RSpace for X Window User's Manual
TR92-007

Rickard E. Faith

Mon Apr 26 13:42:52 1993

Contents

1 Introduction

RSpace is a reciprocal-space modeling tool for use by X-ray crystallographers during the planning of data-collection strategies.

Version 3 of RSpace contains all of the old version 1.3 code which was originally used on VMS and UNIX[1] systems equipped with an Evans and Sutherland PS-300 series graphics display. Although version 3 has not been tested with a PS-300, compilation of a PS-300 version should be possible.

New with version 3, however, is the added ability to use RSpace on UNIX machines running the X Window System.

Users who have used version 1.3 on the PS-300 will want to read Section 5.3 on page 19 for more information about using the mouse for navigation. This is especially important, since navigation is possible *without using the dials!*

The following overview from the original documentation[2] explains the basic features of RSpace:

> It [RSpace] provides a graphical representation of Ewald's construction in which the crystal frame can be rotated, depicting the volume of reciprocal space that has been swept by the projection of the area detector on Ewald's sphere.
>
> The program allows the user to enter experiment-specific parameters, and then specify collection scans by jumping to a chosen

[1] UNIX is a registered trademark of AT&T.

[2] Harris, Mark R.; and Fitzgibbon, Matthew P. *RSpace Users Manual.* The University of North Carolina at Chapel Hill, Department of Computer Science: TR88-032 (June 1988 (Revised December 1988)).

crystal orientation and interactively rotating about a goniometer axis. The volume of reciprocal space swept by the scan is then displayed graphically, superimposed on various structures representing the goniometer, the asymmetric volume of reciprocal space, the laboratory and the crystal axes. Gaps and redundant collections in the strategy can be identified as the simulated collection proceeds. Scanned regions may optionally be mapped to symmetry-equivalent positions; this allows depiction of a contiguous scanned volume when data is collected in regions of reciprocal space that are actually distant from each other, but which have symmetry equivalents that are adjacent.

Both rectangular and circular detectors can be modeled, for use with the Xuong-Hamlin, Xentronics and FAST systems, amongst others.

Broad-spectrum Laue diffraction can also be modeled.

2 Command-line Options

The command line has the form:

`rspace` [RSpace options] `--` [`xclient` options]

The RSpace options are as follows:

-d device_name This option specifies the device for graphical output. In the X Window version, only two devices are available:

- X
- Null_device (this is the default)

(Some users will want to re-compile RSpace for use with PS-300 systems. Version 3 has not been tested with a PS-300 system, and the use of version 3 with the PS-300 is discouraged. However, after special compilation, the 'ps300' device name will be available, and should be used with the `-d` option.)

-i setup_file This option specifies a file of initial commands to execute. The default file is `rspace.init` (in this file, lines starting with `#` or `!` are considered comments). Use the `-s` option to suppress the loading of this file.

-s This option will suppress the loading of `rspace.init`.

-h This option will cause a list of options to be printed and will terminate RSpace.

-t target_system_name This option specifies an Internet address for the system on which `xclient` will be executed. If this option is used, RSpace will use `rsh` to execute the client remotely. This may require the user to type a password to access the account on the other system (if that system does not "trust" the current system). The default behavior of RSpace (i.e., if `-t` is not used) is to `fork(2)` the client on the current machine.

-c client_path If `xclient` is not in the user's default `PATH` on the target system (which was specified with the `-t` option), then this option can be used to specify the path where `xclient` exists. N.B.: `client_path` should *include* the name `xclient`, so this option can also be used to select other client programs.

-D Depending on how RSpace was compiled, this option may activate special debugging, and should not be specified during normal use of RSpace.

The double dash (`--`) delimits the RSpace options from the `xclient` (or other client) options. For `xclient`, the following standard X options[3] are supported (all of these options require one argument):

-bg, -background Background color

-fg, -foreground Foreground color

-bd, -bordercolor Color of widget (i.e., button and dial) borders

-bw, -borderwidth Width of widget borders (in pixels)

[3] A more complete list of the "standard X Toolkit command-line options" can be found in Adrian Nye and Tim O'Reilly. *X Toolkit Intrinsics Programming Manual*, 2nd ed., Vol. 4. O'Reilly & Associates, 1990.

-display Server. The argument specifies an X Window display to use for the `xclient` window (i.e., `-display myworkstation.cs.unc.edu:0.0`).

-fn, -font Font name

-geometry Size and position of the `xclient` window (in pixels). For example, `-geometry -0-0` will place the window in the upper left hand corner of the screen; and `-geometry 1000x1000+0-0` will place a very large window at the upper right hand corner of the screen.

-xrm Set the resource specified. This option allows the user to set, on the command line, resources which would otherwise only be available through the `.Xdefautls` file. For example, `-xrm '*background red'` has exactly the same effect as `-bg red`.

No non-standard options for `xclient` are available.[4]

3 Commands

3.1 Introduction

Commands in RSpace are, in general, entered from the keyboard with the X Window focus in the `xterm` window which was used to launch RSpace. Depending on how your window manager (i.e., `twm`, `uwm`, `mwm`) is set up, you will probably have to move the mouse cursor from the RSpace `xclient` graphics window into the `xterm` window. You may have to press the left mouse button.[5] When you have changed the "focus" to the `xterm` window, you will be able to type letters after `RSpace <> :` prompt. If you are having trouble with this step, or if you would like your window manager to behave differently, please consult your local X guru for help. Because X is such a complex system, the author generally assumes some familiarity with X. There are several books available which explain the X Window System. One of the most thorough is:

[4] Actually, this may be a lie. In particular the `-D` option may be available for debugging purposes (depending on how `xclient` was compiled, but this shouldn't be of concern to regular users.

[5] Because X is highly configurable by the user, it is impossible to predict exactly what you will have to do to change the focus. It is also possible that the focus can be changed in several different ways (i.e., via the mouse or via the keyboard).

Quercia, Valerie; and O'Reilly, Tim. *X Window System User's Guide for Version 11 of the X Window System.* O'Reilly & Associates, 1989.

A printing of this guide may have come with your computer. If you use Motif, a special edition of this manual is available.

3.2 Entering Commands

All of the commands listed below can be abbreviated with a single letter. If you do not enter any parameters after the command, you will be prompted for them. Entering a carriage-return after a prompt is equivalent to entering the default for that parameter (i.e., if you want the default, all you need to do is hit return).

If you enter some but not all of the parameters after the command, the remaining parameters will be set to their default values and you will not be prompted. Further, if any of the parameters are not interpretable, the default will be used.[6] This method is used so that incorrect parameters in script files will not cause RSpace to halt. Therefore, if you get an error entering a command, you must re-enter it with correct parameters.

Commands which cause a major re-build of the image structures are entered at the `Setup <> :` prompt, whereas those that merely manipulate the structures are entered at the `RSpace <> :` prompt.

3.3 Getting Help

The the `RSpace <> :` and `Setup <> :` prompts, the commands "help" and "?" are available. These commands provide help on a particular topic (i.e., "help setup" provides help on the setup command). To exit the help system, type "quit" at the `help <> :` prompt.

3.4 Errors

Most errors from RSpace are presented in a VMS-like format: an initial question-mark is followed by the name of the originating routine, a severity

[6] The "jump" command is an exception to these rules: it requires three parameters and has no defaults values.

code, and an explanation of the condition that caused the message to be sent. The severity code are as follows:

s: Success

i: Informational

w: Warning (The user may have done something wrong, but execution can probably continue.)

e: Error (The user has done something wrong, and the path of execution has changed because of it.)

f: Fatal error (These should cause RSPACE to abort execution.)

a: Apology

Errors from the `xclient` program have a different form. Informational messages are printed without any prefix or suffix. "ERROR," "INTERNAL ERROR," and "FATAL ERROR" messages are prefixed with the error type, and have, as parenthetical suffix, the name of the routine which caused the error. None of these errors are caused directly by the user. The errors arise because of programming errors or poor communication with the RSPACE program. Only fatal errors will cause the `xclient` to terminate execution, although execution after any error may not produce acceptable results.

3.5 Geometry[7]

In Ewald's construct, the active collection area is defined by the projection of the area detector from the centre of Ewald's sphere to its surface. For a square detector, this patch is dish-shaped with curved edges, a most inconvenient shape to display with vector graphics. For this reason we represent the patch as a flat surface with the same shape as the original detector, with the edges of the patch touching Ewald's sphere. This results in the corners being a little too far out, and the centre a little too far in. If this discrepancy becomes bigger than about 10% of Ewald's radius, we draw radiating lines from the true centre of the detector to its rim, but still don't round the edges. The

[7] This section taken from Harris, Mark R.; and Fitzgibbon, Matthew P. *RSpace Users Manual*.

resulting pyramid allows the true curved surface to be easily visualised. Not rounding the edges seems reasonable, since the detectors that can move in close enough to have significant curvature are likely to be circular and thus have no protruding corners. If a range of wavelengths has been specified, the projections corresponding to the two extremes are drawn, with corners joined to form a box.

3.6 Coordinate Systems[8]

All coordinate systems are right handed. The laboratory axes are centred on the crystal, with the positive x axis running along the X-ray beam from the crystal away from the source, the positive y axis pointing towards the ceiling, and the z axis in the horizontal plane. The conventional real and reciprocal crystal axes, a, b, c and a^*, b^*, c^* are used.

The goniometer axes are as follows: Looking from the top of the goniometer with the omega motor at the bottom, a positive *omega* rotation is anticlockwise. At $chi = 0$, the *phi* motor is at the bottom of the *chi* circle, and a positive *phi* rotation is clockwise. Looking from the beam source towards the crystal, with $omega = 0$, a positive *chi* rotation is clockwise. If your system is not like this, the sense of rotation about each axis can be reversed using the command "sense_of_rotation".[9]

If your *phi* motor is opposite the *omega* motor when $chi = 0$, you should change the value of `PHI_START` in `rspace.h` to -1.0 and recompile.

For the detector position, a positive angle corresponds to an anticlockwise rotation when viewed from above.

3.7 The Full Command Set[10]

- setup (These commands require a rebuild of major data structures.)
  - ask_for_parameters
  - group_number

[8] This section taken from Harris, Mark R.; and Fitzgibbon, Matthew P. *RSpace Users Manual.*

[9] There have been reports that this command does not function as documented.

[10] This section taken from Harris, Mark R.; and Fitzgibbon, Matthew P. *RSpace Users Manual.*

  - dimensions_of_cell
  - zone_angles
  - type_of_detector
  - move_detector (this command can also be invoked from the top-level prompt)
  - resolution (this command can also be invoked from the top-level prompt)
  - volume_mapping
  - keep_lab_axes_fixed
  - limits_change
  - sense_of_rotation
  - Wavelength
  - Unique_volume_position
  - exit_from_setup

- omega
- chi
- phi
- jump_to_position
- move_detector
- resolution_change
- undo_last_move
- list_saved_scans_info
- Save_scans
- display_saved_scans
- information_on_current_state

- follow_path
- write_parameter_file
- yank_input_stream_from_file
- help topic
- new_features
- Command_system
- Doze
- Pause
- quit

When you have entered setup mode, you can either set parameters with the individual commands listed above, or invoke "ask" mode, which will then prompt you for all the required parameters.[11]

If you choose "ask" mode, the dialogue will pause at some point, as the program busies itself collating the information that you have given it, in order to display your configuration on the screen. You will then be asked to choose your unique-volume orientation based on the current parameters.

If you choose to change parameters individually, you must indicate completion with the command "exit" when you have made all the changes you desire. If you did not specify the unique-volume orientation explicitly, you will be prompted for this after the other information has been collated.

On exit, all information about previous scans is lost.

If you choose to quit at the setup prompt, changes you have made during the current setup session will not be undone, but nor will they be incorporated into the graphics display. It is not a recommended maneuver unless you have just entered setup. Remember that you can save and restore parameter sets with the "write" and "yank" commands. Quitting at a subcommand within setup is quite safe, and will not change the parameters for which you were being prompted.

[11] "ask" mode is invoked by typing "ask" at the first `Setup <> :` prompt. This is the default for the first prompt, so "ask" mode can be entered by pressing return at the first `Setup <> :` prompt.

ask_for_parameters starts a dialogue under setup, prompting for all the required parameters. It exits setup mode automatically when completed or if a quit is encountered. Quitting early allows you to skip the later questions if you know that you are going to want the defaults.

group_number allows you to specify the space-group number to define symmetry. If you do not know your space-group number and enter 0, or if the program does not recognize your response, you will be prompted for the Laue-group number. If you do not know that, consider whether crystallography is really the career for you.

dimensions_of_crystal sets the cell parameters, for use in constructing the crystal axes. Any units may be used for the cell edge lengths, since they are scaled, but angles should be in degrees.

zone_angles sets the zone angles, and the orientation of the crystal.

The zone angles are the set of *omega*, *chi* and *phi* goniometer rotations at which the crystal axes line up with laboratory axes, and thus account for the arbitrary orientation of crystals in their capillaries. At the zone, individually-mounted examples of a particular crystal form will be aligned identically.

As well as the three angles, it is also necessary to specify which crystal axes and planes should align with which laboratory axes and planes. A numbered menu of options is presented to facilitate this. When setup is exited, the program automatically executes a "jump" to zone.

type_of_detector changes the shape, dimensions and number of the detectors, allowing for both square Xuong-Hamlin detectors and circular Xentronix ones. The program was written for Xuong-Hamlin type detectors, as collection strategies are more complex than for Xentronix types, but differently shaped detectors are easily incorporated.

We support Xentronics detectors simply by changing the shape of the detector, and do not account for the fact that some of these systems have fixed *chi* angles. To use the program with these systems, you must currently specify the fixed *chi* angle for each "jump" and "zone" command. Setting the upper and lower mechanical limits on chi to 1 degree either side of the fixed value will help prevent mistakes.

Up to two detectors may be specified, but their spatial relationship will be fixed, and they will be mapped as a single unit, based on the position of the centre of the second detector. Although the detectors should be represented as projections on to a sphere, RSpace takes the liberty of assuming that they are planar. The discrepancy is small.

move_detector moves the detector to a new angle and distance. It can be invoked from within setup, or during a scan sequence. The angle is specified in degrees, with positive angles corresponding to an anti-clockwise rotation when viewed from above. The distance is measured from the crystal to the centre of the detector, in millimeters.

resolution_change draws a sphere of resolution at an appropriate radius. This is only a visual guide, and is not currently used in any calculations.

volume_mapping[12] selects or deselects unique-volume and freidel mapping for subsequent scans.

For practical reasons, such as mechanical limits on the movement of the goniometer, it is sometimes impossible to collect a complete dataset in a contiguous region of reciprocal space, and collections from symmetry-equivalent positions have to be made. To address the difficulty in judging how such non-contiguous volumes complement each other, the program can optionally map all swept volumes to equivalent regions of a single unique volume. When mapping is enabled, attempts to scan outside the unique region are pushed into it, according to appropriate symmetry rules.

When interactive positioning has been chosen, a mapped detector will be created according to the mapping function appropriate to the starting position. Note that this mapping may not be appropriate once you have moved the detector.

The simplest mapping scheme is "all_bijvoets," which maps all scans into a single unique volume that is displayed on the screen. "freidel" mapping maps the scans into one displayed hemisphere, and "bijvoets" mapping maps the scans into one of two displayed volumes—the specified one and its Freidel mate, related by inversion symmetry.

[12]This option may not work as expected in the X version.

The mapping of the whole detector is based on the position of the centre of the detector at 10 intermediate points between each drawn position, so part of the detecctor may appear outside the chosen volume when near the boundaries. For two-detector configurations, the mapping of both detectors is based on the position of the centre of the second detector. This is fine if the second detector is being used for simultaneous collection of Freidel pairs, but is a limitation under other circumstances. To simulate two independent detectors, you must currently ask for one, perform the scans, and then repeat them with the detector moved to the new position.

The symmetry rules for the mapping operations have been determined for all the protein space groups, but not all have been tested. In particular, there are pairs of space groups that belong to the same laue group, but whose unique volumes align with different axes. There is a mechanism to compensate for this, but it has only been invoked for groups 151/152 and 153/154.

The mapping functions are complex, and are difficult to test fully, so please check their behavior on simple cases when first used on new space groups.

The default is no mapping.

keep_lab_or_crystal_fixed[13] determines which frame of reference stays still during a scan. The default is crystal axes fixed.

limits_change sets the mechanical limits of the machine in terms of lower and upper bounds on *omega*, *chi* and *phi*. Setting the upper and lower mechanical limits on chi to 1 degree either side of the fixed value will help prevent mistakes.

Unique_volume_position allows blind positioning of the depicted unique volume.

The unique volume is the asymmetric unit of reciprocal space, and thus includes the minimum amount of data required for a complete collection.

[13]This option may not work as expected in the X version.

For most space groups there is a finite number of these volumes, since they must be aligned with major crystal axes, but for triclinic and monoclinic groups there are an infinite number because of rotational degrees of freedom. Triclinic groups have two degrees of freedom, and monoclinics have one. Specifying a unique-volume orientation requires a unique-volume number representing one of a finite number of tessellating volumes defined by RSpace, and one arbitrary rotation for monoclinic groups, or two arbitrary rotations for triclinic groups.

The command is primarily intended for use in scripts, using values determined interactively as setup mode is exited. The unique volume is depicted graphically, and also used by the mapping functions.

Wavelength specifies the radiation wavelength. It prompts for a range, so that Laue diffraction can be modeled. Specifying two identical values gives a monochromatic source.

sense_of_rotation changes the direction of rotation about the goniometer axes. Three integers specify the direction for *omega*, *chi* and *phi*. $+1$ gives the rotation senses defined above under "coordinate systems," -1 reverses the direction. You should use this command if your system does not correspond to the definitions above.

omega performs an incremental scan about the *omega* axis.

chi performs an incremental scan about the *chi* axis.

phi performs an incremental scan about the *phi* axis.

jump_to_position jumps to position, without sweeping a volume.

Successive jumps replace each other, so there is no need to "undo" them. The word "zone" can be used as a substitute for the triplet of numbers representing the zone angles.

undo undoes the last jump or increment; successive calls work backwards. Once a scan is undone, it cannot be redisplayed. Use the command "Save" to save scans.

yank_input_stream_from_file starts reading input from a file.

These files can be created with an editor, entering the commands just as if you were answering the interactive prompts. Comments are flagged by a `#` or a `!` in the first column. Command is returned to the keyboard at the end of the file.

write_to_file writes out a file readable by "yank."

It prompts for whether to write just setup parameters, just scan information, or both.

A file of setup parameters can be used as a setup file, and if renamed to `rspace.init`, will be read at each subsequent invocation of RSpace, allowing site-specific or user-specific information to be setup automatically. A file with setup parameters and scan information can be used to reconstruct complete sessions, or to record final results for entry into the real goniometer-driving program.

information_listing list the current parameter values on the screen.

follow_path shows how you got to your current position.

help topic gives help on any topic at the `RSpace <> :` and `setup <> :` prompts, "help" gives help on the current activity at any prompt.

If more than one page of help is presented, you will be prompted to continue or quit ('c' or 'q') after each screenful.

new_features types out the latest release notes.

Doze hibernates the program for a few seconds.

The command is useful in scripts for demos and dodgy ethernet links. An optional parameter specifies the number of seconds.

Pause pauses the program until continuation is requested from the keyboard.

A "quit" will terminate a script if that is where the Pause was issued.

System_command[14] submits a command to the operating system, returning when it is completed.

[14] This command does not function properly under the X Window version. It should not be used, and may be deleted in the future.

quit aborts the current activity, but note that quitting from setup mode cannot restore parameters that were changed. Use "write" to save a state.

exit kills the program after confirmation if issued at the top-level prompt, or returns to the top-level prompt if issued from setup mode, rebuilding major datastructures as it goes.

EOF (Ctrl-D under UNIX) from the keyboard causes immediate exit from the program at any prompt.

4 Known Bugs[15]

1. "Undo" leaves droppings. These are cleaned up on subsequent scans, but are irritating none the less.
2. "Keep_lab_fixed" behaves a bit squirrely.
3. The goniostat representation loses its outer ring when the resolution limit is set below 2 Angstroms or above 4 Angstroms.

5 Using the X Window Interface

The `xclient` program is started automatically when RSPACE is executed. This program provides the graphics display for RSPACE.

The laboratory axes are represented by solid lines, the crystal axes by short dashed lines, and the reciprocal axes by long dashed lines (with muted colors). The x, a, and a^* axes are blue; the y, b, and b^* axes are red; and the z, c, and c^* axes are green.

The goniometer is drawn in red, with a *chi* circle mounted on a stub representing the *omega* motor, and carrying a line segment representing the capillary tube. Further, there are three pink lines drawn along the *omega*, *chi*, and *phi* rotation axes.

The sphere of resolution is red.

[15]This section taken from Harris, Mark R.; and Fitzgibbon, Matthew P. *RSpace Users Manual.*

The unique volume is blue.

The detector is initially red, but changes color with each scan. Each scanned volume is drawn in the color of the last static detector. When the goniometer is being interactively positioned with mapping enabled, both the mapped and the unmapped detectors are displayed. Mapped and unmapped lab axes are also displayed during the interaction.

5.1 The Dials

In order to move a dial, move the mouse cursor into the square which represents the dial. When this happens the dial will "highlight," letting you know that that dial is selected. The cursor will also change into a little dial shape. Pressing the left mouse button will cause the value of the dial to move to the position pointed to by the cursor. If the left mouse button is held down, the cursor can be "dragged," which will interactively change the value of the dial.

Note that the "Omega," "Chi," and "Phi" dials are only active during a scan. Further, only one of these three dials can be active at any one time.

The following dials are available:

Omega Used during "omega" scans.

Chi Used during "chi" scans.

Phi Used during "phi" scans.

X Rot Rotates the image about the x axis (horizontal).

Y Rot Rotates the image about the y axis (vertical).

Z Rot Rotates the image about the z axis (orthogonal to the plane of the screen).

Scale Scales the size of the image.

Note that the "X Rot," "Y Rot," and "Z Rot" dials all move the image with respect to the "world coordinate system." This is an axis system which is anchored to the screen.

5.2 The Buttons

To press a button, move the mouse cursor into the button area. The button will "highlight" when this occurs. Then press the left mouse button. The buttons which "toggle" will light up when in their "on" state. All of the toggles can be reset by pressing the "Show All" button.

The following buttons are available:

Accept Scan This button is "grey" except when a scan is in progress. When a scan is in progress, pressing this button will accept the current "Omega," "Chi," and "Phi" values.

Quit Scan This button is also "grey" except during a scan, when the button is used to abort the scan and return the "Omega," "Chi," and "Phi" dial to their pre-scan values.

Reset Rotation Pressing this button will return the image to the canonical view. This is often used when one becomes confused about the orientation of the image.

Show All Some of the structures can be hidden from view (see buttons below). Pressing "Show All" will make all of the structures visible.

Info Pressing this button will display some information about the `xclient` program and its author(s).

Quit Pressing this button will terminate the RSpace session.

Stereo Pressing this button will activate a "lateral" stereo view of the image. This view consists of two slightly different images displayed side by side. These images can be viewed in stereo through the use of mirrored stereo goggles.[16]

Unique Toggles the display of the unique volume.

Resolut Toggle the display of the sphere of resolution.

Lab Ax Toggle the display of the lab axis.

[16] Some people can "free fuse" stereo images without the aid of goggles. Doing this gives me a headache–the goggles are fairly inexpensive and are highly recommended.

Crys Ax Toggle the display of the crystal axis.

Recp Ax Toggle the display of the reciprocal axis.

Goni Ax Toggle the display of the goniometer.

Scans Toggle the display of the scans.

5.3 Scanning in Stereo

When viewing the display in stereo, the dials and buttons are not visible (because they are not presented as stereo images). Therefore, some other methods of rotating the image and performing scans have been added. (These navigational methods are also available outside the stereo mode.)

Imagine that the viewport has been overlaid by a tic-tac-toe grid:

```
     |     |
  1  |  2  |  3
     |     |
-----------------
     |     |
  4  |  5  |  6
     |     |
-----------------
     |     |
  7  |  8  |  9
     |     |
```

Clicking the left mouse button on a single point will toggle the display of this grid.

The image can be rotated about the x axis (the horizontal axis) by pressing the left mouse button in box 2 or box 8 and then dragging the mouse up or down.

Rotation about the vertical axis (the y axis) is performed by pressing the left mouse button in box 4 or box 6 and then dragging the mouse left or right.

In a similar fashion, pressing the left mouse button in boxes 1, 3, 7, or 9 allows rotations in the z axis.

When doing a scan, the right mouse button can be used for omega, chi, or phi rotations, depending on the type of scan in progress. Further, the middle mouse button will toggle the display of a menu which will either accept or abort the scan. When the menu is displayed, the left mouse button must be used to select the menu option.

6 X Resource Hierarchy

X resources for `xclient` are fully supported. This allows the user to customize the size and color of the dials and buttons in the `xclient` window. Resources can be specified on the command line using the `-xrm` option, but are usually placed in the `.Xdefaults` file in the users home directory. Below, the `xclient` recourse hierarchy is listed:

- RSpace
  - dialBox (`dial`) Note that for dials, the `font` and `label` resources are not yet implemented.
    - Omega
    - Chi
    - Phi
    - X Rot
    - Y Rot
    - Z Rot
    - Scale
  - buttonBox (`button`)
    - Accept Scan
    - Quit Scan
    - Reset Rotation
    - Show All
    - Info
    - Quit
  - toggleBox (`button`)

    - Stereo
    - Tip
    - Unique
    - Resolut
    - Lab Ax
    - Crys Ax
    - Recp Ax
    - Goni Ax
    - Scans
  - graphicsBox
  - infoBox
    - infoText
    - infoDone (`button`)

The user should use `*` to delimit members in the hierarchy, since it is imcomplete. Enough information is, however, provided to uniquely specify any widget of interest. A set of example entries in the `.Xdefaults` file are shown below:

```
RSpace*background:          navyblue
RSpace*foreground:          white
RSpace*indicatorColor:      yellow
RSpace*borderColor:         tan
RSpace*font:                6x13
RSpace*main.geometry:       600x600  # The main is necessary here

RSpace*Omega*height:        75
RSpace*Omega*width:         75
RSpace*Omega*borderColor:   red
RSpace*Chi*height:          75
RSpace*Chi*width:           75
RSpace*Chi*borderColor:     red
RSpace*Phi*height:          75
RSpace*Phi*width:           75
RSpace*Phi*borderColor:     red
```

```
RSpace*X Rot*height:      75
RSpace*X Rot*width:       75
RSpace*Y Rot*height:      75
RSpace*Y Rot*width:       75
RSpace*Z Rot*height:      75
RSpace*Z Rot*width:       75
RSpace*Scale*height:      75
RSpace*Scale*width:       75
```

6.1 Button Resources

Name	Class	RepType	Default Value
background	Background	Pixel	XtDefaultBackground
bitmap	Pixmap	Pixmap	None
borderColor	BorderColor	Pixel	XtDefaultForeground
borderWidth	BorderWidth	Dimension	1
cursor	Cursor	Cursor	None
font	Font	XFontStruct*	XtDefaultFont
foreground	Foreground	Pixel	XtDefaultForeground
height	Height	Dimension	text height
highlightThickness	Thickness	Dimension	2
insensitiveBorder	Insensitive	Pixmap	Gray
internalHeight	Height	Dimension	2
internalWidth	Width	Dimension	4
justify	Justify	XtJustify	XtJustifyCenter
label	Label	String	NULL
width	Width	Dimension	text width

6.2 Dial Resources

Name	Class	RepType	Default Value
background	Background	Pixel	XtDefaultBackground
bitmap	Pixmap	Pixmap	None
borderColor	BorderColor	Pixel	XtDefaultForeground
borderWidth	BorderWidth	Dimension	1
cursor	Cursor	Cursor	None
font	Font	XFontStruct*	XtDefaultFont
foreground	Foreground	Pixel	XtDefaultForground
height	Height	Dimension	100
highlightThickness	Thickness	Dimension	2
indicatorColor	Color	Pixel	Yellow
insensitiveBorder	Insensitive	Pixmap	Gray
internalHeight	Height	Dimension	2
internalWidth	Width	Dimension	4
markerLength	Length	Dimension	6
label	Label	String	NULL
width	Width	Dimension	100

7 History

RSpace was originally written circa 1988 by Mark R. Harris and Matthew P. Fitzgibbon at the University of North Carolina at Chapel Hill. The most recent version of this work is version 1.3. The original "clients" for this work were Professor Charles W. Carter, Jr. and Robert P. Jones in the Biochemistry department at UNC.

In 1990, version 2.0 of RSpace was created at Purdue by B. H. Wang. This author does not know if this version was ever publically released.

In 1991, Rickard E. Faith began work on an X Window System port of RSpace 1.3. Robert Jones has agreed to be the "client" for this port, which is currently being released with version numbers in the 3.x range. At the present time, *none* of the modifications from 2.0 have been included.

8 Overview of the X Window System Port

8.1 Background

At the time RSpace was created, the Evans and Sutherland PS-300 series graphics display was, so I am told, an excellent graphics output device in terms of speed and resolution. However, PS-300 display units were very expensive. Today, PS-300 compatible hardware from Evans and Sutherland remains a niche product when compared to the wide availability of UNIX workstations supporting the X Window System.

In view of the simple wire-frame modeling requirements of RSpace, a dedicated graphics processor is no longer necessary. Although hardware support for advanced stereo viewing is desirable, lateral stereo under X is a usable alternative for RSpace needs. Because of the design considerations outlined below, it is expected that ports to the "stereo X," gl, and phigs environments will be relatively easy. Some of these environments will support stereo using active filters or other methods.

8.2 Paradigm

The RSpace program was divided into two separate programs. One is essentially the "old" RSpace with the PS-300 interface replaced by a generic socket-based interface. The other is a "graphics processor." The "old" section will be called the "server," and the graphics processor will be called the "client."

In this model, the user will start the server with the command

`rspace -d x`,

and the server will then start up the appropriate client and initialize a communications stream. The server will be the same program regardless of the graphics output device which is currently in use—only the client will have to be re-written to port RSpace to a new graphical platform. Indeed, several clients can co-exist on a system which has several supported graphics devices. Command line options can be used to select which client will be used for the current session.

The use of two separate binaries for the server and the client allows the user more flexibility in the use of RSpace. For example, when the user is running on a fast machine with graphics capability, the server can merely `fork(2)` the client and run the output on the local display. Alternatively, the server can be executed on one machine and the client can be executed on another machine across a network. There will be some network delays caused by the transfer of data from the server to the client, but all 3D rotations can be done rapidly by the client on the machine which is being used for display. Since the X Window System provides the ability to run the client on one machine and display output on another machine, RSpace can actually be executed on three machines simultaneously. This arrangement is not recommended, however, because of the high bandwidth of data which would be transmitted from the client to the X Window Server on the display machine.

8.3 Implementation

At the time of this writing, there is only one client program: `xclient`, which provides graphic support for the X Window System. When RSpace is started, it will, by default, use `fork(2)` to start `xclient` and will initialize socket-based communications. Options are provided to specify another machine on which to run `xclient`, in which case the RSpace server will `fork(2)` an `rsh` process to the target machine and then execute `xclient` on that other machine. The requires that a copy of `xclient` executable from some point in the user's default `PATH` on the target machine. The `xclient` binary is the only file that need be present on the other machine, as all file I/O will be done by RSpace on the server side of the connection.

RSpace sends graphical information to `xclient`—this information is identical to that which was previously sent to the PS-300. The `xclient` manages all interactive rotations internally, without communicating with the "server." There are only a few specific events which will cause `xclient` to send information to the RSpace server:

1. In general, the client will support a "ping" operation which will acknowledge a synchronization request from the server. Basically, the server says "ping ?" and the client responds "ready."

2. When omega, chi, or phi scans are performed, the server tells the client which type of scan is being performed, and then waits for a response.

The response is either a number indicating that the scan is accepted for the specified number of degrees, or a special flag which indicates that the scan was aborted.[17]

3. When the "quit" button is pressed in `xclient`, the `xclient` will die, and the server is expected to notice the death of its child and terminate appropriately.

On a machine supporting the X Window System, the RSPACE server will run in one window and will manage all text-based interaction with the user through this window. The familiar `RSpace <> :` prompt will appear in this window. `xclient` will manage another window, with buttons, dials, and a graphics box. The buttons and dials perform functions which are similar to the functions of the buttons and dials on the PS-300 console.

`xclient` was implemented using ANSI C and the X Window System Toolkit (Xt) on a DECstation 5000/120 running ULTRIX V4.2. Widgets from the Athena widget set were used. The Dial widget is an extensive modification of Douglas A. Young's Dial widget. The base code for this widget contains the following copyright notice:

> Copyright 1989 by Douglas A. Young. All Rights Reserved.
>
> Permission to use, copy, modify, and distribute this software for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. The author disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness.
>
> Comments and additions may be sent the author at: dayoung@hplabs.hp.com

Complete code for this widget is included in the RSPACE distribution.

---

[17]The ability of the user to type in a value for the scan is no longer supported as a means of terminating a scan. The user must terminate the scan and then issue another command.

8.4 Hardware Platforms Tested

RSpace has been compiled and tested briefly on the following machines. In most cases, the native compiler (cc) and the GNU C Compiler were both tested. (At the time of this writing, the most recent version of gcc is 2.2.2.)

- DEC DECstation, running ULTRIX 4.x, with cc and gcc
- SUN SPARCstatcion, running SunOS 4.1.x, with cc and gcc
- SGI, running IRIX 4.0.x System V, with cc

8.5 Near Future Plans

Plans for the immediate future include porting the current RSPACE X Window System implementation to the following platforms:

- HP 9000/750, running HP-UX, with cc
- IBM RS-6000, running AIX, with cc and gcc
- Evans & Sutherland

For most of these platforms, the port should be relatively straight-forward, involving only differences in the C compilers and the UNIX versions. The initial ports are *not* expected to take advantage of any special graphics hardware which may be available—the ports will use only standard X Window features.

8.6 Distant Future Plans

In the distant future, some of the version 2.0 bug fixes from the Purdue version will be incorporated into RSPACE version 3. Further, additional clients will be written which can take advantage of the native 3D graphics ability the SGI machines (i.e., a client that uses gl instead of X). Other clients may be written for other graphics libraries.