not required for normal instrument use.

Fuses

External fuses are located in the mains connector on the left side of the instrument and may be replaced by the user. For required fuse ratings, see Specifications above.

Internal fuses are located inside the processing unit. These fuses may only be replaced by approved Biacore service personnel.

LED display

The LEDs on the instrument front panel indicate the processing unit status as follows:

- READY (green)

No light	Power off
Steady light	The instrument is ready for use
Flashing light	Communication with PC is broken

- Error (red)

No light	OK or power off
Steady light	Unrecoverable hardware error

- Temperature (Yellow)

No light	Temperature control disabled
Steady light	Detection unit maintained at set temperature
Flashing light	Temperature equilibrium not reached

- Sensor chip (green)

No light	No sensor chip in position
Steady light	Sensor chip docked and ready
Flashing light	Sensor chip in position, undocked

- Run (green)

No light	BIACORE at standby or power off
Steady light	Continuous flow

Detection unit

The optical detection unit uses four separate linear arrays of light-sensitive diodes, one for each flow cell. The spacing between diodes corresponds to a difference of approximately 0.1° in incident light angle. Computer algorithms automatically calculate the angle at which minimum reflection occurs (the SPR angle) with a resolution of about 10^{-4}°.

Constant temperature at the detector is maintained by continuous circulation of air in a thermally insulated box. The temperature is regulated by Peltier elements, which are controlled by temperature sensors in the optical unit close to the flow cells and in the circulated air.

Flow cells

The volume of each flow cell is about 0.02 µl. The total sensor chip area in each flow cell is 1.2 mm^2.

The focused incident light illuminates an area measuring about 1.6 mm along the flow cell (in the direction of flow). The detector array reads reflected light from a width of 0.16 mm across the flow cell.

The dead volume between the inlet valve and the flow cell is ≈0.5 µl. In serial injection mode the dead volume is ≈0.5 µl to flow cell 1 and an additional 0.3 µl per subsequent flow cell. This gives a time delay in sensorgrams between flow cells corresponding to about 1 s at 20 µl/min (see also Section 8.2.2).

Pumps

The two pumps can deliver flow in the range 1-5000 µl/min. User-controlled flow rates are limited to 1-100 µl/min. Higher flow rates are used for certain system routines. The pumps are controlled by a stepper-motor which delivers 3 nl/step with a 500 µl pump syringe.

The pumps are mounted vertically to aid removal of air bubbles at start-up and eluent change. The right side of the valve at the top of each pump is the inlet side. Tubing from the buffer bottle is connected to the inlet of both pumps. The outlet from the continuous flow pump (left side of the valve) is connected to the connector block inlet. The outlet from the autosampler pump is connected to the needle in the delivery arm of the autosampler.

Material for inlet tubing to both pumps and outlet tubing from the continuous flow pump is PEEK, which gives extremely low gas permeability. Outlet tubing from the autosampler pump is Tefzel.

Autosampler

The tubing between needle and pump (ID= 0.8 mm) has a volume of ≈600 µl and acts as a container for aspirated liquid. The maximum aspirating volume is 500 µl (including air segments). The recommended range is 5-450 µl. The BIACORE 2000 control software commands (e.g. `DILUTE`, `TRANSFER`, `MIX` and the various `INJECT`s) are designed to maintain 1-3 air segments of about 5 µl.

The volume in the needle is < 6 µl (ID = 0.3 mm). In order to avoid pressure differences while aspirating/dispensing liquid to a septum-capped vial, an air inlet groove is cut in the needle.

A vial dislodger on the needle positioning arm ensures that vials are not lifted out of position by friction between the needle and the vial septum when the needle is raised.

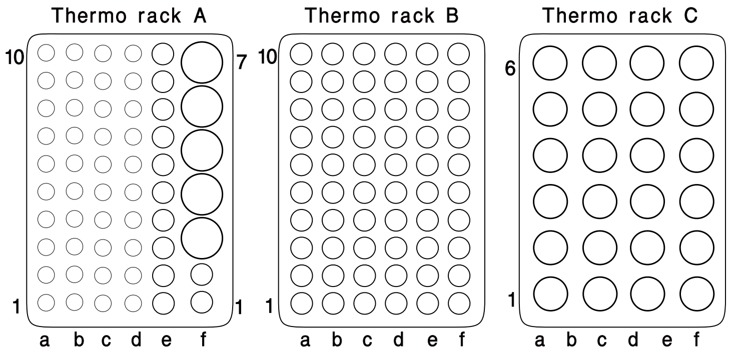
The positioning work space is 170 × 104 × 55 mm (l × w × h) (i.e. no vials, including caps, may be higher than 55 mm). Maximum total travel time between any two positions is approximately 3-4 s.

For thermostatting the autosampler rack base, circulate temperature controlled liquid (water or <20% ethanol in water) through the thermostat inlets on the right-hand side of the instrument. The maximum inlet pressure for thermostatting the rack base is 2 bar (0.2 MPa).

Note: At high ambient humidity, condensation may be considerable on cooled parts, and special care should be taken to insulate connecting tubing and connections. Capped vials should be used to avoid condensation in samples.

Thermo-racks

Three types of thermo-racks are available with vial positions as illustrated below. Note that row f on thermo-rack A has only 7 positions.



Supported microtitre plates have 8 rows (a-h) of 12 positions per row.

IFC

Liquid flow is directed through the IFC channels by pneumatic micro pinch valves operated by air pressure from an internal compressor.

Channels from the air bus to the IFC valves are filled with glycerol to prevent diffusion of air from the air bus into the IFC channels.

The IFC channels are moulded in silicone plastic. The smallest channel cross-sectional area is 0.015 mm^2, so that a volume flow rate of 10 µl/min corresponds to a linear flow rate of ≈1 cm/s.

Recovery and fractions

The volume of the recovery channel between the flow cell and the recovery cup is approximately 15 µl. This channel is emptied of buffer when not in use.

When sample recovery is requested, the injected sample is chased through the recovery channel with running buffer so that the whole injected volume is transferred to the recovery cup. Sample dispersion at the boundary between sample and chasing buffer will result in a slight dilution of the recovered sample (by forward dispersion of buffer) and loss of sample (by backward dispersion of sample). The extent of this dispersion effect will depend primarily on the flow rate. Absolute dispersion effects are higher at higher flow rates.

When recovery is complete, the recovery channel is washed to waste in the reverse direction with buffer from the autosampler pump, and then emptied of buffer by introducing air with the autosampler pump. The recovery cup is washed with running buffer or with solution from a user-specified rack position. A small volume of buffer will remain on the walls of the recovery cup and channel and will contribute to dilution of the next recovered sample. The extent of this dilution will depend largely on the wetting properties of the recovery cup with the buffer used.

For fraction collection during the dissociation phase, the first fraction will be diluted by buffer wetting the walls of the recovery channel and recovery cup. Dilution of following fractions is negligible.

Automatic preparation and finishing routines

Automatic preparation and finishing routines wash the liquid handling system at the beginning and end of methods, method cycles, manually controlled sensorgrams and injections. The summary below does not include command-specific operations, such as aspirating sample before an injection and `EXTRACLEAN`, `DIPNEEDLE`, `FRACTIONS` and `RECOVER` operations.

Method control	
Start method (`MAIN`)	Fresh buffer is delivered to IFC and wash station. The flow is set to 10 µl/min and all IFC valves are opened.
Start method cycle (`APROG`)	The continuous flow pump is filled and the sample line in the IFC is primed with fresh buffer. Flow rate and flow path are set.
Start injection	Pumps are refilled if required (as estimated from buffer consumption during the injection).
End injection	The needle and IFC sample line are washed.
End method cycle (`APROG`)	The IFC sample line and flow cells are washed with buffer. Any flow path changes requested with `FLOWPATH` in the `APROG` are restored. Buffer flow is set to 10 µl/min.
End method (`MAIN`)	The IFC sample line and flow cells are washed with buffer. Flow is set to zero and all IFC valves are opened.
Manual control	
Start sensorgram	A short pulse of buffer is pumped through all flow cells. The needle is washed.
Start injection	Pumps are refilled if required (as estimated from buffer consumption during the injection).
End injection	The needle and IFC sample line are washed.
End sensorgram	All IFC valves are opened. Flow is set to zero and pumps are reset to starting position. IFC channels are washed if an injection has been performed.

Important notes

Automatic routines are kept to a minimum under manual control to reduce the times taken for sensorgram start/end and injections. It is the user's responsibility to ensure that appropriate wash commands are executed before sample injection under manual control. Remember in particular to perform a **Wash** with **Recovery** option before injections which use **Fractions** or **Recover**. If this is not done the recovery line will not be emptied and the sample delivererd to the recovery cup will be diluted with up to 15-20 µl buffer.

B. Application guidelines

Low signal noise coupled with multichannel analysis capabilities give BIACORE 2000 excellent sensitivity characteristics, allowing detection and measurement of low molecular weight analytes and weak interactions. Careful experimental design is however needed to exploit this potential to the full. This chapter gives some guidelines for obtaining the best performance from BIACORE 2000, and lists method constructions for typical application situations.

Follow the recommendations below for best results in experiments where high sensitivity is required. With suitable precautions, BIACORE 2000 can reliably detect relative responses down to 10 RU or even lower. Bear in mind, however, that most applications work well above the limits of sensitivity of the instrument, and will give fully adequate results without extensive tuning of experimental parameters.

B.1 General considerations

The main factors which may interfere with real-time measurement of interaction processes are listed below. Some or all of these factors may be irrelevant for some applications, depending on the application purpose and experimental conditions.

- Sample dispersion, i.e. dilution of sample by mixing with buffer in the autosampler tubing or IFC sample loop. This is minimized by the IFC construction and injection technique, but must be taken into consideration for critical applications. The effects of sample dispersion are illustrated in Section B.5.3.

- Bulk refractive index effects, i.e. difference in bulk refractive index between sample and running buffer. If the sample has a higher refractive index than the buffer, this will give a positive response during injection which may resemble a binding event. A negative bulk effect may conversely obscure binding. On-line reference subtraction helps to minimize the effects of bulk refractive index changes.

- Mass transfer effects, i.e. limitations on the rate of transport of analyte between bulk solution and the sensor chip surface. These effects are deleterious to kinetic measurements but are desirable in concentration assays (see the BIAapplications Handbook, Chapter 6).

- Matrix effects, i.e. changes in extension of the dextran matrix on the sensor chip surface as a result of changes in pH or ionic strength.

The effect of these variations is usually small in terms of absolute response, but may be significant in experiments where highest sensitivity is required. Matrix effects may have time constants ranging from seconds (for e.g. small changes in pH or ionic strength) to minutes or even hours (larger changes as for example following regeneration with extreme conditions). Matrix effects are normally directly related to the mass concentration of immobilized ligand, with greater effects at higher concentrations.

- Conformational changes in the immobilized ligand resulting from changes in buffer composition. These changes may resemble matrix effects, but are caused by changes in the ligand rather than in the dextran matrix.
- Sample depletion between flow cells in multichannel applications, and in single flow cells for sensitive kinetic measurements.
- Carry-over of sample between injections, between vials or between flow cells.
- Unwanted binding of analyte to the surface (giving a false positive binding) or to the autosampler tubing or IFC channel walls (resulting in sample depletion).
- Temperature drift, which may obscure slow association or dissociation events.

B.2 Types of application

Applications can be broadly divided into two groups, with significantly different requirements in experimental design:

- Applications relying on relative response levels. A baseline is established before the injection, and the response level is measured at a fixed time either during or after the injection. Examples of this type of application are epitope mapping, identification of binding partners and many concentration assays. For these applications, sample dispersion at the beginning and end of the injection is usually irrelevant. If the relative response level can be measured after the injection (dissociation of bound analyte from the surface is slow), bulk refractive index effects are eliminated. Mass transfer effects are generally an advantage for this type of assay.
- Applications relying on measurement of interaction rates, typically determination of kinetic parameters for the interaction or measurement of e.g. enzymatic activity. Here, sample dispersion and bulk refractive index effects should be reduced as much as possible, since these effects will obscure rapid binding and dissociation events. Mass transfer limitations are undesirable, except in rate-based concentration measurement, where mass transfer limitation is an advantage (see the BIAapplications Handbook, Chapter 6).

Carry-over and non-specific binding are undesirable in both types of application.

B.3 Manual versus automated operation

For best performance, always run analyses in automated mode, i.e. under method control or from application wizards. There are a number of routines used in methods and wizards (e.g. liquid system wash at the end of each cycle) which help to ensure high performance and which can easily be missed under manual control. For example, it is natural to run a series of injections in a single sensorgram under manual control, thereby losing the automatic cleaning routines between each method cycle.

In addition, automatic control provides a more complete documentation of the run, since the method or wizard template is saved with the result file.

Use manual control for development work, testing immobilization conditions, ligand level and so on.

Most of the recommendations below assume that analyses are performed under method control.

B.4 Special features of BIACORE 2000

The artifacts discussed briefly above can be circumvented or compensated by suitable control experiments in all BIA instruments. However, multichannel analysis in BIACORE 2000 offers special facilities for in-line control experiments.

B.4.1 Gradient surfaces

By immobilizing different levels of ligand in several flow cells and then analysing the interaction of analyte in a single injection over all surfaces, reliable comparisons can be made between interactions at different ligand densities. A series of flow cells prepared in this way is referred to as a *gradient surface* (it is strictly a step gradient surface since the surface is discontinuous between the flow cells).

Gradient surfaces are useful in two contexts:

- In optimizing the ligand density for kinetic measurements and other analyses. A range of ligand densities can be screened with a single injection of analyte, saving sample and time.

- In measuring interactions with low molecular weight analytes or weak binding. The signal from these interactions is often obscured by bulk refractive index effects, either because the response is low or because both association and dissociation are very rapid. Comparison of the response at different ligand densities can distinguish between binding and bulk refractive index effects, since the former varies with the ligand density while the latter may be taken as constant since it is in the range of a few hundred RU. Comparisons made from single channel measurements are often difficult because of minor variations in the response between separate sample injections. The use of a gradient surface to improve the detection limit is described in Application Note 307 from Biacore AB.

B.4.2 In-line reference cell

The surface in one flow cell can provide a reference cell in multichannel analyses. This reference may be a blank cell for correction of bulk refractive index effects, temperature drift, non-specific binding etc, or a “functional blank” aimed to eliminate a particular feature of the biomolecular interaction being studied. On-line reference subtraction provides direct display of the blank-corrected results. See Section B.5.5 for practical considerations concerning reference cells.

B.5 Experimental considerations

B.5.1 Sample preparation

In preparing the analyte sample, try to reduce the difference between sample and running buffer composition as much as possible. Differences in refractive index give bulk effects which can obscure rapid binding events. Differences in pH and ionic strength can give matrix effects which may take from seconds to hours to stabilize. The extent of matrix effects (if any) will vary with the nature and amount of the immobilized ligand.

If the sample contains additives such as detergents in known concentration, add the same concentration to the running buffer.

If differences in composition between sample and running buffer cannot be avoided and matrix effects are suspected (seen as a relatively long-term drift in the signal), include a `WAIT` commands in the `APROG` to allow signal stabilization. You may also use a reference cell to compensate for the matrix effects. Ideally, the reference cell should contain a control ligand immobilized at the same level as the true ligand. It is otherwise difficult to compensate fully for matrix effects using a reference cell since the extent and rate of matrix effects are sensitive to the amount of immobilized material.

For samples that will be kept for relatively long periods (> 1h) in the autosampler, use capped vials to eliminate both sample evaporation and condensation inside the vials (particularly if the autosampler racks are cooled).

B.5.2 Ligand immobilization

Amount of immobilized ligand
The optimal level of immobilized ligand depends on the purpose of the analysis. In general, use the lowest reasonable level for kinetic analyses to minimize the effect of mass transfer limitations. For concentration measurement, the optimal ligand level is often related to the required dynamic range of the assay. See BIAapplications Handbook, Chapters 5 and 6 for further details.

Preparing gradient surfaces
Gradient surfaces for multichannel applications can be conveniently prepared by using variable contact times for either the surface activation step or the ligand binding step. The choice of which step to vary will depend on the characteristics of the immobilization procedure: with standard amine coupling, varying the surface activation time is generally recommended for controlling the amount of immobilized ligand.

At low flow rates, sample depletion between flow cells can become significant. Flow rates of 20 µl/min or higher during ligand immobilization are recommended to minimize sample depletion.

A gradient surface created using variable contact times in the immobilization procedure has the highest ligand concentration in the first flow cell. It may seem that this would aggravate the problem of sample depletion for kinetic measurements, where accurate knowledge of the analyte concentration is required. However, kinetic measurements use high flow rates and low ligand levels, so that sample depletion is rarely a problem. If you do want to create a gradient surface with the lowest immobilization level in the first cell, perform the immobilization in each flow cell separately.

Immobilizing different ligands for multichannel applications
Multichannel applications may be designed with different ligands immobilized in different flow cells, for monitoring multiple analytes in complex samples. In preparing surfaces of this kind, it is essential to avoid contamination of the surface of one flow cell with ligand from another flow cell. The surfaces should be prepared in separate cycles (either manual runs or separate APROGs in a method), the automatic cleaning procedures between cycles will eliminate carry-over. ***The apparently convenient technique of activating all surfaces with a single multichannel injection and then injecting the different ligands separately is not recommended since there is significant risk for carry-over of ligands between flow cells. In addition, the delay between activation and coupling should be kept as short as possible since the NHS-esters on a surface activated with NHS/EDC break down over time.***

B.5.3 Flow rate

The flow rate influences sample consumption, sample dispersion at the beginning and end of the injection, sample depletion during the injection and mass transfer of analyte to the surface. The optimal flow rate is a function of these three factors in combination. In general:

- Use high flow rates (>30 µl/min) for kinetic experiments to reduce sample depletion and mass transfer limitations.
- Use moderate flow rates (5-20 µl/min) for immobilization and for analyses based on relative response levels to reduce sample consumption.
- For multichannel analyses, use higher flow rates than for the corresponding single channel application, to minimize sample dispersion and depletion effects.
- Remember that you can change the flow rate during a run (either manual or method-controlled) if different parts of the run have different flow rate requirements.

Sample consumption

The determining factor in many injection protocols is the contact time of sample with the surface, determined by the injected volume divided by the flow rate. If sample is limiting, a low flow rate may have to be used to achieve the required contact time. When availability of sample is not critical, however, sample consumption is a secondary consideration and the flow rate should be chosen with respect to the other factors discussed here. It is generally better to dilute the sample and use a larger volume and high flow rate than to use a high concentration and a low flow rate.

Sample dispersion

Sample dispersion refers to mixing between sample and buffer at the beginning and end of the injection. The use of air segments separating sample from buffer minimizes dispersion in the autosampler tubing and sample loop, and the design of the IFC minimizes convective mixing at the boundaries during injection. A certain amount of dispersion is however unavoidable, and may be significant in applications which measure fast binding rates. Dispersion is most strongly influenced by the number of air segments, which differs between different injection commands (see Section 4.4.6).

Dispersion may be measured in terms of the finite rise and fall times in response at the beginning and end of an injection of a sample with a high refractive index (e.g. glycerol or sucrose). Rise and fall times are reduced at higher flow rates (see Figure 5-1).

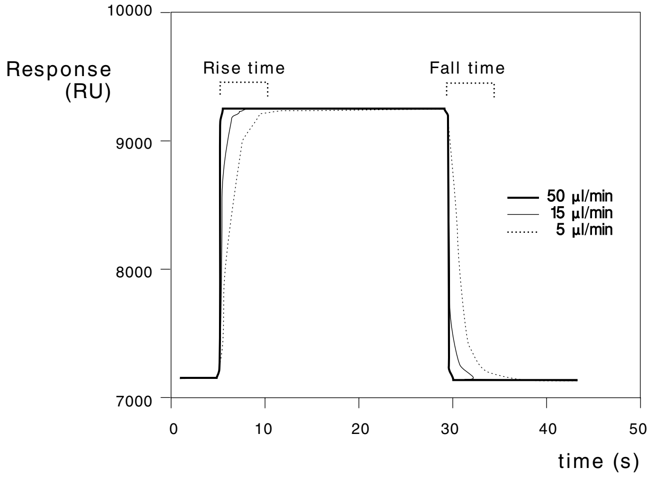


Figure 5-1. *Illustration of sample dispersion. The values indicate the extent of dispersion that can be expected for an injection of sucrose at different flow rates.*

The effects of sample dispersion on bulk refractive index changes can obscure rapid binding and dissociation events at the beginning and end of the injection respectively. In general, the rate constant describing the change from buffer to sample and *vice versa* should be at least 10 times faster than the interaction rate constant for reliable kinetic measurements. Typically, the concentration reaches 99% of the target value in the first flow cell within a volume of about 0.5 µl, so that the critical time scale is about 2 seconds at a flow rate of 15 µl/min. Correspondingly faster events can be detected at higher flow rates. In multichannel applications, the rise and fall times are longer in the later flow cells, and flow rates of 30 µl/min or higher are recommended for kinetic applications on multiple surfaces.

Sample dispersion is most easily visualized in terms of distortion of a refractive index pulse, but it is important to realize that dispersion effects are present even if the refractive index is matched between buffer and sample. Bulk refractive index changes may prevent rapid binding events from being visible in the sensorgram. At least as important however is the effect of sample dispersion on the analyte concentration at the beginning and end of the injection, which can distort the actual binding kinetics by affecting analyte concentration.

Sample depletion

Depletion of analyte from the sample in the flow cell can be significant at low flow rates and high surface binding capacities. Depletion is reduced at higher flow rates: with continuous flow at 30 µl/min the sample in the flow cell is replenished approximately 10 times/second. As a general guideline, sample depletion for an analyte of molecular weight 50,000 daltons will not exceed 4% per flow cell at 30 µl/ml.

Sample depletion is rarely a problem in single channel applications, but may be significant in multichannel analyses where the sample passes over several flow cells in series with the same ligand immobilized. In general, use flow rates above 20 µl/min for multichannel applications to minimize this effect. Sampe depletion is proportional to $(flow)^{-2/3}$.

Mass transfer effects

The dependence of observed interaction rates on mass transfer of analyte to the surface is a complex question which is discussed in some detail in the BIAtechnology Handbook, Chapter 5. As general guidelines:

- For kinetic applications where mass transfer limitations are to be minimized, use a low ligand density and the highest flow rate that is consistent with a reasonable sample consumption.
- For concentration assays where mass transfer limitations are desirable, use as high ligand density and a flow rate that gives mass transfer-limited binding. The assay will then be independent of small changes in ligand activity, and the dynamic range can be controlled directly by varying the contact time.

Note that the effect of flow rate on mass transfer is relatively small (a 10-fold increase in flow rate gives slightly more than a 2-fold increase in mass transfer rate). Minimizing ligand density is the most effective way of reducing mass transfer limitations.

Other considerations

During extended operation at low flow rates (< 5 µl/min), air bubbles may develop in pump tubing and IFC channels. This problem is aggravated by detection temperatures above room temperature. Run the working tool **Prime** or **Flush** to clear air bubbles from the system after prolonged operation at low flow rates. The problem is less marked at higher flow rates. See Section 8.2.2 for further details of how to minimize problems with air bubbles.

B.5.4 Injection techniques

BIACORE 2000 offers a range of application-oriented injection commands as described in Section 4.4.6. These commands differ in capacity (`BIGINJECT`), functionality (`COINJECT`) and the balance of preparation time and injection performance (`QUICKINJECT`, `KINJECT`). Use the command that best suits to your application.

Time-controlled injections
During a method, injections are prepared as soon as the autosampler is available, even if the injection is timed to start at a later point. For example, in the APROG below the injection will be *prepared* as soon as the APROG starts, but will be *started* at time 10 minutes.

```
DEFINE APROG analyse
       FLOW 30
 10:00 INJECT r1a1 30
END
```

For applications requiring high injection performance (low sample dispersion) this can be a disadvantage since the sample waits in the autosampler tubing for about 10 minutes and sample dispersion will increase. To avoid this, program a time-controlled autosampler event (e.g. wash the IFC channels) just before the injection:

```
DEFINE APROG analyse
       FLOW 30
 10:00 WASH s
       INJECT r1a1 30
END
```

With this construction, the injection will not be prepared until the wash is complete. Time-control of the actual injection start is less precise (the injection will start at 10 minutes + time for wash + time for preparing the inject), but the residence time of the sample in the autosampler tubing is kept as short as possible. Use the event log to determine the actual time of injection start.

Note: A timed `RPOINT` command cannot be used for this purpose, since `RPOINT` does not occupy the autosampler and the injection will be prepared at the beginning of the `APROG`.

B.5.5 Using a reference cell

Multichannel detection in BIACORE 2000 allows one cell to be used as an in-line reference, to compensate for bulk refractive index effects, non-specific binding etc. In designing applications with a reference cell to compensate for artifacts, bear the following considerations in mind:

- Use a cell adjacent to the measuring cell as a reference. Adjust the time axis of the reference sensorgram to coincide with the sample cell in BIAevaluation. The delay between adjacent cells in the raw data corresponds to 0.3 µl liquid (2 seconds at 10 µl/min).

- If possible, immobilize a “neutral” molecule in the reference cell at the same level as in the sample cell. Choose a molecule with similar molecular weight and charge characteristics to the ligand. This will help in using the reference cell to compensate for matrix effects.

B.5.6 Reducing carry-over

The term “carry-over” includes a number or potential artifacts:

- transfer of sample between successive injections.
- transfer of sample or buffer between different vials.
- transfer of sample between flow cells.
- incomplete regeneration of the surface.
- leakage of contaminants from surfaces in the liquid handling system.

In general, the first few analysis cycles with a new surface are more sensitive to carry-over and “memory effects” than later cycles when the surface has stabilized. It is often advantageous to run one or more blank analysis cycles (complete cycles including regeneration – use buffer in place of sample if sample is valuable) after preparing a new surface.

Carry-over between injections

This may be reduced by the correct choice of injection command (e.g. do not use `QUICKINJECT` if avoidance of carry-over is essential). Carry-over in the autosampler needle may be further reduced by using the `DIPNEEDLE` command between injections. To eliminate carry-over arising from residual sample in the IFC channels, include an `EXTRACLEAN` command after each injection.

Carry-over between vials and from running buffer

In applications that perform many injections from the same vial, use septum-capped vials and perform a `DIPNEEDLE` in a vial containing the sample buffer before each injection.

Carry-over between flow cells

The flow cells in the BIACORE 2000 IFC are not physically separated by valves on both sides, so that samples injected over long periods into one flow cell will gradually diffuse into the adjacent flow cell. For small molecules (< 1000 daltons), about 1% carry-over has been observed after 30 minutes. Levels will be correspondingly lower for larger molecules with lower diffusion constants. To minimize the effect of this diffusion:

- Do not use variable contact time injections with more than 30 minutes difference in contact time between adjacent cells.

- Include an EXTRACLEAN command before changing the flow path to wash unwanted sample out of the connecting channels between flow cells.
- If you are analysing or exploiting enzymatic reactions on the sensor surface, avoid single flow cell analyses where the adjacent flow cell also has enzyme-sensitive material immobilized. It is very difficult to eliminate diffusion of active levels of enzyme between adjacent flow cells.
- Certain small molecules (e.g. ethanol and other alcohols in low concentrations) appear to diffuse quite easily between flow cells. Do not use higher concentrations than necessary of low molecular weight compounds. We do not recommend multichannel applications where sensitive ligands (e.g. lipid structures which may be disrupted by ethanol) are immobilized in several channels but analyzed in one channel at a time.

Carry-over between cycles
Analyte carry-over between analysis cycles generally indicates insufficient regeneration of the surface, and is generally restricted to sandwich assay techniques for concentration measurement. Note that concentration measurements performed under mass transfer-limiting conditions are insensitive to the amount of ligand and are therefore not affected by small amounts of carry-over between cycles. Optimize the regeneration conditions to minimize carry-over (see the BIAapplications Handbook, Chapter 2).

Carry-over from IFC channels
Certain compounds tend to adsorb to the IFC channel walls and leak out in subsequent injections. One example is formic acid, which is used in some regeneration protocols. If you suspect that carry-over from the tubing and/or IFC channels is a problem, wash the IFC with a WASH i command and then use a WAIT command to allow buffer flow until the baseline is stable after exposure to the compound in question. Run **Desorb** at least every week if biomolecules from samples adsorb to the IFC walls and interfere with analyses.

In clearing carry-over from the IFC channels, time is often more important than the washing volume, since dissolved materials leach out at a characteristic rate. A useful test of carry-over is a so-called "zero injection", i.e. an autosampler transfer and injection using buffer instead of sample:

```
DEFINE APROG zero_inject
       FLOW 30
       WASH n               !wash needle
       TRANSFER B r1a1 450  ! 'B' is buffer
       INJECT r1a1 30
END
```

Use a new glass vial with a septum for the transfer and inject position. With physiological salt concentration in the buffer, a response of about 10 RU from a zero injection is not uncommon, and can be traced to salt deposits on the needle.

B.5.7 Temperature changes

A programmed temperature increase of 5°C takes about 1 hour to stabilize. The equilibration time for cooling by the same interval may be somewhat longer, and depends more on the ambient temperature. Increasing the temperature above ambient can cause problems with formation of small air bubbles in the buffer flow, particularly at >30°C. As a general recommendation:

- For temperature dependence studies below ambient, start at the lowest temperature to minimize equilibration time.

- For temperature dependence studies above ambient, degas buffer thoroughly before use and start at the highest temperature to minimize air bubble problems.

The `TEMP` command in a method causes the system to wait until the temperature is stable before continuing the analysis. The `TEMP` command can only be used in the `MAIN` block in a method, so that temperature changes can only be programmed between cycles. The flow rate is set to 10 by the system between cycles.

For best results include `RINSE` or `PRIME` and `NORMALIZE` directly after the `TEMP` command to clear air bubbles from the flow system and normalize the signal response.

If low baseline drift is important for the application, include a timed `WASH` command at the beginning of the `APROG` to allow for stabilization of the signal at the new flow rate.

The example below illustrates these features in a method for temperature dependence analysis in single channel mode. Note that the `MAIN` block includes a `PRIME` command to clean the system thoroughly and a `NORMALIZE` command to reset the detector at each new temperature. By starting the method with the detector temperature close to the setting for the first cycle, the temperature equilibration delay before the first cycle is minimized.

The method will abort if `NORMALIZE` fails.

```
DEFINE APROG analyse
       FLOW 25
 10:00 WASH s       ! 10 minutes stabilization
      ...
END
```

```
DEFINE LOOP temp_series
       LPARAM %degrees
       TIMES 1
              5
              10
              15
              ...
END

MAIN
    ! Start method at 5°C or below
    LOOP temp_series STEP
       TEMP %degrees
       PRIME
       NORMALIZE r2f1    ! glycerol position
       APROG analyse
    ENDLOOP
END
```

B.5.8 Regeneration

Effective regeneration is necessary to eliminate carry-over of analyte between analysis cycles. Regeneration considerations are discussed in the BIAapplications Handbook, Chapter 2.

With the multichannel IFC construction in BIACORE 2000, it is important to wash the IFC channels thoroughly after regeneration, to avoid carry-over of regeneration solution to the next injection. Two precautions are recommended:

- Place the regeneration injection last in the APROG, so that the automatic cleaning routines at the end of each cycle wash out any residual regeneration solution.

- Include an EXTRACLEAN command after the regeneration injection for added security. This is particularly important if the interaction being studied is sensitive to low concentrations of regeneration solution.

If carry-over of regeneration solution between cycles is a persistent problem in spite of these precautions, include a PRIME command in the MAIN block just before each APROG.

Remember that regeneration solutions can cause relatively slow matrix effects in the ligand surface, which can cause baseline drift at the beginning of each cycle. Introduce a WAIT command at the beginning of the APROG to allow baseline stabilization if this is a problem.

B.5.9 Washing the liquid handling system

During normal operation, the liquid handling system is washed automatically at the end of each cycle where the autosampler is used. For high performance work, there are a number of additional washing facilities:

- `WASH n` washes the autosampler needle.
- `WASH s` washes the injection channel in the IFC.
- `WASH l` washes the sample loop (only relevant with `COINJECT`).
- `WASH f` washes the flow cell area in the IFC.
- `WASH i` washes the injection channel, sample loop and flow cell area in the IFC. The process of washing the flow cell area may cause baseline disturbances.
- `FLUSH` and `PRIME` wash the pump tubing and IFC channels at a high flow rate. This can help to remove air bubbles trapped in the system.
- `EXTRACLEAN` performs an additional wash of the IFC channels immediately after an injection and may be used to minimize carry-over between injections.

The `WASH` and `WASHREC` commands can be used either in the `MAIN` block or in an `APROG` in a method. `FLUSH` and `PRIME` may only be used in `MAIN`. `EXTRACLEAN` must be placed immediately after an injection and may only be used in an `APROG`.

C. Chemical resistance and recommended solutions

This appendix contains chemical resistance information for BIACORE 2000 components together with recommended buffers and solutions for applications.

Safety material charts for the chemicals in the Amine Coupling Kit are included.

Chemical abbreviations which are not explained in this appendix may be found in the Biochemicals, Organic Compounds for Research and Diagnostic Reagents catalogue from Sigma Chemical Company, St Louis, Mo, USA.

Terminology

Continuous means that the substance is harmless to the system and may be used regularly.

Pulse means that BIACORE 2000 can be exposed to the substance for short periods of time separated by buffer or water wash. Resistance has been verified for the sensor chip by injections of up to 50 pulses of 3 min and for the IFC by immersion for up to 30 minutes.

Do not use means that the substance is not recommended for use in BIACORE instruments.

Some solutions can cause relatively slow matrix effects in the ligand surface, which can cause baseline drift at the beginning of each cycle. Introduce a `WAIT` command at the beginning of the `APROG` to allow baseline stabilization if this is a problem.

C.1 General chemical resistance

The table below lists some of the most common solutions used for BIACORE 2000 applications are found. The table may be used as a guide for usage of other substances similar to the ones listed. Resistance to chemicals at higher concentrations than those specified in the table is not guaranteed.

Sparing use of organic acids and alcohols is generally recommended. However, formic acid at concentrations up to 70% and ethanol at concentration up to 70% can be used in short pulses.

Solution	**Concentration**	**pH**	**Resistance**
Acetic acid	100%		do not use
Acetonitrile	50%	7.5	pulse
Ammonium sulphate	2 M		continuous
Benzamidine	10 mM	6.2	continuous
Borate buffer	10 mM	8.5	continuous
Chlorhexidine	0.0002%		pulse
Chlorobutanol	0.05%		pulse
Citrate buffer	100 mM	3.0	continuous
Dithioerythritol in sodium bicarbonate	100 mM/100 mM	8.8	pulse
DMF	50%		pulse
DMSO	100%		do not use
DMSO	50%		pulse
DMSO	10%		continuous
Ethanol	70%		pulse
Ethanol	10%		continuous
Ethanolamine-hydrochloride + sodium hydroxide	1 M	8.5	continuous

Ethylene glycol	50%	10	pulse
Formic acid	70%		pulse
Formamide	40%		continuous
Formamide + dextran sulphate	50%/10%		pulse
Formic acid	70%		pulse
Germall II[*] in HEPES buffer	0.25%/10 mM	7.5	continuous
Glycine-HCl	1.5 M	2.5	pulse
Glycine +lithium sulphate	0.1 M/0.033 M	2.5	pulse
Glycine + sodium chloride	1.5 M/3 M	9.0	continuous
Glycerol	100%		do not use (high viscosity)
Glycerol	50%		continuous
Guanidine hydrochloride	6 M		continuous
HBS[†]	7.4		continuous
Hydrochloric acid	100 mM	1.0	pulse
HEPES	10 mM	7.5	continuous

[*] Commercially available solution for sanitization. Follow instructions for use recommended by the manufacturer.

[†] 10 mM HEPES pH 7.4 containing 0.15 M NaCl, 3.4 mM EDTA and 0.005% v/v Surfactant P20.
Surfactant P20 is a polysorbate of high quality specially prepared for use in real-time BIA. It is available from Biacore AB.

Isopropanol	100%		do not use
Lithium lauryl sulphate	0.5%		continuous
MES	10 mM	6.5	continuous
Methanol	100%		do not use
Methanol	50%		pulse
Phosphoric acid	100 mM		pulse
Sodium acetate buffer	10 mM	4.0	continuous
Sodium carbonate	10 mM	10.0	continuous
Sodium chloride	1 M		continuous
Sodium hydrogen carbonate	100 mM	9.2	continuous
Sodium hydroxide	100 mM	13.0	pulse
Sodium hydroxide + acetonitrile	100 mM, 20%		pulse
Sodium hydroxide	0.1M	13.0	pulse
Sodium hypochlorite	5%		pulse
SDS in water	0.5%		pulse
Sodium perchlorate	6 M		pulse
Urea	8 M		pulse

C.2 Buffer solutions

The choice of buffer used in an application depends on the properties of the interactants and with the aim of the experiment. This section lists examples of buffers suitable for use with BIACORE 2000. The table may be used as a guide when designing methods.

Note: Buffers containing primary amines should be avoided for the amine coupling reaction.

Solution	Concentration	pH	Resistance
ACES	50 mM	6.8	continuous
ADA	50 mM	6.6	continuous
Barbital+ANS+P20	75 mM/20 mM/ 0.005%	8.6	pulse
BES	50 mM	7.1	continuous
BICINE	50 mM	8.3	continuous
BIS-TRIS	50 mM	6.5	continuous
BIS-TRIS-PROPANE	50 mM	9.0	continuous
Borate	100 mM	8.8	continuous
CAPS	50 mM	10.4	continuous
CHES	50 mM	9.3	continuous
Citrate	50 mM	3.0	continuous
EPPS	50 mM	8.0	continuous
Glycine	50 mM	2.3	pulse
Glycine	50 mM	9.6	continuous
HEPES	50 mM	7.5	continuous
MES	50 mM	6.1	continuous
MOPS	50 mM	7.2	continuous
MOPSO	50 mM	6.9	continuous

Phosphate	50 mM	7.5	continuous
PIPES	50 mM	6.8	continuous
POPSO	50 mM	7.8	continuous
TAPS	50 mM	8.4	continuous
TAPSO	50 mM	7.6	continuous
TED	50 mM	7.5	continuous
TES	50 mM	7.5	continuous
TRICINE	50 mM	8.1	continuous
TRIS-HCl+ANS+P20	75 mM/5 mM/ 0.05%	8.0	continuous
TRIZMA BASE	50 mM	8.1	continuous

C.3 Regeneration solutions

This section lists regeneration solutions which are compatible with BIACORE 2000. Regneration conditions should be chosen with respect to the chemical properties of the immobilized ligand (see BIAapplications Handbook for further discussion).

Solution	Concentration	pH	Resistance
Acetonitrile	50%	7.5	pulse
Acetonitrile in sodium hydroxide	20%/100 mM	10.4	pulse
Hydrochloric acid	100 mM	1.3	pulse
Ethanol	70%		pulse
Ethanol	25%		pulse
Ethylene glycol	50%	10	pulse
Formic acid*	70%		pulse
Glycine	1.5 M	2.5-3.5	pulse
Phosphoric acid	100 mM		pulse
Sodium carbonate	200 mM	11.5	pulse
Sodium chloride	1 M		continuous
Sodium hydrogen carbonate	100 mM	9.2	continuous
Sodium hydroxide	250 mM		pulse
Sodium hydroxide + sodium chloride	100 mM/3 M		pulse

* 70% formic acid may dissolve to some degree in the IFC channel walls and leach out in subsequent injections. If formic acid is used for regeneration, be sure to wash the system until the baseline is stable before performing the next injection.

C.4 Disinfectants

Sanitization should be performed for decontamination after exposure to infectants and before storing the system to inhibit bacterial growth during storage. Choose a disinfectant which has the best effect on the type of contaminating material at hand. Be aware that the sanitization procedure may destroy the sensor chip.

Solution	Concentration	pH	Resistance
Ethanol	70%		pulse
Chlorhexidine	0.0002%		pulse
Chlorobutanol	0.05%		pulse
Germall II	0.25%		pulse
Glycine + lithium sulphate	0.1 M/0.033 M	2.5	pulse
Sodium hydroxide	0.1M	13.0	pulse
Sodium hypochlorite	5%		pulse

D. Status and event log messages

D.1 Status window messages

This section lists messages which may appear in the status window and which require explanation or user action. Self-evident messages like **Method running** are not listed.

Not Connected to Instrument
This instance of the software is not connected to the instrument. This is normal for second or subsequent instances of the software, and for instances started from the Windows File Manager (see Section 4.1.2).

Instrument Offline
The software has attempted to connect to the instrument but has failed. This typically occurs if the instrument is turned off or the communication cable is not properly connected. Connection will be established automatically if the communication error is rectified.

Instrument Disconnected
The software has been logically disconnected from the instrument and will not try to establish a new connection. This is normally caused by pressing Ctrl-Break (Abort) during self-diagnostics or by an invalid hardware configuration in PC or instrument. This message is normally preceded by an explanatory message in a message box.

Performing Self Diagnostics
This message appears when the control software is started and establishes connection with the instrument.

Instrument Self Diagnostics Failed
Validation of instrument software and configuration or the resulting automatic program loading failed. This may be caused by instrument startup failure and/or invalid instrument software. The software has no connection to the instrument.

Initiating System
When connection has been established and instrument software validated the software is initialised according to the currently connected instrument. This initialisation includes verifying that the hardware configuration (in the PC) corresponds with instrument configuration.

Initialization failed
The control software initialization failed. The software will be connected to the instrument, but system control may be unreliable. The problem should be identified and corrected before the instrument is used. The message will remain displayed until the software is closed or a connection is successfully initialized.

D.2 Event log messages

This section explains the messages which may be entered in the event log. Messages are listed in alphabetical order.

Connection lost
Indicates that the run was terminated because connection between the PC and the instrument was lost.

Data collection
Records the data collection rate in Hz at the beginning of the method cycle or manual sensorgram and when a change in data collection rate is requested.

Data reduction in sensorgram
Indicates that data reduction has occurred in the sensorgram from the specified flow cell. See Section 9.8 for a description of data reduction.

Detection mode
Identifies single- or multichannel detection mode. Note that this is independent of the flow path.

Dilute: End
Indicates completion of a dilute operation.

Dilute: Start
Indicates the start of a dilute operation. The final concentration and source and destination positions are recorded (e.g. **Dilute: Start 30% RIA1+R1A2->R1A3** indicates that position r1a3 will contain a mixture of 60 µl from r1a1 and 140 µl from r1a2).

Dissociation: Ready
Indicates the end of the dissociation time specified in a `KINJECT` command.

End OPERATION
Indicates completion of an injection requested from the manual control interface, including th automatic clean-up routines following the injection. Note that **End INJECT** does not indicate the end of the actual sample injection (see **Inject: Ready**).

<Error messages>
Software errors are recorded in the event log with the appropriate information. Errors arising in the instrument microprocessor are documented by exit codes. Errors arising in the control software are described by a text message.

Flow
Indicates the current flow rate for continuous flow.

Flow path
Indicates the current flow path. Note that this is independent of the detection mode.

Flowfill:Ready
Indicates the end of a pump refill operation. Normally 6-7 seconds after **Flowfill:Start**.

Flowfill:Start
Indicates the start of a pump refill operation.

Fraction: Ready
Indicates the end of a collected fraction. The autosampler position for the fraction is recorded.

Inject: Accumulated Volume
Records the accumulated injected volume for a manual injection. This message is generated whenever the loop is reloaded and at the end of the injection.

Inject: Ready
Indicates the end of sample injection over the sensor surface. The volume of sample injected is recorded. The type of injection command is recorded for injections performed under manual control.

Inject: Start
Indicates the start of sample injection over the sensor surface. The flow cell for which the injection applies and autosampler position for the sample are recorded. For a variable contact time injection in multichannel mode, an **Inject: Start** message is recorded as each succcessive flow cell is switched into the flow path. The type of injection command is recorded for injections performed under manual control.

Inject: Stop
Indicates that an injection was stopped by the user in manual control. The volume of sample injected is recorded. The type of injection command is recorded for injections performed under manual control.

Load Loop: End
Indicates completion of a loop loading operation.

Load Loop: Start
Indicates the start of a loop loading operation (manual injection only). The loading volume and autosampler position are recorded (e.g. **Load loop: Start 80 µl R1A1**).

Low data quality
Indicates low data quality in the sensorgram from the specified flow cell. This message is only displayed in the extended event log.

Mix: End
Indicates completion of a mix operation.

Mix: Start
Indicates the start of a mix operation. The mixing volume and autosampler position are recorded (e.g. **Mix: Start 120 µl R1A1**).

Normal data quality
Indicates that data quality has returned from low to normal in the sensorgram from the specified flow cell. This message is only displayed in the extended event log.

Run aborted
Indicates that the user aborted the run by pressing Ctrl-Break (see Section 4.4.14).

Set temperature
Records the requested measuring temperature at the time the request is made. No record is entered if the requested temperature is the same as the actual temperature.

Start <injection>
Indicates the point at which the user requested an inject command in the manual control interface. The type of inject command is recorded in the message. Note that **Start INJECT** does not indicate the start of the actual sample injection (see **Inject: Start**).

Temperature
Records the actual measuring temperature at the beginning and end of the method cycle or manual sensorgram.

Temperature stable
Indicates that the measuring temperature has become stable.

Temperature unstable
Indicates that the measuring temperature has become unstable.

Terminate run
Indicates that the run was terminated as a result of a control software error.

Timestamp
Records the date and time at the beginning of the method cycle or manual sensorgram.

Transfer: End
Indicates completion of a transfer operation.

Transfer: Start

Indicates the start of a transfer operation. The volume transferred and the source and destination positions are recorded (e.g. **Transfer: Start 80 µl R1A1->R1A2**).

Wash: <item>

Indicates the start of a wash operation. The item to be washed is recorded (e.g. **Wash: Recovery R1A1** indicates that the recovery cup is washed with solution from positoin r1a1).

Wash:Ready

Indicates the end of a wash operation.

E. Writing BIACORE 2000 methods

Operation of BIACORE 2000 can be controlled from the PC through ***methods***, written in the BIACORE Method Definition Language (MDL). The language gives full control over pre-programmed BIACORE 2000 operation.

Methods are written as text files, and may be created or modified with any text editor. BIACORE Control Software includes a method editor, similar to Windows Notepad but augmented with some extra features specifically designed to help in writing methods.

Like all programming languages, MDL requires exact syntax (keywords must be spelled correctly, words must be separated by spaces and so on). As an aid to getting methods right, there is a pre-run facility which checks through a method for syntax errors without actually controlling instrument operation. In this way, valuable samples and instrument time need not be lost as a result of syntactically incorrect methods. Methods are always checked for syntax errors before they are run.

This chapter describes how to write BIACORE 2000 methods in MDL. Appendix F provides a full reference to MDL.

E.1 Basic method programming

This section covers the basic requirements for writing methods in MDL. If you are already acquainted with programming techniques, these principles will be largely familiar to you. If you are new to programming, on the other hand, read this section carefully – it will help you to avoid many errors in your methods!

E.1.1 Language syntax

Commands in MDL consist of ***keywords***, ***parameters*** and ***options***.

Keywords are words which are strictly defined in the language, such as `FLOW`: they must always be written out in full, exactly as specified in Appendix F. Keywords which are incorrectly spelled cause an “Unrecognized command” error when a method is prerun. Keywords are written in Appendix F in uppercase letters to identify them as such, but the case used in methods does not matter.

Parameters are values which qualify the action of a command, such as "30" in the command `FLOW 30`. Parameters always follow the keyword for the command to which they apply.

Some commands may also be qualified by ***options***, such as "-f" in the command `FLOW 30 -f` (this option refills the buffer pump). Options generally stand for less commonly used functions.

Permitted values for parameters vary for the different commands. Some commands, like `FLOW`, require a numerical parameter: others, like `INJECT` (to position the autosampler needle) require an autosampler rack position specification followed by a number. Permitted values for the parameters for each command are listed in Appendix F.

Each command (keywords with parameters) in a method must be written on a new line, and contained entirely on the line. You can leave blank lines between commands or method blocks if you wish. Within a command, keywords and parameters must be separated from each other by at least one so-called "white-space" character (i.e. space or tabulator character). It does not matter how many white-space characters you use.

E.1.2 Constants and variables

Parameters are values supplied to commands, as explained above. It is most natural to provide these values directly as numbers or character sequences, such as `FLOW 30` or `INJECT r1a1 10`. These values are called ***constants***.

Methods are more flexible if ***variables*** are used instead of constants. A variable is a placeholder for a value, which is assigned an actual value in the current context of the method. For instance, the command `INJECT %sample %vol`, where `%sample` and `%vol` are variables, will inject a volume defined by the value currently assigned to `%vol` from the position identified by the value currently assigned to `%sample`.

Variable names

You choose your own names for variables, with the following restrictions: names may be up to 30 characters long, and contain any of the (English) letters a-z and A-Z, the digits 0-9, and the special characters %, _ and &. Names may not begin with a digit or underscore character. Note that names may not contain spaces. Use underscore characters instead of spaces in names, like `%mixing_position` or `%flow_rate`.

Command keywords, like `FLOW` and `INJECT`, should not be used as names.

A useful convention, followed in this manual, is to start all variable names with a special character such as %. This serves two purposes: there is no risk that you will confuse keywords and variable names (no keywords begin with %), and it is easy to see in a method file which words are variable names.

E.1.3 Comments

You can add comments to your method to help you understand what it is meant to do. Comments are introduced by an exclamation mark (!) and consist of all characters from the exclamation mark to the end of the line. Comments may be written on the same line as commands or on separate lines.

You can also use an exclamation mark to remove commands temporarily from a method. Compare the two method fragments below (refer to Appendix F if you want to understand the details of the command formulations):

```
...
TRANSFER r2a1 r2b1 10
MIX r2b1 200
INJECT r2b1 25
...
```

Transfer 10 µl from position r2a1 to r2b1, mix in r2b1 by aspirating and dispensing 200 µl, then inject 25 µl.

```
...
TRANSFER r2a1 r2b1 10
! MIX r2b1 200
INJECT r2b1 25
...
```

Transfer 10 µl from position r2a1 to r2b1, then inject 25 µl immediately. The MIX command is now a comment and does nothing.

E.2 Method blocks

BIACORE Control Software methods are divided into a number of separate sections or ***blocks***. There are 5 block types, each with a defined function in the method:

Block keyword	*Function*
DEFINE ALERT	Gives the user a chance to terminate or continue the method.
DEFINE APROG	(Analysis program.) Defines the sequence of operations in one run or series of runs. A sensorgram is collected and analyzed.
DEFINE LOOP	Defines the parameters to be used for the repeated operations in a LOOP block.
DEFINE MESSAGE	Displays a message on the screen.
MAIN	Defines the sequence of block execution in a method.

The beginning of a block is defined by a line containing the block keyword and any parameters that may be required. The end of each block is identified by a line containing the single keyword END. All other commands must be written within blocks: in other words, there may only be comments or empty lines between the END terminating one block and the block keyword starting the next.

The MAIN block is required in every method, and may only occur once. All other blocks are optional, and may occur as many times as required. Different blocks of the same type are identified by names.

The order in which blocks are written in a method does not matter.

E.3 Constructing methods

This section shows you how to construct fairly simple methods using MDL, to give you a feel for how to write methods. The examples in Appendix B illustrate complete methods using a range of MDL commands.

E.3.1 The MAIN block

The `MAIN` block defines the sequence of operations in the whole method. This is the only required block in a method, and is the only block which may not be used more than once.

The `MAIN` keyword is not followed by any parameters. The end of the `MAIN` block is identified by the `END` keyword.

Most commonly, the `MAIN` block calls a series of `APROG`s (see Section E.3.2) which constitute the “body” of the method. The `MAIN` block can also set the detection mode and the default flow path for the `APROG`s with the `DETECTION` command (see Section E.4 for more details).

```
MAIN
    DETECTION 2
    APROG immob
END
```

E.3.2 The analysis program The DEFINE APROG block

Simple analysis programs

The sequence of operations for one cycle is written in a `DEFINE APROG` (*analysis program*) block. A method may contain any number of `DEFINE APROG` blocks. Each one is identified by a unique name.

A very simple example sets the flow rate to 20 µl/min and injects 20 µl sample from position r2a1:

```
DEFINE APROG injection
       FLOW 20
       INJECT r2a1 20
END
```

An `APROG` is ***not*** a complete method: it is executed from the `MAIN` block by writing the command `APROG` followed by the identifying name (see Section E.3.1):

```
MAIN
    DETECTION 1
    APROG injection
END
```

The complete method is these two blocks together:

```
DEFINE APROG injection
       FLOW 20
       INJECT r2a1 20
END

MAIN
     DETECTION 1
     APROG injection
END
```

Note that the `DEFINE APROG` block starts with a `FLOW` command, to set the flow rate. ***Every `APROG` must set the flow rate at the beginning of the `APROG`.***

Every `APROG` starts with an automatic initiation sequence that refills the buffer pump if necessary and starts sensorgram recording, and ends with a termination sequence that washes the autosampler needle and IFC channels and resets all IFC valves.

A `MAIN` block can call more than one `APROG` to run multiple analysis cycles. Each `APROG` represents one cycle in the method and generates one or more sensorgrams depending on the detection mode. The example below immobilizes a ligand to the sensor chip surface using amine chemistry, then injects analyte sample over the ligand surface:

```
DEFINE APROG immob
       FLOW       10
       DILUTE     r1a1 r1a2 r1a3 50 !Mix EDC/NHS
    *  INJECT     r1a3 70           !Inject EDC/NHS
 -0:30 RPOINT     Baseline -b
       DIPNEEDLE  r1b1              !Ligand buffer
       INJECT     r1b2 70           !Ligand
    *  INJECT     r1c1 70           !Deactivation
       EXTRACLEAN
  8:00 RPOINT     immob             !Immobilized
level
END

DEFINE APROG injection
       FLOW 20
       INJECT r2a1 20
END

MAIN
    DETECTION   1          !Sensorgram in flow cell 1
    APROG      immob
    APROG      injection
END
```

Using variables

To make the APROG more flexible, you can use variables instead of constant values for e.g. sample positions and volumes. The variables are replaced by values when the APROG is called from a MAIN block (see below).

To use variables in the DEFINE APROG block, you replace the constant parameter values with variable names. At the same time, you must add one or more PARAM commands immediately after the DEFINE APROG line, to tell the software which variable names you will use in the analysis program. For example:

```
DEFINE APROG injection
       PARAM %position %volume
       FLOW 20
       INJECT %position %volume
END
```

Note that a constant is used for the flow rate in this example, while variables are used for sample position and volume.

To use an analysis program with variables, add the values that you want to give the variables to the APROG command line, after the name of the program. For instance, using the following method will inject first 20 µl from position r1a1, then 35 µl from position r2a5:

```
DEFINE APROG injection
       PARAM %position %volume
       FLOW 20
       INJECT %position %volume
END

MAIN
    DETECTION 1
    APROG injection r1a1 20
    APROG injection r2a5 35
END
```

The order in which you give the parameter values in the APROG must be the same as the order in which they are specified in the PARAM command(s) within DEFINE APROG. The first value is assigned to the first variable name, the second value to the second variable name and so on.

You can use the same variable names in several different APROGs within the same method. The names are *local* to the APROG. You cannot on the other hand use the same name for two different variables within the same APROG.

Setting the sensorgram caption

The CAPTION command allows you to set a caption text for your sensorgram. You can use a variable name as part of the caption (or as the whole caption if appropriate). The example below sets the text “Injection of ... µl” as the sensorgram caption, with the respective sample volume filled in for each sensorgram:

```
DEFINE APROG injection
       PARAM %position %volume
       CAPTION Injection of %volume µl
       FLOW 20
       INJECT %position %volume
END

MAIN
    DETECTION 1
    APROG injection r1a1 20
    APROG injection r2a5 35
END
```

Timed events in analysis programs

So far, the APROG examples have executed commands in a straightforward sequence. One command starts as soon as the previous command is completed.

You can specify the time when a given command will start, relative either to the beginning of the analysis program or to some defined point within the program. You do this by putting a time specification before the command:

```
DEFINE APROG injection
       FLOW 20           ! time 0
  0:50 INJECT r1a1 20    ! start at 50 s
  6:00 INJECT r1a2 30    ! start at 6 min
END
```

In this APROG, the two injections will be performed 50 seconds and 6 minutes respectively after the start of the APROG.

A command with a time specification is executed at the time specified ***or as soon as possible thereafter***. In the APROG above, for example, the first injection occupies the autosampler for about 4 minutes. If the second injection were timed to start at 3 minutes (impossible since the autosampler is occupied), the second injection would be performed as soon as the first was completed. Time specifications in APROGs are thus *minimum* times, and do not necessarily represent the time of execution of the command. The actual time when the command is executed is recorded in the event log.

To set an injection, transfer or mixing time as a new zero time point within a program, put an asterisk (*) before the time specification. ***Do not type a space between the asterisk and the time specification.***

```
DEFINE APROG injection
       FLOW 20           ! time 0
 *0:30 INJECT r1a1 20   ! start at 30 s
  5:00 INJECT r1a2 30   ! start 5 min after
                        ! first inject
END
```

Here the second injection is timed to start 5 minutes after the beginning of the first injection.

You cannot use variables for time specifications.

Creating report tables

MDL gives you full freedom in creating report point tables (lists of SPR signal levels at specified times – see Section 9.2.4) from sensorgrams. You specify the contents of your report table through commands in the analysis program.

Report points are set with the RPOINT command, which should always be preceded by a time specification (otherwise the software does not know when to set the point). The command keyword is followed by an identification, which appears in the “Id” column in the report point table. A commonly used option with RPOINT is -b, which sets a new baseline at the report point. A simple example follows:

```
DEFINE APROG report
       FLOW 20
     * INJECT r1a1 10       ! set this as time 0
 -0:15 RPOINT baseline -b   ! set baseline
  0:30 RPOINT plateau       ! record plateau level
END
```

Note the negative time specification for the first RPOINT, setting the report point to 15 s *before* the first injection. This is often a convenient way to set report points relative to an event.

A report point table will always contain a minimum standard set of columns. You can specify additional columns with the KEYWORD command, which requires a header for the column and a value to be placed in the column. The APROG below will generate a report point table with an additional column headed “Antibody” containing the value “MAb-451”. In addition, the keyword “Antibody” will be included as a search key in the result file.

```
DEFINE APROG report
       KEYWORD Antibody MAb-451
       FLOW 20
     * INJECT r1a1 10       ! set this as time 0
   -15 RPOINT baseline -b   ! set baseline
    30 RPOINT plateau       ! record plateau level
END
```

The RPOINT and KEYWORD commands can use variables to help in documentation of the report point table.

E.3.3 Repetitive analyses
The DEFINE LOOP block

Basic loop constructions

The LOOP command and corresponding DEFINE LOOP block allow you to construct methods for series of repetitive analyses.

The following method analyses four different samples in duplicate – a total of 8 runs. We assume that ligand has previously been coupled to the sensor chip surface:

```
DEFINE LOOP cycle
       LPARAM  %sample  %position  %volume
       TIMES 2
               Sample1  r1a1       10
               Sample2  r1a2       15
               Sample3  r1a3       10
               Sample4  r1a4       20
END

DEFINE APROG analyse
       PARAM %sample %position %volume
       KEYWORD sample %sample
       FLOW 20
     * INJECT %position %volume
 -0:15 RPOINT baseline -b
  2:00 RPOINT plateau
  5:00 INJECT r2a1 10          ! regenerate
END

MAIN
    DETECTION 1
    LOOP cycle STEP
       APROG analyse %sample %position %volume
    ENDLOOP
END
```

The essential features of this construction are as follows:

- Variables are defined with the LPARAM command in the DEFINE LOOP block. This block establishes sets of variable values (4 sets of 3 variables each in this example). The TIMES 2 command in DEFINE LOOP specifies that the loop will go through the variable sets twice.

- The same variable names are set up as parameters for APROG analyse using the PARAM command in DEFINE APROG.

- In the MAIN block, the LOOP cycle...ENDLOOP construction executes the commands between LOOP and ENDLOOP once for every variable set in DEFINE LOOP cycle. In this example, the commands will be repeated 8 times (4 variable sets with a TIMES 2 repeat). Variables defined in the DEFINE LOOP block can be used as parameters for any command in the LOOP...ENDLOOP construction (in this case they are passed to APROG analyse).

The LOOP command in the MAIN block specifies one of the keywords STEP, ORDER or RANDOM. These determine the order in which the rows in DEFINE LOOP are processed:

STEP goes through the first row once, then the second row once and so on to the end, then repeats the cycle (samples 1, 2, 3, 4, 1, 2, 3, 4).

ORDER goes through all repeats of the first row, all repeats of the second row, and so on to the end (samples 1, 1, 2, 2, 3, 3, 4, 4).

RANDOM processes the rows in a random order. This is particularly useful for avoiding systematic effects in extensive series.

You can vary the number of replicate analyses within a DEFINE LOOP simply by adding more TIMES commands. The TIMES number applies to all subsequent rows in the DEFINE LOOP until a new TIMES is set:

```
DEFINE LOOP cycle
       LPARAM  %sample %pos  %vol
       TIMES 2       ! do the following rows twice
               Sample1 r1a1  10
               Sample2 r1a2  15
               Sample3 r1a3  10
       TIMES 1       ! do the last row only once
               Sample4 r1a4  20
END
```

Repeating multiple commands

In the examples above, the LOOP command activates just one analysis program. You can however include any command which is permitted in MAIN within a loop. The whole command sequence between LOOP and ENDLOOP will be repeated for the rows in DEFINE LOOP. The example below uses a loop construction to run a sample through different flow cells in four separate injections with a different injection volume in each flow cell:

```
DEFINE APROG analyse
       PARAM %volume
       FLOW 20
       INJECT r1a1 %volume
       ...
END

DEFINE LOOP flowcells
       LPARAM  %cell  %volume
       TIMES 1
               1      10
               2      20
               3      30
               4      40
END

MAIN
    LOOP flowcells STEP
         DETECTION %cell
         APROG analyse %volume
    ENDLOOP
END
```

Advanced LOOP constructions

Loop constructions can be *nested*, that is a loop can be called from within another loop. This technique is particularly useful for applications that work with a matrix of samples, testing all pair-wise combinations (e.g. epitope mapping). The example below runs a 3×3 matrix, with the samples A, B and C in positions r1a1-r1a3.

```
DEFINE LOOP cycle1
       LPARAM  %s1   %p1
               A     r1a1
               B     r1a2
               C     r1a3
END

DEFINE LOOP cycle2
       LPARAM  %s2   %p2
               A     r1a1
               B     r1a2
               C     r1a3
END

MAIN
    DETECTION 1
    FLOW 20
    LOOP cycle1 STEP
       LOOP cycle2 STEP
          APROG analyse %s1 %s2 %p1 %p2
          APROG regenerate
       ENDLOOP
    ENDLOOP
END
```

This works as follows:

1. At `LOOP cycle1 STEP`, values for `%s1` and `%p1` are taken from the first row in `DEFINE LOOP cycle1` (“A” and “r1a1” respectively).

2. The method then runs through all three rows in `DEFINE LOOP cycle2` using the values “A” and “r1a1” for `%s1` and `%p1`. This runs the combinations A-A, A-B and A-C.

3. When `LOOP cycle2` is completed, the next step of `LOOP cycle1` is executed, taking “B” and “r1a2” for `%s1` and `%p1`. `LOOP cycle2` now runs again, generating the combinations B-A, B-B and B-C.

4. In the same way, the last step of `LOOP cycle1` generates the combinations C-A, C-B and C-C.

This example is only a 3×3 matrix, but the economy of nested loop constructions is readily apparent if you consider for example a 10×10 matrix with 100 combinations.

E.3.4 Communicating with the user The DEFINE ALERT and DEFINE MESSAGE blocks

The two commands ALERT and MESSAGE, with their corresponding DEFINE ALERT and DEFINE MESSAGE blocks, are used to inform the user of the progress of a method. ALERT activates a dialogue window which allows the user to abort or continue the method. MESSAGE simply displays a text window. The command BEEP sounds a beep to call the operator's attention, and can often be appropriate in connection with ALERT or MESSAGE.

The instrument door is unlocked whenever an ALERT window is displayed, giving the user access to the autosampler compartment for e.g. inserting samples or changing racks. The door is re-locked when the user acknowledges the window and continues with the method.

For both DEFINE ALERT and DEFINE MESSAGE, text messages are written in lines of up to 40 characters. The commands LTEXT, CTEXT and RTEXT place text messages either left-justified, centred or right-justified respectively on the line. The text may include variable names, which are declared in one or more PARAM commands and which are given values when the ALERT or MESSAGE is called from the MAIN block, in exactly the same way as parameters are passed to APROG (see Section E.4.2). Additional functions specific to the two types of information block are:

- DEFINE ALERT block should have a GO command, which generates a dialogue box button for continuing the method. The STOP command generates a similar button for aborting the method. Each command requires a parameter which is the text to be shown in the button.

- DEFINE MESSAGE may include a CLOCK command, which displays a count-down timer in the upper right-hand corner of the MESSAGE window. The clock is for information only, and does not affect method operation in any way. There is a LIFETIME command which may be used during method development to find out how long a process takes to complete, to give a suitable starting value for CLOCK.

An ALERT window remains displayed until the user presses the GO or STOP button. A MESSAGE window remains displayed until one of the following conditions is met:

- another MESSAGE window or an ALERT window is displayed

- the window is removed by issuing the command MESSAGE with no parameters

- the method ends.

The following example illustrates the use of ALERT in a method where an unstable sample is to be thawed immediately before use:

```
DEFINE ALERT insert_sample
       TITLE Unstable sample
       BEEP
       LTEXT
       LTEXT Continuation of this method requires
       LTEXT sample in position r2a1. Make sure that
       LTEXT the sample is completely thawed before
       LTEXT placing it in the autosampler rack.
       LTEXT
       LTEXT Close the door and click on Continue
       LTEXT when you are ready.
       GO    Continue
       STOP  Cancel
END

MAIN
    RACK 1 thermo_A
    RACK 2 thermo_B
    ALERT insert_sample
    APROG cycle1
    ...
END
```

E.3.5 Additional data curves The CURVE command

A new cycle is automatically created for every APROG in a method. With the CURVE command, you can record status information as graphs in parallel with the sensorgram. Place this command at the beginning of the APROG: it may only be preceded by PARAM, MODE, CAPTION, SCALE or KEYWORD.

The status items that can be recorded are buffer flow and remaining volume in the continuous flow pump and temperature in the detection unit (see Appendix F). Use buffer flow and temperature to check for stability throughout a method. Monitor buffer volume during method development to check when the pumps are refilled, to ensure that refilling does not occur at a critical point in the sensorgram.

The command CURVE creates a status curve related either to the left (RU) axis as default or to an optional axis on the right of the sensorgram. The following example plots temperature and buffer flow as overlays on the sensorgram, with the temperature axis on the right:

```
DEFINE APROG monitor
       CURVE TEMP -o     ! use right-hand axis
       CURVE bufflow     ! use default axis
       FLOW 20
       ...
       ...
END
```

You can set report points on status curves using the `-c` option for the `RPOINT` command.

E.3.6 Closing down
The APPEND command

The END command at the end of every `APROG` includes automatic functions for washing the IFC and flow cells with buffer. You can append the working tool **Standby** to a method if the instrument is not going to be used for up to 4 days after completion of the method. This will flush the IFC and maintain a low buffer flow for up to 4 days (see Section 7.2.1). For example:

```
DEFINE APROG analyse
       ...
END

DEFINE APROG regenerate
       ...
END

MAIN
    DETECTION 1
    APROG analyse
    APROG regenerate
    APPEND standby
END
```

Note: For compatibility with earlier versions of MDL, the command `APPEND continue` is equivalent to `APPEND standby`.

E.4 Method techniques

This section considers some features of method programming from the viewpoint of application requirements.

E.4.1 Detection mode and flow path

Detection mode
BIACORE 2000 operates in single or multichannel detection mode, with data collection from one, two or four flow cells.

You set the detection mode and the initial flow path with a DETECTION command in the MAIN block of the method. You can change the detection mode between APROGs in the MAIN block.

```
MAIN
    DETECTION 1            ! single channel mode
    APROG single
    DETECTION 1,2,3,4      ! multichannel mode
    APROG multi
END
```

If there is no DETECTION command before the first APROG in the MAIN block, the software will prompt for detection mode when the method is run:

The selection in this dialogue box corresponds directly to the DETECTION command in the MAIN block. The sensorgram will not start until you click on **OK** in this dialogue box.

Methods written without a DETECTION command are flexible in that you can use the same method for analysis in different flow cells.

The DETECTION command may only be used in the MAIN block.

Flow path

The flow path set by the DETECTION command remains open until one of the following occurs:

- a new DETECTION command is issued in the MAIN block
- a FLOWPATH command is issued in an APROG (see below)
- an INJECT command is issued in an APROG with a flow cell requirement different from the current flow path.

You can use the FLOWPATH command in an APROG to change the flowpath during the cycle. You may for instance wish to bind analyte to the ligand in all four flow cells and then test the binding of four different secondary ligands, one in each flow cell:

```
MAIN
    DETECTION 1,2,3,4
    APROG epitope
END

DEFINE APROG epitope
       PARAM %pos1 %pos2 %pos3 %pos5 %pos5
       FLOW 5
       FLOWPATH 1,2,3,4
       INJECT %pos1 5         ! first Mab
       INJECT r1c1 5          ! analyte
       ...
       ...
       FLOWPATH 1
       INJECT %pos2 5         ! second Mab 1
       ...
       FLOWPATH 2
       INJECT %pos3 5         ! second Mab 2
       ...
       FLOWPATH 3
       INJECT %pos4 5         ! second Mab 3
       ...
       FLOWPATH 4
       INJECT %pos5 5         ! second Mab 4
       ...
       FLOWPATH 1,2,3,4
       INJECT r1d3 15         ! regenerate
       ...
END
```

The FLOWPATH command may only be used in an APROG block.

IMPORTANT
The DETECTION command sets both the detection mode and the initial flow path. However, the FLOWPATH command does not affect the detection mode. Take care not to set the flow path to include cells where data is not being collected, e.g.

```
MAIN
       DETECTION 4     ! single channel detection in Fc
4
       APROG missing_data
END

DEFINE APROG missing_data
       FLOW 20
       FLOWPATH 1,2,3  ! sample flow through Fc 1-2-3
                       ! detection only in Fc 4
       ...
END
```

The same mistake can easily be made with a variable contact time injection, which sets the flow path implicitly (see Section E.4.3).

The design of the **Detection** and **Inject** dialogue boxes prevents this mistake under manual control, but there is no equivalent safeguard in methods. To avoid missing data in method runs:

- Use FLOWPATH with care. Few applications require changes in flow path during an APROG. It is often preferable to use separate APROGs, setting the flow path for each one with a DETECTION command in MAIN.

- Set DETECTION 1,2,3,4 at the beginning of the MAIN block if you are unsure of the requirements of your APROGs. This will set multi-channel detection mode, and no data will be lost.

On-line reference subtraction
To use on-line reference subtraction, use one of the following parameters with the DETECTION command:

Parameter	Detection mode	Flow path	Reference subtraction
2-1	1,2	1,2	2-1
2-1,4-3	1,2,3,4	1,2,3,4	2-1, 4-3
2-1,3-1,4-1	1,2,3,4	1,2,3,4	2-1,3-1,4-1

An example is given below

```
MAIN
    DETECTION 2-1,4-3      ! single channel mode
    APROG subtract_reference
END
```

E.4.2 Sample injection types

BIACORE 2000 offers a range of sample injection types for different application purposes. These are available from the manual control interface (see Section 4.4.6), and all except manual injection can be programmed in methods.

To make sure that an injection will not be disturbed by a pump refill operation, issue a `FLOW` command with the `-f` option just before the `INJECT` command, e.g.

```
DEFINE APROG smooth_inject
       FLOW 35
 10:00 WASH n
       FLOW 35 -f    ! refill pump
       KINJECT r1a1 140 240
END
```

Any `INJECT` command issued when the flow rate is zero causes a run-time error. This situation is not detected by method pre-run.

Command	***Comments***	***Injected volume***	***Sample consumption***	***Max flow rate***
INJECT	General purpose inject.	5-325 µl	Injected vol +30 µl	100 µl/min
KINJECT	Low dispersion inject for kinetics and for samples sensitive to dilution with running buffer. Controlled dissociation time after sample injection.	10-250 µl	Injected vol +40 µl	100 µl/min
QUICKINJECT	Low sample consumption. Less preparation time than INJECT but more sample dispersion.	5-325 µl	Injected vol +10 µl	100 µl/min
COINJECT	Injection of second sample immediately following the first (e.g. for dissociation in the presence of ligand). Variable contact times for first sample only. Recovery of either first or second sample, but not both.	Sample 1: 10-100 µl Sample 2: 15-250 µl	Sample 1: injected vol +40 µl Sample 2: injected vol +40 µl	100 µl/min
BIGINJECT	Large volume inject. Direct inject up to 325 µl, sample loop between 326 and 425 µl, direct between 426 and 750 µl.	326-750 µl	Injected vol +52 µl	50 µl/min*

*BIGINJECT cannot be performed at higher flow rates because the auto-sampler tubing must be refilled during injection from the sample loop for samples larger than 425 µl.

E.4.3 Variable contact time injections

The sample volume parameter for `INJECT` commands can be either a single volume or a list of 2, 3 or 4 volumes separated by commas.

Single volume	In single channel mode, this is the sample volume injected through the currently selected flow cell. In multichannel mode, this volume will be injected over all flow cells in the currently selected flow path.
Volume list	A list of volumes specifies variable contact times in multichannel mode. The first, second, third and fourth volumes apply to flow cells 1, 2, 3 and 4 respectively.

Each volume in the volume list must be less than or the same as the preceding volume. Thus `INJECT 35,25,20,10` and `INJECT 35,25,25,10` are valid, but `INJECT 35,20,`***`25`***`,10` is not.

An INJECT command with a volume list sets the flow path as required for the duration of the injection. When the injection is complete, the flow path is restored to the previous setting:

```
MAIN
    DETECTION 1,2,3,4   ! All four flow cells in flow
path
    APROG immob
END

DEFINE APROG immob
       FLOW 5
       ...
       INJECT r1a1 35,25 ! Flow cells 1 and 2 for INJECT
      ...                ! Back to all four flow cells
END
```

Remember to use DETECTION 1,2,3,4 in the MAIN block to set multi-channel detection if you use variable contact times in INJECT commands. If you set single channel detection mode, you will inject sample over multiple flow cells with variable contact times, but you will record the sensorgram from one cell only (see note above).

Note: An INJECT command with a single volume parameter does not affect the flow path.

E.4.4 Sample recovery and fraction collection

To program sample recovery or fraction collection, place the RECOVER or FRACTIONS command directly after the INJECT command to which it applies. You can however enter a comment or an EXTRACLEAN command between the INJECT and the RECOVER or FRACTIONS command.

E.4.5 The EXTRACLEAN command

Automatic routines wash the liquid handling system after each INJECT command and after each APROG. If more thorough cleaning is required, use the EXRACLEAN command (corresponding to **Extra Cleanup** in the **Inject** dialogue box in the manual control interface). See Appendix B for recommendations concerning when to use the extra cleanup routines.

The EXTRACLEAN command must be placed directly after the INJECT command to which it applies. You can however enter a comment or a FRACTIONS or RECOVER command between the INJECT and the EXTRACLEAN command.

E.4.6 User tools in methods

The working tools **Flush, Normalize, Prime** and **Rinse** and the service tool **Unclogging** can be used between cycles in a method. Enter the corresponding command in the `MAIN` block (see Appendix F for command details).

The user tool **Standby** may be added at the end of a method using the `APPEND` command in the `MAIN` block. This puts the instrument in standby mode when the method is completed.

E.5 Common errors and error handling

This section describes some of the most common mistakes you will make when writing methods, and how BIACORE Control Software handles incorrect methods.

E.5.1 General error handling

All methods are tested in the BIACORE 2000 Control Software prerun facility before they are run. You may choose to do this explicitly at any stage: alternatively, you may try to run the method directly, in which case the prerun is done automatically before the method is actually activated.

The prerun facility checks the syntax and consistency of your method. If there are any errors, error messages are written in the appropriate places in the actual method file. An error message is identified by the two characters “>>” at the beginning of the line. A message above the error identifies the nature of the error, and a line below the error has a caret (^) pointing to the incorrect word:

```
...
>>  Invalid vial position
    * INJECT r4a1 20
>>           ^

! r4 is not a valid rack number
...
```

If you correct errors in a method but leave the error messages in place, all lines beginning with “>>” are stripped out of the method file at the next pre-run. You do not need to remove the error message lines manually. You can also use **Edit:Adjust Method** to remove error messages from the method.

E.5.2 Syntax errors

Among the most common errors are syntax errors, breaking the rules of MDL. For instance, forgetting the END at the end of a DEFINE APROG is a syntax error.

You may make syntax errors through incorrect usage (for instance, trying to put a RACK command in a DEFINE APROG block) or through typing mistakes (for instance, writing APORG instead of APROG). Normally, BIACORE Control Software cannot distinguish between these situations – both give an "Unrecognized command":

```
DEFINE APROG test
       FLOW 20
>> Unrecognized command
       RACK 1 Thermo_a
>> ^

! RACK cannot be used in APROG
```

```
MAIN
>> Unrecognized command
    APORG test
>>  ^

! APORG should be APROG
```

Note that some errors may arise as a direct consequence of previous mistakes, even though the syntax in itself is correct. In the example below, the END command is missing from the DEFINE APROG, with the result that the syntax checker has difficulty interpreting the (correctly written) MAIN block:

```
DEFINE APROG test
       FLOW 20
       INJECT r1a1 10

! END missing

>> Expecting keyword END
MAIN
>>^
     DETECTION 2
>> Characters outside DEFINE-END or MAIN-END blocks
     APROG test
>>   ^
END
```

To make interpretation of consequential errors easier when you have several error messages in one method, correct the errors one at a time starting at the beginning of the method, and repeat **Prerun** after each correction.

E.5.3 Incorrect variable names

Misspelling a variable name causes an error in two contexts:

- Within a DEFINE block (defining ALERT, APROG or MESSAGE), the variable names declared in the PARAM command must be the same as those used in commands within the block.

- The variable names declared in the LPARAM command in a DEFINE LOOP must be the same as those used in commands within the LOOP construction.

Mistakes in these situations generate the following error messages:

```
DEFINE APROG test
       PARAM %flow %pos %vol
>> Expecting an integer value, min=0 max=100
       FLOW %folw
>>          ^

! variable name should be %flow

       INJECT %pos %vol
END

DEFINE LOOP cycle
       LPARAM %flow %pos  %vol
              5     r1a1  10
              10    r1a1  10
END

MAIN
    DETECTION
    LOOP cycle STEP
>> Expecting an integer value, min=5 max=325
       APROG test %flow %pos %volume
>>                           ^

! variable name should be %vol

    ENDLOOP
END
```

E.5.4 Incorrect parameter values

Many parameters have a restricted range of values. You cannot for instance select flow cell number 5 (there are only 4 flow cells) or move the autosampler needle to position r4a1 (the position does not exist). Permitted ranges are given in the detailed descriptions for each command (Appendix F). BIACORE Control Software warns you for incorrect parameter values:

```
...
>>  Invalid vial position
    * INJECT r4a1 20
>>           ^

! r4 is not a valid rack number
...
```

"Invalid vial position" is a special case, in that the different positions are valid for different racks. If the method does not contain a RACK command, the syntax will be checked in the context of the current rack setting.

E.5.5 Incorrect parameter ordering

Parameters (values or variable names) must be given in the right order when you call an APROG, ALERT or MESSAGE from the MAIN block. The order must be that given in the PARAM declaration.

If you make a mistake in the ordering, the chances are that you will generate a syntax error. The following incorrect method for instance tries to use the value 10 for autosampler position and r1a1 for volume:

```
DEFINE APROG test
       PARAM %vol %pos
       FLOW 20
       INJECT %pos %vol
END

MAIN
    DETECTION 1
>> Expecting a value value, min=5 max=325
    APROG test r1a1 10
>>             ^

! APROG test is defined with volume before position

END
```

In some cases, incorrect parameter ordering might go undetected by BIACORE Control Software. The following method is syntactically valid:

```
DEFINE APROG test
       PARAM %flow %pos %vol
       FLOW %flow
       INJECT %pos %vol
END

DEFINE LOOP cycle
       LPARAM %vol  %pos  %flow
              10     r1a1  5
              20     r1a1  10
END

MAIN
    DETECTION 1
    LOOP
       APROG test %vol %pos %flow
    ENDLOOP
END
```

The parameter order is reversed between the `APROG` call and the `PARAM` declaration, but the method passes perfectly valid parameter values to the `APROG`, with the result that 5 µl is injected at 10 µl/min instead of 10 µl at 5 µl/min. Errors of this kind can be detected by inspection of the event log for the run.

E.5.6 Incorrect number of variables

You must declare all variables used in a program. An error results otherwise:

```
DEFINE APROG test
       PARAM %pos %vol
>> Expecting an integer value, min=0 max=100
       FLOW %flow
>>          ^

! variable %flow not declared

      INJECT %pos %vol
END
```

You may declare more variables than you use – for instance, during development work you may “comment out” commands which use variables, but you do not need to remove the variable names from the declaration:

```
DEFINE APROG test
       PARAM %sample %mix %vol
       FLOW 20
 ! TRANSFER %sample %mix %vol
 ! MIX %mix 100
       INJECT %mix 10
END

! parameters %sample and %volume not used
```

In the same way, parameter values must be supplied for all variables used in a program. If you do not supply enough values, an error results:

```
DEFINE APROG test
       PARAM %flow %pos %vol
       FLOW %flow
       INJECT %pos %vol
END

MAIN
     DETECTION 1
>> Missing parameter
     APROG test 25 r1a1
>>                     ^

! APROG test requires 3 parameters.
! No value given for %vol

END
```

E.5.7 Prerun error messages

This section lists the error messages that can be inserted into the method file when prerun encounters a syntax error.

... commands must precede all other instrument command types

Variable declarations (PARAM in APROG, LPARAM in LOOP) must be the first commands in the block. KEYWORD, MODE, CURVE and FLOW must precede other commands in APROG.

... has already been used

You have repeated one of RECOVER, FRACTIONS or EXTRACLEAN after an injection command.

... must be the first command in ...
RACK must be the first command in MAIN, with the exception of ALERT which can precede RACK.

... not allowed after APPEND
No commands may be written after APPEND in the MAIN block.

... not allowed with ...
You have used a combination of commands which is not permitted:
RECOVER cannot be used with FRACTIONS
RECOVER or FRACTIONS cannot be used with BIGINJECT
FRACTIONS cannot be used with COINJECT.

Buffer overflow
This error can arise if a line in the method definition exceeds 255 characters.

Characters outside DEFINE-END or MAIN-END blocks
All characters in a method file must be within method definition blocks. A common source of this error is omission of DEFINE at the beginning of a block. Consequential error messages can easily arise as a result.

Conflicting LPARAM ...
Nested loops use parameter names in common. The error message points to the inner LOOP command.

ENDLOOP without matching LOOP
The LOOP command must be balanced with an ENDLOOP after the commands to be repeated by the loop. Omitting the LOOP gives this error message.

Expecting...
Depending on the context of the error, the syntax checker may be able to suggest an expected keyword, value or character.

Expecting a real number, min=... max=...
The command requires a real number (i.e. integer or decimal value) as a parameter. This message can arise if a variable name is incorrect, so that the parameter value supplied to the command is not a real number in the correct range.

Expecting a time value, min=... max=...
A time parameter has been given a value which is outside the range permitted in the context (for example a time which is longer than 3600 s).

Expecting an integer value, min=... max=...
The command requires an integer value as a parameter. This message can arise if a variable name is incorrect, so that the

parameter value supplied to the command is not an integer in the correct range.

Fractional time is not allowed
This message appears when a decimal time is specified where only integer time specifications are allowed (e.g. `WAIT`).

Incompatible parameters
You have used a variable in different contexts which are not compatible with each other. This message also appears if you use the same variable for the volume with `BIGINJECT` and another injection command.

Invalid block name
You have used a name which is not valid as the name of a method definition block. Names may contain letters, digits and the special characters & _ and %, but may not begin with a digit or underscore character.

Invalid rack type
The parameter you have provided for the `RACK` command is not a valid rack type. Valid types are `THERMO_A`, `THERMO_B`, `THERMO_C` and `MICRO`.

Invalid time
You have used an invalid time specification. Give times in seconds (ssss), minutes and seconds (mmm:ss) or hours, minutes and seconds (hh:mm:ss). The specification 1:70 (1 minute and 70 seconds) is invalid.

Invalid variable name
You have used a name which is not valid as the name of a variable. Names may contain letters, digits and the special characters & _ and %, but may not begin with a digit or underscore character.

Invalid vial position
You have specified a rack position that is not valid for the current rack type. Positions are “valid” in the context of the current rack definition. Check the position specification and the rack type currently defined in **Command:Define Rack Base**.

Method ... not found
You have used `APPEND` with an incorrect method name. Only the internal method `STANDBY` may be called with `APPEND`.

Missing "
Quotation marks around text (e.g. in `LTEXT`, `CTEXT`, `RTEXT`, `CAPTION` commands) must be matched.

Missing parameter

You have called a block with fewer parameters than are declared in the corresponding block definition.

Missing reference time

You have specified a negative time for an RPOINT command in APROG with no preceding *-marked command to set the reference time.

Name duplicated

You have used the same name twice in the same context. Two blocks of the same type may not have the same name, and variables declared with PARAM must be unique within one block. Use the **Edit:Find** command in the method editor to find different occurrences of the same name.

Only one decimal allowed

This message appears when more than one decimal is specified for a number that only supports one decimal.

Only one MAIN allowed

A method may not contain more than one MAIN block definition.

Option without value

You have used a command option without specifying a value where one is required, for example CURVE temp -p.

String too long

You have used a string in for example a TITLE, CTEXT, LTEXT or RTEXT command that is longer than the maximum permitted length.

Syntax error

There is an error in the method definition that the syntax checker cannot identify.

This command cannot be used as a reference point

You can only set a reference point for timed events in an APROG, i.e. for the commands BIGINJECT, COINJECT, DILUTE, INJECT, KINJECT, QUICKINJECT, MIX and TRANSFER.

This command must follow immediately after ...

You have used a command in the wrong command order. The command in question must follow directly after an inject command. This error may occur with the commands RECOVER, FRACTIONS and EXTRACLEAN.

Time may not exceed ... s

The maximum time allowed for 1 cycle is 1.5×10^6 s (about 17 days). You have probably made a mistake in a time specification.

Time not supported by command

You can only set time specifications in an APROG for the commands FLOW, BIGINJECT, COINJECT, INJECT, KINJECT, QUICKINJECT, RPOINT, MIX, TRANSFER and WASH.

Note: The first FLOW command in an APROG cannot be timed.

Timing order out of sequence

Timed events in an APROG must be written in chronological sequence.

Too many parameters

A command has been written with more parameters than the syntax defines, or a method block has been called with more parameters than are declared in the corresponding DEFINE block.

Too many variables

The number of variables declared with PARAM commands in one APROG may not exceed 20.

Undefined name

You have called a method block using a name that is not defined. Check that the block is properly defined and that the name is correctly spelt.

Unrecognized command

You have used a command in a method definition block where it is not allowed, or you have spelt a command keyword incorrectly.

Unrecognized option

You have specified an option which is not valid for the command.

Variable not allowed

You have used a variable in a context where it is not allowed. The first parameter supplied to the command KEYWORD may not be a variable.

F. Method language reference

This chapter is a complete reference to the method editor commands.

- Keywords are written in upper-case.
- Parameter identifiers are written in lower-case italics.
- Optional components are enclosed in square brackets ([...]).
- An ellipsis (...) is used to indicate that a list of components may be extended indefinitely.
- Options are used only for special purposes, such as maintenance and test methods. All commands have default option settings which are suitable for normal applications.
- "Default values" are values that will be used for parameters and options if no other value is specified. The default values apply even if the option is omitted from the command.

Command summary

The tables below list the commands which can be used in each method block together with a brief indication of the command purpose. The remainder of this chapter describes the commands in detail, listed in alphabetical order.

DEFINE ALERT block

`BEEP`	Sounds a beep to alert the operator
`CTEXT`	Writes text centered horizontally in box
`GO`	Displays a button for "continue" function
`LTEXT`	Writes text left-aligned in box
`PARAM`	Declares variables used in `ALERT`
`RTEXT`	Writes text right-aligned in box
`STOP`	Displays a button for "abort" function
`TITLE`	Sets the title bar text for the `ALERT` box

DEFINE APROG block

`BIGINJECT`	Large volume injection 326 - 750 µl to IFC
`CAPTION`	Sets caption for sensorgram or curve
`COINJECT`	Dual injection to IFC
`CURVE`	Plots values for a status item
`DILUTE`	Dilutes one sample with another
`DIPNEEDLE`	Predips needle in vial
`EXTRACLEAN`	Extra wash procedure after injection
`FLOW`	Sets flow rate
`FLOWPATH`	Defines flow path
`FRACTIONS`	Recovers dissociating sample during buffer flow
`INJECT`	Injects sample to IFC
`KEYWORD`	Defines column in report point table
`KINJECT`	Low dispersion injection
`MIX`	Mixes sample
`MODE`	Sets data collection parameters
`PARAM`	Declares variables used in `APROG`
`QUICKINJECT`	Quick injection to IFC
`RECOVER`	Recovers injected sample or reagent
`RPOINT`	Sets report point
`SCALE`	Sets default scale for sensorgram and status curves
`TRANSFER`	Transfers sample between autosampler vials
`WAIT`	Waits a specified time
`WASH`	Washes IFC channels and/or autosampler needle
`WASHREC`	Washes recovery cup

DEFINE LOOP block

`LPARAM`	Declares variables used in `DEFINE LOOP`
`TIMES`	Sets the repeat count for sets of variables

DEFINE MESSAGE block

`BEEP`	Sounds a beep to alert the operator
`CLOCK`	Displays count-down timer
`CTEXT`	Writes text centered horizontally in box
`LIFETIME`	Shows time for which message has been displayed
`LTEXT`	Writes text left-aligned in box
`PARAM`	Declares variables used in `MESSAGE`
`RTEXT`	Writes text right-aligned in box
`TITLE`	Sets the title bar text for the `MESSAGE` box

MAIN block

`ALERT`	Displays an `ALERT` box
`APPEND`	Runs a termination routine after the method
`APROG`	Runs an analysis program (`APROG`)
`BEEP`	Sounds a beep to alert the operator
`DETECTION`	Selects detection mode and initial flow path
`DILUTE`	Dilutes one sample with another
`ENDLOOP`	Ends a loop definition
`FLOW`	Sets flow rate
`FLUSH`	Runs the working tool **Flush**
`LOOP`	Repeats one or more commands
`MESSAGE`	Displays a `MESSAGE` box
`MIX`	Mixes sample
`NORMALIZE`	Runs the working tool **Normalize**
`PRIME`	Runs the working tool **Prime**
`RACK`	Sets the rack base type
`RINSE`	Runs the working tool **Rinse**
`TEMP`	Sets the detection temperature
`TRANSFER`	Transfers sample between autosampler vials
`UNCLOG`	Runs the working tool **Unclogging**
`WAIT`	Waits a specified time
`WASH`	Washes liquid handling system components
`WASHREC`	Washes recovery cup

ALERT *name* [*parameter ...*]

Displays an ALERT box.

Usage: MAIN block.

Parameters: *name* Name of the ALERT box as specified in DEFINE ALERT.

parameter Parameter supplied to the ALERT box.

Parameters supplied to the ALERT must correspond sequentially to the variables declared in the PARAM command in DEFINE ALERT.

Within a LOOP construction, parameters may be supplied as either constant values or variable names. Variable names must be the same as those declared in the corresponding DEFINE LOOP. Outside a LOOP, parameters must be supplied as constant values.

Example:

```
DEFINE ALERT check_solution
       PARAM %pos
       LTEXT Place test solution in position %pos
       GO OK
END

MAIN
    ALERT check_solution r1a1
END
```

APPEND *standby*

Runs the working tool **Standby** after the method.

Usage: `MAIN` block.

Parameters: *standby* Name of the tool to run after the method. At present, only the **Standby** tool is accepted.

The `APPEND` command must be the last command in the `MAIN` block.

Note: The working tool **Continue** provided in previous versions of BIACORE control software has been renamed to **Standby**. For backward compatibility, the name **Continue** may still be used with the `APPEND` command.

Example:

```
MAIN
    DETECTION 1
    APROG immob
    APROG analyse ...
    APROG regenerate
    APPEND standby
END
```

APROG name [parameter ...]

Runs an analysis program (APROG).

Usage: MAIN block.

Parameters: *name* Name of the program as specified in DEFINE APROG.

parameter Parameter supplied to the APROG.

Parameters supplied to the APROG must correspond sequentially to the variables declared in the PARAM command in DEFINE APROG.

Within a LOOP construction, parameters may be supplied as either constant values or variable names. Variable names must be the same as those declared in the corresponding DEFINE LOOP. Outside a LOOP, parameters must be supplied as constant values.

Example:

```
DEFINE APROG injection
       PARAM %position %volume
       FLOW 20
       INJECT %position %volume
END

MAIN
    DETECTION 1
    APROG injection r1a1 20
    APROG injection r2a5 35
END
```

[*] [time] BIGINJECT *position volume*

Large volume injections (326 - 750 µl).

Usage: `DEFINE APROG` block.

Parameters: *time* Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.

position Rack position of sample.

volume Volume to be injected, or a list of up to four volumes separated by commas for variable contact time injections (see Section E.4.3). Range 326-750 µl.

`BIGINJECT` first loads the IFC sample loop with 100 µl sample. The first 325 µl of the injection are injected directly, followed by up to 100 µl from the loop, then followed by direct injection to complete the volume if necessary. Slight disturbances in the sensorgram may be apparent as the injection switches between direct and loop injection modes.

The maximum flow rate during `BIGINJECT` is 50 µl/min. This limitation is necessary so that the autosampler has time to re-fill the tubing for the second direct injection during the loop injection.

`RECOVER` and `FRACTIONS` cannot be used with `BIGINJECT`.

Sample consumption is injected volume + 52 µl.

Example:

```
DEFINE APROG large_vol
       FLOW 50
 *0:30 BIGINJECT r1a1 500
  4:00 RPOINT response   !rpoint 4 min
                         !after injection start
END
```

BEEP

Sounds a beep from the computer to alert the operator.

Usage: `DEFINE ALERT`, `DEFINE MESSAGE` and `MAIN` blocks.

Parameters: None.

Example:

```
MAIN
    APROG immob
    BEEP
    APROG analyse
END

DEFINE MESSAGE box
       BEEP
       BEEP
       LTEXT Don't miss this message!
END
```

CAPTION *text*

Sets the caption of the current graph.

Usage: `DEFINE APROG` block.

Parameters: *text* Text to be used as caption.

Example:

```
DEFINE APROG temp_check
       CAPTION Temperature check
       CURVE temp
       FLOW 10
       INJECT r1a1 10
END
```

CLOCK time [option]

Displays count-down timer.

Usage: `DEFINE MESSAGE` block.

Parameters: *time* Starting time for timer. Format HH:MM:SS, MMM:SS or SSS.

Option: *-m* Display time in minutes only.

Example:

```
DEFINE MESSAGE countdown
       RTEXT minutes remaining
       CLOCK 5:00 -m
END
```

[*] [time] COINJECT *position1 volume1 position2 volume2*

Low dispersion inject with two direct following injections, e.g. for competing ligand during dissociation phase.

Usage: `DEFINE APROG` block.

Parameters: *time* Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.

position1 Rack position for sample 1.

volume1 Volume of sample 1, or a list of up to four volumes separated by commas for variable contact time injections (see Section E.4.3). Range 10-100 µl.

position2 Rack position for sample 2.

volume2 Volume of sample 2, range 15-250 µl.

Sample 1 is injected from the IFC sample loop. Sample 2 is injected in direct inject mode. Variable contact times can only be specified for sample 1.

`FRACTIONS` cannot be used with `COINJECT`. `RECOVER` can be used for either sample 1 or sample 2 but not both.

Sample consumption is injected volume + 40 µl for both sample 1 and sample 2.

Example:

```
DEFINE APROG double_inject
       FLOW 50
 *0:30 COINJECT r1a1 50 r1a2 150
  0:50 RPOINT resp1   !rpoint 1st inject
  3:50 RPOINT resp2   !rpoint 2nd inject
END
```

CTEXT [*text*]

Writes centered text in an `ALERT` or `MESSAGE` box.

Usage: `DEFINE ALERT` and `DEFINE MESSAGE` blocks.

Parameters: *text* — Text containing up to 40 characters. A blank line is inserted in the `MESSAGE` box if *text* is omitted.

Example:

```
DEFINE MESSAGE demo
       TITLE DEMONSTRATION BOX
       LTEXT This text is left-adjusted
       CTEXT This text is centered
       RTEXT This text is right-adjusted
END
```

CURVE *item* [*option ...*]

Plots the value of a status item in the sensorgram window.

Usage: `DEFINE APROG` block.

Parameters: *item* Name of the status item to plot:

temp	detection unit temperature
bufflow	buffer flow rate (continuous flow pump)
buffvol	volume of buffer remaining in the continuous flow pump

Options

-o Optional plot. Relate values to the right-hand axis.

-p*n* Pen colour identified by *n*:

0	black	8	grey
1	blue	9	light blue
2	green	10	light green
3	cyan	11	light cyan
4	red	12	light red
5	magenta	13	light magenta
6	yellow	14	light yellow
7	white	15	bright white

-d*n* Reading rate interval. Values for n in seconds. Range 0.2-65 s. Default values are 30 for *temp* and 1 for *bufflow* and *buffvol*.

Example:

```
DEFINE APROG temp_check
       CURVE temp -o
       FLOW 10
       INJECT r1a1 10
END
```

DEFINE ALERT

Defines a dialogue box prompting with “continue” and “abort” buttons.

Usage: Block definition.

Syntax:

```
DEFINE ALERT name
     [PARAM variable ...]
     other commands
  END
```

The GO and STOP commands respectively are used in the DEFINE ALERT block to specify text on the “continue” and “abort” buttons in the ALERT box.

If the block contains neither STOP nor GO commands, the box will include a “continue” button with the text **Continue** and an “abort” button with the text **Abort**. If the block contains either STOP or GO but not both, the box will include one button as appropriate but not the other. Note that a DEFINE ALERT block which contains only a STOP command will display a dialogue box where the only option is to abort the method.

The instrument door is unlocked and the method is paused while the alert box is displayed. Clicking on the “continue” button resumes method operation. Clicking on the “abort” button stops the method immediately.

DEFINE APROG

Defines a series of operations as an analysis program.

Usage: Block definition.

Syntax:

```
DEFINE APROG name
       FLOW flowrate
       other commands
   END
```

Each APROG must contain a FLOW command. The FLOW command must be the first operational command in the APROG. Commands that affect settings (such as CAPTION, MODE, CURVE) are placed before the FLOW command.

DEFINE LOOP

Sets up a table of values for repetitive analyses.

Usage: Block definition.

Syntax:

```
DEFINE LOOP name
       LPARAM  variable  ...
       [TIMES count]
               value     ...
       ...       ...
END
```

See Section E.3.3 for details of loop definition.

DEFINE MESSAGE

Defines a message box.

Usage: Block definition.

Syntax:

```
DEFINE MESSAGE name
   [PARAM variable ...]
   other commands
END
```

A message box displays information for the operator. The message remains displayed until a new `MESSAGE` command is issued, an `ALERT` box is displayed or the method ends.

Displaying a message box does not interrupt the method or unlock the instrument door (compare `DEFINE ALERT`).

A message box cannot be closed interactively during method operation. If you want to define a message to be acknowledged by the operator, use a `DEFINE ALERT` block and include a `GO` command in the block.

DETECTION *fc*

Sets detection mode and initial flow path.

Usage: `MAIN` block.

Parameters: *fc* Mode settings.
Values: see table below.

The `DETECTION` command sets detection mode, flow path and reference subtraction according to the following table:

fc value	Detection	Flow path	Reference subtraction
1	1	1	not applicable
2	2	2	not applicable
3	3	3	not applicable
4	4	4	not applicable
1,2	1,2	1,2	none
2-1	1,2	1,2	2-1
1,2,3,4	1,2,3,4	1,2,3,4	none
2-1,4-3	1,2,3,4	1,2,3,4	2-1, 4-3
2-1,3-1,4-1	1,2,3,4	1,2,3,4	2-1,3-1,4-1

Other combinations of flow cell numbers in the *fc* parameter are invalid.

The flow path (but not the detection mode or reference subtraction) can subsequently be changed with the `FLOWPATH` command in an `APROG`. The `INJECT` command will also temporarily set the flow path if required. See Section E.4.1 for more details.

`DETECTION` replaces the `FLOWCELL` command used in earlier versions of the software. `FLOWCELL` is still accepted for compatibility with methods written in earlier versions.

Example:

```
MAIN
    DETECTION 1,2,3,4
    APROG immob
    APROG analyse
END
```

[*] [time] DILUTE position1 position2 mixpos percent

Mixes two samples in a third position. Used for diluting and mixing.

Usage: `DEFINE APROG` and `MAIN` blocks.

Parameters: *time* Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.

position1 Rack position for sample 1.

position2 Rack position for sample 2.

mixpos Rack position for the mixed product.

percent Final percentage concentration of sample 1. Range 5-95%.

The final volume of mixed product is fixed at 200 µl. Thus `DILUTE r1a1 r1a2 r1a3 20` takes 40 µl from r1a1 and 160 µl from r1a2 and mixes them in position r1a3.

A `DILUTE` command is essentially equivalent to two `TRANSFER` commands followed by a `MIX` command. `DILUTE` is however optimized for best performance with the fixed final volume of 200 µl.

`DILUTE` issued in a `MAIN` block will not be recorded in the event log.

The dilution accuracy may be reduced if the two solutions being mixed have significantly different viscosities.

Example:

```
DEFINE APROG incubation
       FLOW 25
 *0:30 DILUTE r1a1 r1a2 r1b1 50
  5:00 INJECT r1b1 50 ! inject 5 min after
                      ! mixing sample
END
```

DIPNEEDLE *position*

Predips needle in operator defined vial to decrease carry-over effects.

Usage: `DEFINE APROG` block.

Parameters: *position* Position in which needle is to be dipped.

10 µl solution is aspirated into the needle from the predip position to rinse the needle. Use this command in connection with repeated sampling from one position where the sample solution differs greatly from buffer in refractive index, pH, ionic strength etc.

Example:

```
DEFINE APROG predip
       FLOW 25
       DIPNEEDLE r1a2     !sample buffer
       INJECT r1a1 50
END
```

END

Defines the end of a block.

Usage: Block definition.

Parameters: None.

ENDLOOP

Defines the end of a loop construction in the `MAIN` block.

Usage: `MAIN` block.

Parameters: None.

EXTRACLEAN

Performs extra cleaning of the IFC.

Usage: `DEFINE APROG` block.

Parameters: None.

When used, `EXTRACLEAN` must always be placed directly after the injection command to which it refers. You can however enter a comment or a `RECOVER` or `FRACTIONS` command between the `INJECT` and the `EXTRACLEAN` command. The cleaning procedure will be executed directly after the injection.

Example:

```
DEFINE APROG injection
       FLOW 20
       INJECT r1a1 30
       EXTRACLEAN  !extra clean-up after injection
END
```

[*] [time] FLOW flowrate [option ...]

Sets flow rate.

Usage: DEFINE APROG and MAIN blocks.

Parameters:	*time*	Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.
	flowrate	Flow rate in µl/min. Range 0-100 µl/min.
Options:	-f	Fill continuous flow pump before starting flow.

The FLOW command is obligatory in a DEFINE APROG block, and may only be preceded by the commands CAPTION, CURVE, KEYWORD, MODE, PARAM and SCALE.

The time specification is not supported by FLOW used in a MAIN block or by the first FLOW command in an APROG.

Between APROGs in the MAIN block, the flow rate is set to 10 µl/min be default. Use a FLOW command in the MAIN block if a different flow rate is required.

Example:

```
DEFINE APROG sample_inject
       FLOW 10
       INJECT r1a1 10
END
```

[*] [*time*] FLOWPATH *fc*

Sets the buffer flow path.

Usage: `DEFINE APROG` block.

Parameters:		
	time	Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.
	fc	Flow cells to be included in the path. Values "1", "2", "3", "4", "1,2", "1,2,3" and "1,2,3,4".

`FLOWPATH` directs buffer flow through the specified flow cells but does not affect the detection mode. Use `DETECTION` in the `MAIN` block to set both detection mode and flow path.

> ***IMPORTANT!***
> `FLOWPATH` does not alter the detection mode set by `DETECTION` in `MAIN`. In single channel detection mode, it is possible to set the flow path to cells where data is not collected. See Section E.4.1 for more details.

Flow path changes are local to the current `APROG`. When the `APROG` is completed, the flow path is restored to the setting that applied at the start of the `APROG`.

The flow path setting can be temporarily overridden by a variable contact time `INJECT` command. The flow path is set to the requirements of the `INJECT` command for the duration of the sample injection (see Section E.4.1).

Example:

```
DEFINE APROG analytes
       FLOW 25
       FLOWPATH 1
       INJECT r1a1 50  !from r1a1 to Fc1
       FLOWPATH 2
       INJECT r1a2 50  !from r1a2 to Fc2
       FLOWPATH 3
       INJECT r1a3 50  !from r1a3 to Fc3
END

MAIN
   DETECTION 1,2,3,4    !multichannel mode
   APROG analytes
END
```

FLUSH

Runs the working tool **Flush** (see Section 7.2.1).

Usage: `MAIN` block.

Parameters: None.

FRACTIONS *startpos volume no*

Recovers dissociating sample during buffer flow. Fractions can only be collected during dissociation phase.

Usage: `DEFINE APROG` block.

Parameters:	*startpos*	Rack position for the first fraction.
	volume	Fraction volume. Range 5-325 µl.
	no	Number of fractions. Range 1-96.

This command must always follow directly after one of the injection commands `INJECT`, `KINJECT` or `QUICKINJECT`. You can however enter a comment or an `EXTRACLEAN` command between the `INJECT` and the `FRACTIONS` command. `FRACTIONS` is not supported for `BIGINJECT` or `COINJECT`.

The minimum time to collect one fraction is 12 s (5 fractions/min). If the requested fraction size is smaller than the volume delivered in this time, the actual fraction size will be the smallest that can be collected at the current flow rate. For example, requesting 5 µl fractions at a flow rate of 100 µl/min would require 20 fractions/min: the actual fraction size will be 20 µl.

If the rack position for the first fraction is specified so that there are not enough positions for all fractions, the fraction collection will stop at the last position in the rack and remaining effluent will be lost to waste.

Example:

```
DEFINE APROG collect
                FLOW 20
                INJECT r1a1 50
                FRACTIONS r2a1 20 5  !collect
          5 fractions
                                     !of 20
          µl
                EXTRACLEAN
          END
```

GO *name*

Displays a button for "continue" function in an ALERT box.

Usage: DEFINE ALERT block.

Parameters: *name* Text of up to 10 characters which will be displayed as the name of the button.

Example:

```
DEFINE ALERT demo
       LTEXT This text is left-adjusted
       CTEXT This text is centered
       RTEXT This text is right-adjusted
       GO OK
       STOP CANCEL
END
```

[*] [*time*] INJECT *position volume*

General purpose sample injection.

Usage: `DEFINE APROG` block.

Parameters:

time Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.

position Rack position for sample.

volume Volume to be injected, or a list of up to fou r volumes separated by commas for variable contact time injections (see Section E.4.3). Range 5-325 µl.

Sample consumption is injected volume + 30 µl.

Example:

```
DEFINE APROG immob
       FLOW 5
       DILUTE r1a5 r1a6 r1b1 50 !NHS/EDC
       INJECT r1b1 35           !NHS/EDC
     * INJECT r2a1 35           !ligand
  5:00 INJECT r1c1 35           !ethanolamine
END
```

KEYWORD *header value*

Defines column and search keyword in report point table.

Usage: `DEFINE APROG` block.

Parameters: *header* Text for the column heading in the report point table.

value Column content, specified as a variable or a constant text value.

The header parameter may not be supplied as a variable.

This command can be used to record concentrations for evaluation.

Example:

```
DEFINE APROG report
      PARAM %conc
      KEYWORD Antibody Mab-451
      KEYWORD Conc %conc
      FLOW 10
    * INJECT r1a1 10
  -15 RPOINT baseline -b
   30 RPOINT plateau
END
```

[*][*time*] KINJECT *position volume disstime*

Low dispersion sample special injection for kinetic applications.

Usage: `DEFINE APROG` block.

Parameters:

time — Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.

position — Rack position for sample.

volume — Volume to be injected, or a list of up to four volumes separated by commas for variable contact time injections (see Section E.4.3). Range 10-250 µl.

disstime — Dissociation time (for buffer flow through flow cell). Range 1-100 000 s.

The `KINJECT` command uses three air segments separating sample from buffer in the autosampler tubing to minimize sample dispersion and improve injection performance. A controlled dissociation time is specified after the injection, during which the sensorgram will not be disturbed by the automatic washing operation following the injection.

Sample consumption is injected volume + 40 µl.

Example:

```
DEFINE APROG kinetics 1
       FLOW 10
     * KINJECT r1a1 30 240 !4 min dissociation
   -20 RPOINT baseline
       INJECT r1a3 5       !Regeneration
END
```

LIFETIME

Displays the time for which a message box has been active.

Usage: `DEFINE MESSAGE` block.

Parameters: None.

This command is recommended for development purposes only. It may be used to determine suitable starting values for the `CLOCK` command.

Example:

```
DEFINE MESSAGE develop
       LTEXT This process has been running for
       LIFETIME
END
```

LOOP *name order*
command
[...]
ENDLOOP

Repeats a sequence of one or more commands, using parameters as specified in `DEFINE LOOP`.

Usage: `MAIN` block.

Parameters:

name — Name of the `LOOP` as specified in `DEFINE LOOP`.

order — One of the keywords `STEP`, `ORDER` or `RANDOM`. This parameter defines the order in which rows in `DEFINE LOOP` will be handled (see also Section E.3.3):

`STEP`	step through the variable sets and then repeat according to the `TIMES` command.
`ORDER`	repeat the first variable set according to the `TIMES` command, then the second set and so on.
`RANDOM`	work through the variable sets in random order.

command — Any command which is valid within the `MAIN` block.

Commands within the `LOOP` construction require parameters in the form of variable names. The names used in the `MAIN` block must be the same as those declared in the corresponding `DEFINE LOOP`.

Example:

```
DEFINE LOOP cycle
       LPARAM  %sample  %pos  %vol
       TIMES 3
               Sample1  r1a1  10
               Sample2  r1a2  15
               Sample3  r1a3  10
END

MAIN
    DETECTION 1
    LOOP cycle RANDOM
       APROG analyse %sample %pos %vol
    ENDLOOP
END
```

LPARAM *variable ...*

Declares variables used in DEFINE LOOP.

Usage: DEFINE LOOP block.

Parameters: *variable* Name of the variable used in the DEFINE LOOP block. See Section E.1.2 for variable naming rules.

The LPARAM command must always be the first command in a DEFINE LOOP block, and may not be used more than once in one block.

Example:

```
DEFINE LOOP cycle
       LPARAM  %sample  %pos  %vol
       TIMES 3
               Sample1  r1a1  10
               Sample2  r1a2  15
               Sample3  r1a3  10
END

MAIN
    DETECTION 1
    LOOP cycle RANDOM
       APROG analyse %sample %pos %vol
    ENDLOOP
END
```

LTEXT [*text*]

Writes left-aligned text in an ALERT or MESSAGE box.

Usage: DEFINE ALERT and DEFINE MESSAGE blocks.

Parameters: *text* Text containing up to 40 characters. A blank line is inserted in the MESSAGE box if *text* is omitted.

Example:

```
DEFINE MESSAGE demo
       TITLE DEMONSTRATION BOX
       LTEXT This text is left-adjusted
       CTEXT This text is centered
       RTEXT This text is right-adjusted
END
```

MAIN

Defines the ordering of APROGs and commands in a method.

Usage: Block definition.

Syntax:

```
MAIN
   other commands
END
```

Each method must contain one and only one MAIN block.

Commands in the MAIN block are not recorded in the sensorgram or the event log. Sensorgram and event log recording are active only during execution of an APROG which is called from the MAIN block.

MESSAGE [*name*] [*parameter ...*]

Displays a MESSAGE box.

Usage: MAIN block.

Parameters:		
	name	Name of the MESSAGE box as specified in DEFINE MESSAGE.
	parameter	Value for display in the MESSAGE box.

Parameters supplied to the MESSAGE must correspond sequentially to the variables declared in the PARAM command in DEFINE MESSAGE.

Within a LOOP construction, parameters may be supplied as either constant values or variable names. Variable names must be the same as those declared in the corresponding DEFINE LOOP. Outside a LOOP, parameters must be supplied as constant values.

The message box remains on the screen until a new message box or an alert box is displayed or the method ends. A MESSAGE command with no parameters will erase any existing message box from the screen.

Example:

```
DEFINE MESSAGE no_break
       LTEXT Do not interrupt this cycle!
END

MAIN
    MESSAGE no_break
    APROG unbroken
    MESSAGE     !clear message box
    ...
END
```

[*][*time*] MIX *position volume*

Mixes sample in autosampler vial.

Usage: `DEFINE APROG` and `MAIN` blocks.

Parameters: *time* Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.

position Rack position for mixing.

volume Volume in µl to mix. Range 15-450 µl.

`MIX` issued in a `MAIN` block will not be recorded in the event log.

Example:

```
DEFINE APROG sample_mixing
       FLOW 10
       TRANSFER r1a1 r1a2 60
       TRANSFER B    r1a2 30
       MIX r1a2 75
       INJECT r1a2 40
END
```

MODE *option* [...]

Sets the data collection parameters for an APROG.

Usage: DEFINE APROG block.

Parameters: None.

Options:	-d*n*	Reading rate interval for each sensorgram in seconds. Range 0.1–10 s.
	-l*n*	Maximum number of collected data points at the current reading rate interval (see Section 9.8). Range 100–15000. Default 12000.

If MODE is set in APROG any settings specified with the **Run:Data collection rate** menu command are overridden. If MODE is omitted, the settings specified with the menu command apply. Using a MODE command at the beginning of an APROG will ensure that the reading rate is correct for the method.

MODE may only be preceded by the commands CAPTION, PARAM, SCALE and KEYWORD.

Example:

```
DEFINE APROG low_resolution
       MODE -d10   ! reading rate 1 point/10 s
       FLOW 10
       INJECT r1a1 10
END
```

NORMALIZE *position*

Runs the working tool **Normalize** (see Section 7.2.1).

Usage: `MAIN` block.

Parameters: *position* Rack position for the glycerol solution (BIAnormalizing solution).

If normalization does not succeed, the method will be terminated.

PARAM *variable ...*

Declares variables used in an `ALERT`, `APROG` or `MESSAGE` blocks.

Usage: `DEFINE ALERT`, `DEFINE APROG`, and `DEFINE MESSAGE` block.

Parameters: *variable* Name of the variable. See Section E.1.2 for variable naming rules.

A block may contain as many `PARAM` commands as required. All `PARAM` commands should precede any other commands in the block.

For variables used in a `DEFINE LOOP` block, see `LPARAM`.

Example:

```
DEFINE APROG injection
       PARAM %position %volume
       FLOW 5
       INJECT %position %volume
END

MAIN
    DETECTION 1
    APROG injection r1a1 20
    APROG injection r2a5 35
END
```

PRIME

Runs the working tool **Prime** (see Section 7.2.1).

Usage: `MAIN` block.

Parameters: None.

[*] [time] QUICKINJECT *position volume*

Fast inject with low sample consumption.

Usage: `DEFINE APROG` block.

Parameters:

time — Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.

position — Rack position of sample.

volume — Volume to be injected, or a list of up to four volumes separated by commas for variable contact time injections (see Section E.4.3). Range 5-325 µl.

This injection uses a single air segment separating the sample from buffer to reduce sample consumption and speed up the injection command. Do not use `QUICKINJECT` in applications where avoidance of sample dispersion is critical.

Sample consumption is injected volume + 10 µl.

Example:

```
DEFINE APROG quickie
       FLOW 20
       QUICKINJECT r1a1 500
END
```

RACK *base type*

Sets the autosampler rack type.

Usage: `MAIN` block.

Parameters: *base* 1 = left position
2 = right position *type* **THERMO_A** = thermo-rack type A
THERMO_B = thermo-rack type B
THERMO_C = thermo-rack type C
MICRO = microtitre plateThe `RACK` command defines the rack positions that are accepted in method pre-run. If no `RACK` command is used, positions are accepted according to the current settings under **Command:Define Rack Base** in the manual control interface (see Section 4.2.4). This ensures that the current settings are valid for the method.

The `RACK` command can be repeated to define alternative permitted racks. The rack positions accepted in method pre-run will however then be restricted to the positions that are common to the specified racks. For example, with the commands `RACK 1 THERMO_A` and `RACK 1 MICRO`, the position r1f10 is not valid (f10 exists in a microtitre plate but not in thermo-rack A).

If a `RACK` command is used, all racks that are referenced in the method must be specified (e.g. if the `MAIN` block includes only `RACK 1 THERMO_A`, positions in rack 2 will not be valid).

If the current rack settings when the method is started do not correspond to the settings requested in the `RACK` command, the method will prompt for rack settings using the **Define Rack Base** dialogue box from the manual control interface (see Section 4.2.4). If you choose **Cancel** from this dialogue box, the method will be aborted. This prompt serves as a reminder that the racks may need to be changed: the only settings which can be entered in the dialogue box are those specified in the `RACK` command.

The `RACK` command may only be preceded by an `ALERT` command within the `MAIN` block. Thus you cannot change rack settings between `APROG`s in a method.

Example:
```
MAIN
    RACK 1 thermo_A
    RACK 2 thermo_B
   APROG analysis
    ...
END
```

RECOVER *position volume [option]*

Recovers injected sample or reagent. The command is placed directly after an inject command.

Usage: `DEFINE APROG` block.

Parameters:	*position*	Rack position to which the sample will be recovered.
	volume	Volume to be recovered. Range 5-350 µl (10-350 after COINJECT). Recommended >25 µl for best precision in recovered volume.
Options:	-s*n*	Sample to recover (COINJECT only). Permitted values for n are 1 (first sample) and 2 (second sample). Default value 1.

RECOVER may only be used with COINJECT, INJECT, KINJECT and QUICKINJECT. You can however enter a comment or an EXTRACLEAN command between the INJECT and the RECOVER command. RECOVER is not supported for BIGINJECT.

The RECOVER command must be placed immediately after the INJECT command to which it applies. If you want to be sure of recovering all the injected sample, specify about 10 µl more than the volume of sample injected.

With COINJECT, the option -s2 may be used to specify recovery of the second injected sample. This option is not valid with other INJECT commands. If no option is specified with COINJECT, the first sample is recovered.

RECOVER and FRACTIONS may not be used with the same INJECT command.

The volume specified for RECOVER is the volume which will be taken from the recovery cup. This is not necessarily the same as the injected volume: e.g. if the recovery cup is primed with buffer as in the example below, the recovered volume should be the sum of the priming volume and the injected volume.

Example:

```
DEFINE APROG recovery
       FLOW 10
       TRANSFER r1b1 R 200  !prime recovery cup
       INJECT r1a1 100
       RECOVER r2a1 310
END
```

RINSE

Runs the working tool **Rinse** (see Section 7.2.1).

Usage: `MAIN` block.

Parameters: None.

time RPOINT *id* [*option ...*]

Sets a report point.

Usage: `DEFINE APROG` block.

Parameters:	*time*	Time specification in format HH:MM:SS, MMM:SS or SSS.
	id	Identification (up to 30 characters, alphanumeric or underscore). The identification is recorded in the report point table.
Options:	-b	Treat point as baseline. Subsequent points will be calculated relative to this point.
	-w*n*	Time window for calculation of slope, mean and standard deviation. Value *n* in format H:MM:SS, range 0-9:00:00. Default value 0:00:05 (5 s).
	-c*n*	Set report point in curve number *n*. Curves are identified by number in the order of their creation: *n* = 0 : sensorgram *n* = 1 : first status curve *n* = 2 : second status curve and so on.

A time window of 0 gives a report point with values from a single data point. A single data point is also used if the window does not include at least one data point on either side of the report point position.

Report points are set on all sensorgrams in multichannel mode.

Example:

```
DEFINE APROG report
       KEYWORD Mab-451
       FLOW 10
     * INJECT r1a1 10
   -15 RPOINT baseline -b
    30 RPOINT plateau
END
```

RTEXT [*text*]

Writes right-aligned text in an ALERT or MESSAGE box.

Usage: DEFINE ALERT and DEFINE MESSAGE blocks.

Parameters: *text* Text containing up to 40 characters. A blank line is inserted in the box if *text* is omitted.

Example:

```
DEFINE MESSAGE demo
       TITLE DEMONSTRATION BOX
       LTEXT This text is left-adjusted
       CTEXT This text is centered
       RTEXT This text is right-adjusted
END
```

SCALE axis min max [option ...]

Sets the default scale for a sensorgram or curve.

Usage: `DEFINE APROG` block.

Parameters: *axis* Axis to scale. Permitted values:

x	abscissa
y	left-hand ordinate
o	right-hand (optional) ordinate

min Minimum value on axis.

max Maximum value on axis.

Options:

-l*text*	Use *text* as label on axis.
-t*value*	Set ticks at interval *value*.
-a	Autoscale.

For cycles where the `SCALE` command is used, the settings define the default scale both during the run and when the result file is opened. After changing the scale, the method scale can be restored using the **Method** option in the **View:Scale** dialogue box.

For cycles where `SCALE` is not used, the default scale setting is **Auto.**

The right hand (optional) ordinate is used for curves activated with the -o option in the CURVE command.

Example:

```
DEFINE APROG fix_scale
        SCALE y 10000 20000
        FLOW 10
        INJECT r1a1 10
END
```

STOP *name*

Displays a button for “abort” function in an ALERT box.

Usage: DEFINE ALERT block.

Parameters: *name* Text of up to 10 characters which will be displayed as the name of the button.

Example:

```
DEFINE ALERT demo
       TITLE DEMONSTRATION BOX
       LTEXT This text is left-adjusted
       CTEXT This text is centered
       RTEXT This text is right-adjusted
       GO OK
       STOP Cancel
END
```

TEMP *degrees*

Sets the temperature at the flow cell.

Usage: `MAIN` block.

Parameters: *degrees* Required temperature (ºC). Range 4-40 ºC.

The `TEMP` command causes the method to wait until the temperature has stabilized at the requested value. This takes typically about 1 hour for a change of 5 ºC. Temperature changes can only be requested from the `MAIN` block (between `APROG`s). Buffer flow continues at 10 µl/min during the temperature stabilization process.

Do not set the temperature to lower than 20ºC below ambient. The instrument is not guaranteed to maintain a temperature interval greater than 20ºC, and requesting a lower temperature can result in an indefinite wait time. To stop the method during temperature stabilization, you must use the emergency stop Ctrl-Break (see Section 4.4.14).

The `TEMP` command does not affect the temperature in the autosampler compartment. Samples and buffer are fully equilibrated to the flow cell temperature during passage through the IFC.

Example: This example uses a `LOOP` construction to repeat an analysis at a series of temperatures.

```
DEFINE LOOP temp_series
       LPARAM %degrees
       TIMES 1
              10
              15
              20
              25
       END

DEFINE APROG analysis
       ...
END

MAIN
    LOOP temp_series STEP
       TEMP %degrees
       NORMALIZE r2f2
       APROG analysis
    ENDLOOP
END
```

TIMES *count*

Sets the repeat count for sets of variables.

Usage: `DEFINE LOOP` block.

Parameters: *count* Repeat count. Values 1-5000 (default 1).

The repeat count applies to subsequent rows or variables within the `DEFINE LOOP` until a new repeat count is set. The repeat count may be set as many times as required in a `DEFINE LOOP`.

Setting a repeat count of 3 means that subsequent rows will be processed three times.

Example:

```
DEFINE LOOP cycle
       LPARAM  %sample  %pos  %vol
       TIMES 3
               Sample1  r1a1  10
               Sample2  r1a2  15
               Sample3  r1a3  10
END

MAIN
    DETECTION 1
    LOOP cycle RANDOM
       APROG analyse %sample %pos %vol
    ENDLOOP
END
```

TITLE *text*

Sets the title bar text for an `ALERT` or `MESSAGE` box.

Usage: `DEFINE ALERT` and `DEFINE MESSAGE` blocks.

Parameters: *text* Text containing up to 40 characters. The title may not be left blank.

Example:

```
DEFINE ALERT demo
       TITLE DEMONSTRATION BOX
       LTEXT This text is left-adjusted
       CTEXT This text is centered
       RTEXT This text is right-adjusted
       GO OK
       STOP CANCEL
END
```

[*][*time*] TRANSFER *from-pos to-pos volume*

Transfers sample between autosampler vials.

Usage: `DEFINE APROG` and `MAIN` blocks.

Parameters:

time	Time specification in format HH:MM:SS, MMM:SS or SSS. An asterisk (*) before the time specification sets this event time as a zero time reference for subsequent time specifications.
from-pos	Rack position to transfer from.
to-pos	Rack position to transfer to.
volume	Volume in µl to transfer. Range 5-450 µl.

The special value B for the *from-pos* parameter transfers liquid from the eluent vessel.

The special value R for the *to-pos* parameter transfers liquid to the recovery cup.

`TRANSFER` issued in a `MAIN` block will not be recorded in the event log.

Example:

```
DEFINE APROG recovery
       FLOW 10
       TRANSFER r1b1 R 200  !prime recovery cup
       INJECT r1a1 100
RECOVER r2a1 310
END
```

UNCLOG

Runs the working tool **Unclogging** (see Section 7.2.1).

Usage: `MAIN` block.

Parameters: None.

WAIT *time*

Waits for a specified time.

Usage: `DEFINE APROG` and `MAIN` blocks.

Parameters:	*time*	Wait time in the format HH:MM:SS, MM:SS or SSS.

Flow continues during a wait.

Within a `LOOP` construction, parameters may be supplied as either constant values or variable names. Variable names must be the same as those declared in the corresponding `DEFINE LOOP`. Outside a `LOOP`, parameters must be supplied as constant values.

Example:

```
DEFINE LOOP wait_time
       LPARAM %delay
       TIMES 2
              5:00
              10:00
MAIN
    DETECTION 1
    WAIT 2:00        !2 minute delay
    LOOP wait_time STEP
       WAIT %w       !vary wait time
       APROG analyse %sample %pos %vol
       APROG regenerate
    ENDLOOP
END
```

[*time*] WASH *item*

Washes IFC or autosampler needle.

Usage: `DEFINE APROG` and `MAIN` blocks.

Parameters: *time* Time specification in format HH:MM:SS, MMM:SS or SSS.

item Item to wash:
f = flow cell area
i = all IFC channels
l = IFC sample loop
n = needle
s = IFC sample line

`WASH f` and `WASH i` require a non-zero flow rate. An attempt to execute these commands with the flow rate set to zero results in a run-time error.

Example:

```
DEFINE APROG immob
       FLOW 5
       DILUTE r1a5 r1a6 r1b1 50
       TRANSFER r1a6 r1b1 70
       MIX r1b1 115
       INJECT r1b1 35
       WASH n              ! extra needle wash
       INJECT r2a1 35
       INJECT r1c1 35
END
```

WASHREC *[option]*

Washes recovery cup. With an option the operator can use washing liquid from a vial position.

Usage: `DEFINE APROG` and `MAIN` blocks.

Options	-p *pos*	Specify rack position for washing solution.

If the -p option is not specified the recovery cup will be washed with eluent buffer.

```
Example: DEFINE APROG recovery
                FLOW 25
           2:00 WASHREC -pr2a1
                INJECT r1a1 50
                RECOVER r2b1 55
         END
```

Index

INSTRUMENT HANDBOOK