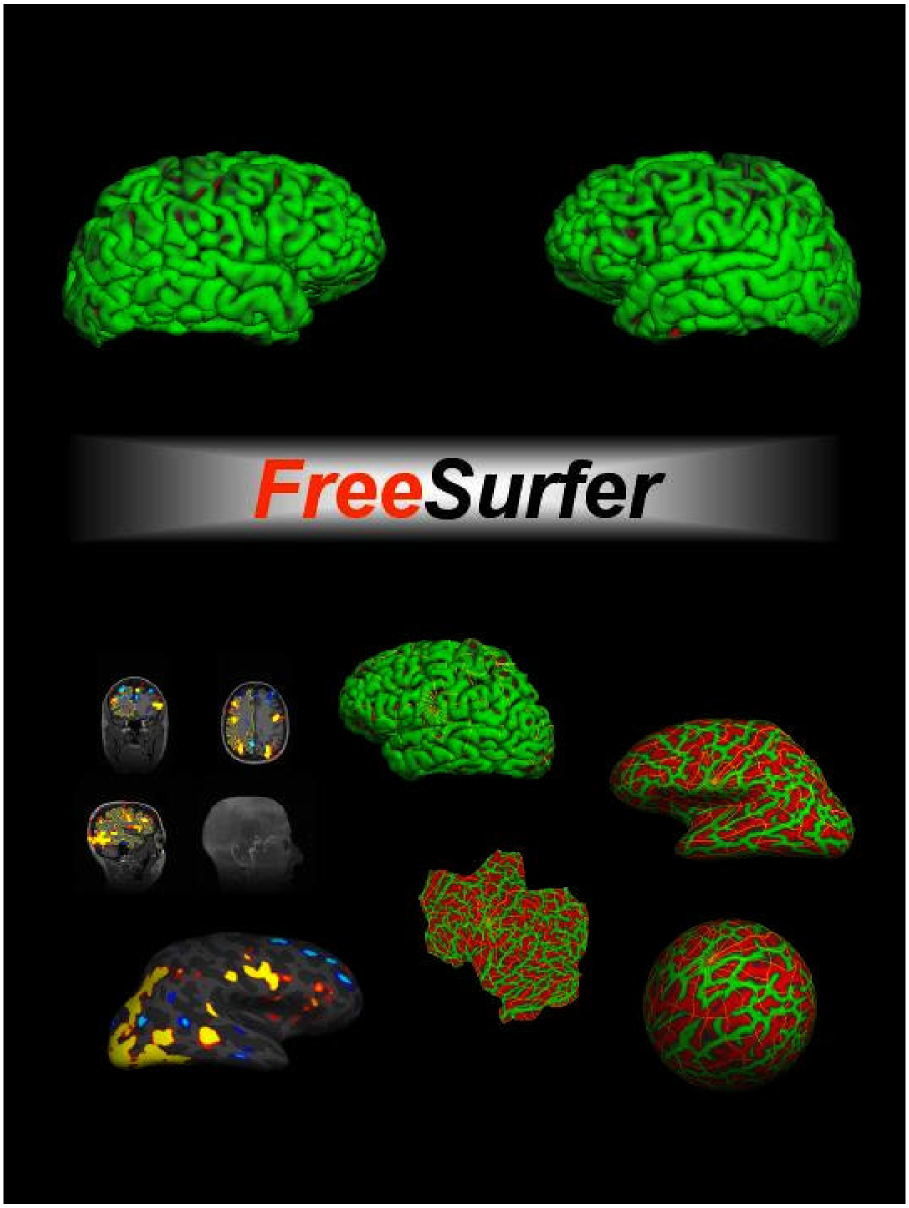


*Free**Surfer*** Manual
Version 0.8

FreeSurfer is a set of semi-automated tools for reconstruction of the brain's cortical surface and overlay of functional data onto the reconstructed surface.

csurf is a graphical interface to those reconstruction and functional overlay tools.

The general steps in the reconstruction process are:

Conversion of the MRI data into 256 coronal slices with 256 x 256 in-plane voxels (i.e. 1 mm x 1 mm x 1 mm voxels)

Intensity normalization of the MRI volume to remove variations in intensity due to magnetic susceptibility artifacts and RF-field inhomogeneities

Removal of extrameningeal tissues.

Segmentation of white matter with minimal manual editing to remove any topological defects (a fully-automated version is currently being tested)

Tessellation, smoothing and inflation for each cortical hemisphere

Cutting of the cortical surface

Flattening of the cortical surface

Morphing of the cortical surface into a sphere

A more technical description of the reconstruction process can be found in Dale, Fischl and Sereno. Cortical Surface-Based Analysis: I. Segmentation and Surface Reconstruction. NeuroImage (9): pp. 179-194. 1999. A more technical description of the cortical surface inflation, flattening, and surface-based coordinate system can be found in Fischl, Sereno, and Dale. Cortical Surface-Based Analysis: II. Inflation, Flattening, and a Surface-Based Coordinate System. NeuroImage (9): pp. 195-207. 1999.

The general steps in the functional overlay process are:

Conversion of the functional data into bshorts

Sampling of the statistical volume using the reconstructed surface (we refer to as "painting") which associates a statistical value with each vertex in the cortical surface

Rendering of the cortical surface that has been "painted" with the statistics.

Additional controls are provided to vary the visual appearance or the rendering

The statistical volume can also be viewed overlaid in the high resolution MRI volume.

This manual can be found on the Web at:

http://www.nmr.mgh.harvard.edu/freesurfer

http://www.cortechs.net/

This software is distributed by the Massachusetts General Hospital NMR Center and CorTechs©.

Anders Dale, Martin Sereno, Bruce Fischl, Sean Marrett, Douglas Greve, Thomas Witzel, Arthur Liu, Christian Hasselgrove, Eric Halgren

End-User Licensing Agreement ("EULA") for FreeSurfer

IMPORTANT- READ CAREFULLY: This End-User License Agreement ("EULA") is a legal agreement between you, the "licensee" (either an individual or a single entity), and CorTechs. For the purposes of this EULA, FreeSurfer includes computer software, documentation and sample datasets, the associated media, any printed materials, and any online or electronic documentation. By installing, copying, or otherwise using FreeSurfer, the licensee agrees to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, then you may not install, copy or otherwise use FreeSurfer, and you are required to destroy the CD and remove any files that you may have already installed.

1. GRANT OF LICENSE. This EULA grants the licensee the following rights:

Software. The licensee may install one copy of FreeSurfer on a single computer.

Network Use. The licensee may use FreeSurfer over an internal network, and the licensee may distribute FreeSurfer to other computers over an internal network.

Documentation. The licensee may make a copy of the documentation for internal use only.

2. LIMITATIONS

Academic Version. FreeSurfer can only be used by Colleges, Universities, and other Non-Profit Research Organizations. For-profit organizations and companies (including hospitals) are explicitly prohibited from using this software.

Rental. The licensee may not rent or lease FreeSurfer.

Commercial Use. The licensee may not charge anyone for any activity that uses FreeSurfer. For example, the licensee may not charge for cortical surface reconstruction using FreeSurfer. The licensee may not incorporate FreeSurfer or any of its parts into any commercial code or product of any kind.

Software Transfer. The licensee may not transfer FreeSurfer to any third party.

Termination. Without prejudice to any other rights, CorTechs may terminate this EULA if the licensee fails to comply with the terms and conditions of this EULA. In such an event, the licensee must destroy all copies of FreeSurfer and all of its component parts.

3. COPYRIGHT and TRADEMARKS.

FreeSurfer. All title and copyrights in and to FreeSurfer (including but not limited to any code, documentation, images, text or data) are owned by CorTechs. FreeSurfer is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. FreeSurfer is licensed, not given away or sold.

Trademarks. This License does not grant any rights to use the trade names "FreeSurfer", "CorTechs", "CSurf," or any other trademarks or trade names belonging to CorTechs unless explicitly permitted in writing by CorTechs.

Tck/Tk. Parts of this software are based on the Tcl/Tk software copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The original license terms of the Tcl/Tk software distribution is included in the file docs/license.tcltk. (Please see Appendix 1)

Tix. Parts of this software are based on the Tix software copyrighted by the Ioi Kim Lam, and Expert Interface Technologies. The original license terms of the Tix software distribution is included in the file docs/license.tix. (Please see Appendix 2)

HTML. Parts of this software are based on the HTML Library software copyrighted by Sun Microsystems, Inc. The original license terms of the HTML Library software distribution is included in the file docs/license.html_lib. (Please see Appendix 2)

Mesa. Parts of this software link to the Mesa library software copyrighted by Brian Paul. This software is covered under the GNU LIBRARY GENERAL PUBLIC LICENSE, Version 2, which is included in the file docs/license.gnu. (Please see Appendix 3)

4. NO SUPPORT OR WARRANTY. FreeSurfer is provided with no support whatsoever. CorTechs may, at its own discretion, provide bug reports or upgrades at www.nmr.mgh.harvard.edu and/or www.cortechs.net. FreeSurfer may contain in whole or in part pre-release, untested, or not fully tested works. FreeSurfer may contain errors that could cause failures or loss of data, and may be incomplete or contain inaccuracies. LICENSEE expressly acknowledges and agrees that use of FreeSurfer, or any portion thereof, is at LICENSEE's sole and entire risk. FreeSurfer is provided "AS IS" and without warranty, upgrades or support of any kind. CorTechs EXPRESSLY DISCLAIMS ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY OR SATISFACTORY QUALITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. CORTECHS DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN FREESURFER WILL MEET LICENSEE'S REQUIREMENTS, OR THAT THE OPERATION OF FREESURFER WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN FREESURFER WILL BE CORRECTED. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY CORTECHS OR A CORTECHS AUTHORIZED REPRESENTATIVE SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. LICENSEE ACKNOWLEDGES THAT FREESURFER IS NOT INTENDED FOR CLINICAL USE AND SHOULD NOT BE USED FOR DIAGNOSIS, TREATMENT PLANNING, OR ANY OTHER CLINICAL PURPOSE.

6. INFRINGEMENT. If any portion of, or functionality implemented by, FreeSurfer becomes the subject of a claim of infringement, CorTechs may, at its option: (a) attempt to procure the rights necessary for CorTechs and LICENSEE to continue using the Affected

Code; (b) modify the Affected Code so that it is no longer infringing; or (c) suspend LICENSEE's rights to use, reproduce, modify, sublicense and distribute the Affected Code until a final determination of the claim is made by a court or governmental administrative agency of competent jurisdiction and CorTechs lifts the suspension as set forth below. Such suspension of rights will be effective immediately upon CorTechs's posting of an email notice to such effect to the LICENSEE. Upon such final determination being made, if CorTechs is legally able, without the payment of a fee or royalty, to resume use, reproduction, modification, sublicensing and distribution of the Affected Code, CorTechs will lift the suspension of rights to the Affected Code by posting of an email notice to such effect to the LICENSEE.

7. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES SHALL CORTECHS BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE OR LICENSEE'S USE OR INABILITY TO USE FREESURFER, OR ANY PORTION THEREOF, whether under a theory of contract, warranty, tort (including negligence), products liability or otherwise, even if CORTECHS has been advised of the possibility of such damages AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY. In no event shall CorTechs's total liability to LICENSEE for all damages under this License exceed the amount of fifty dollars ($50.00).

8. MISCELLANEOUS.

U.S. Government End Users. The Covered Code is a "commercial item" as defined in FAR 2.101. Government software and technical data rights in the Covered Code include only those rights customarily provided to the public as defined in this License. This customary commercial license in technical data and software is provided in accordance with FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for Department of Defense purchases, DFAR 252.227-7015 (Technical Data -- Commercial Items) and 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). Accordingly, all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

Waiver; Construction. Failure by CorTechs to enforce any provision of this License will not be deemed a waiver of future enforcement of that or any other provision. Any law or regulation which provides that the language of a contract shall be construed against the drafter will not apply to this License.

Dispute Resolution. Any litigation or other dispute resolution between LICENSEE and CorTechs relating to this License shall take place in the Commonwealth of Massachusetts, and LICENSEE and CorTechs hereby consent to the personal jurisdiction of, and venue in, the state and federal courts within that Commonwealth with respect to this License. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded.

Entire Agreement; Governing Law. This License constitutes the entire agreement between the parties with respect to the subject matter hereof. This License shall be governed by the laws of the United States and the Commonwealth of Massachusetts, except that body of Massachusetts law concerning conflicts of law.

Quebec. Where LICENSEEs are located in the province of Quebec, Canada, the following clause applies: The parties hereby confirm that they have requested that this License and all related documents be drafted in English. Les parties ont exigé que le présent contrat et tous les documents connexes soient rédigés en anglais.

Authority. The person submitting this registration warrants that he/she has the authority to bind to this Agreement the party which he/she represents.

Installation

FreeSurfer (Version 0.8) is being distributed with executables for both Linux/Intel and SGI/Irix. This is an alpha release and has not been extensively tested.

To install FreeSurfer

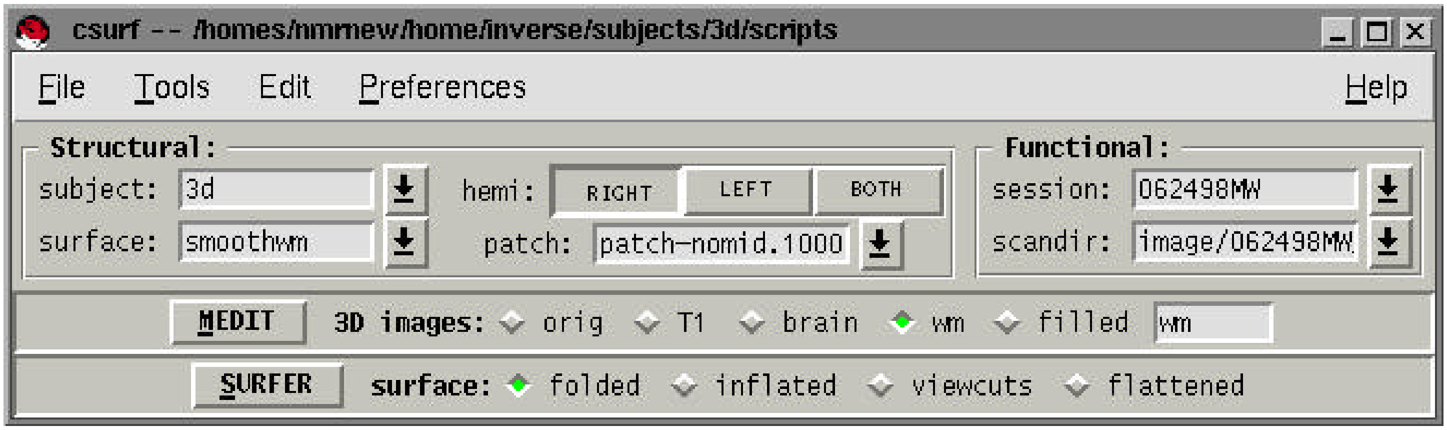
1) cp -rp <CD-ROM drive location> <destination directory>
2) cd <destination directory>
3) edit FreeSurferDefs.csh and change the following line
    setenv CSURF_DIR   XXXXXXXX   # CHANGE THIS DEFINITION
CSURF_DIR should be defined as the directory which contains the installation. It should be possible to run the installation as a demo directly from the CD-ROM.
4) Make sure that you are running csh or tcsh, and then source the FreeSurferDefs.csh file.
5) Type csurf at the prompt.

Note: you will need a password to run FreeSurfer. It can be obtained at **www.cortechs.net**.

Starting csurf

At the prompt, type:
csurf

This will start the **csurf** interface.



There are five pull down menus: **File**, **Tools**, **Edit**, **Preferences** and **Help**.

Below the pull down menus are two boxes labeled **Structural** and **Functional**.

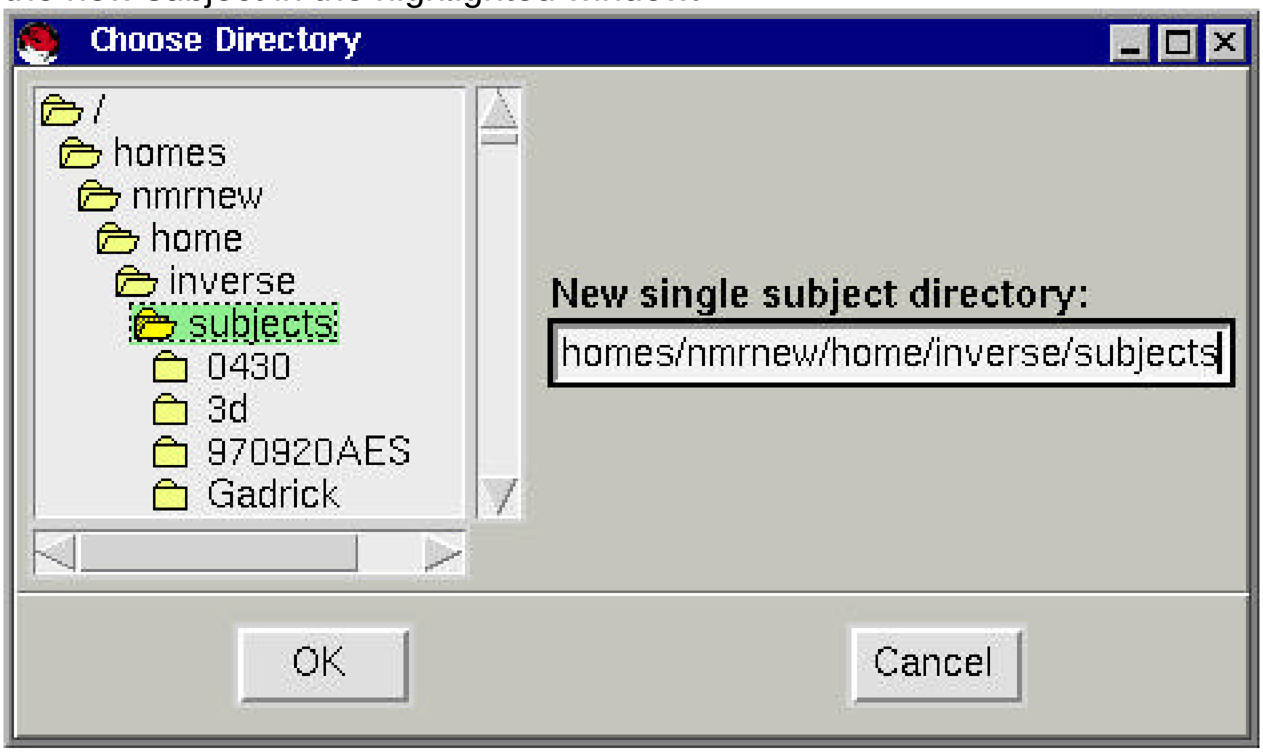
For the following examples, the subject to be created is named “test_subject.”

File Menu

The File pull down menu can also be accessed by pressing **Alt-F**.

New Subject

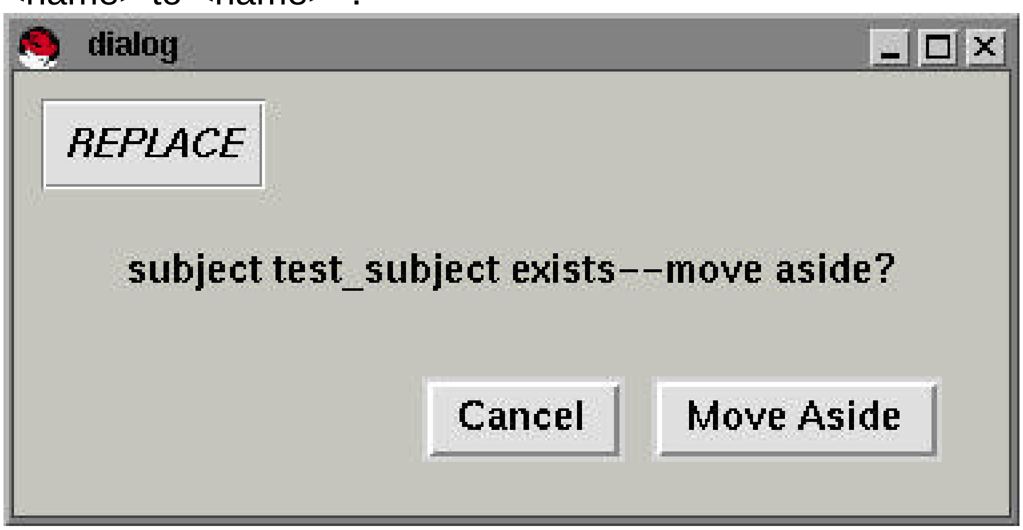
Creates the necessary directories for a subject's structural data. Specify the name of the new subject in the highlighted window.



If the subject does not exist, the following directories required for the structural data will be made:

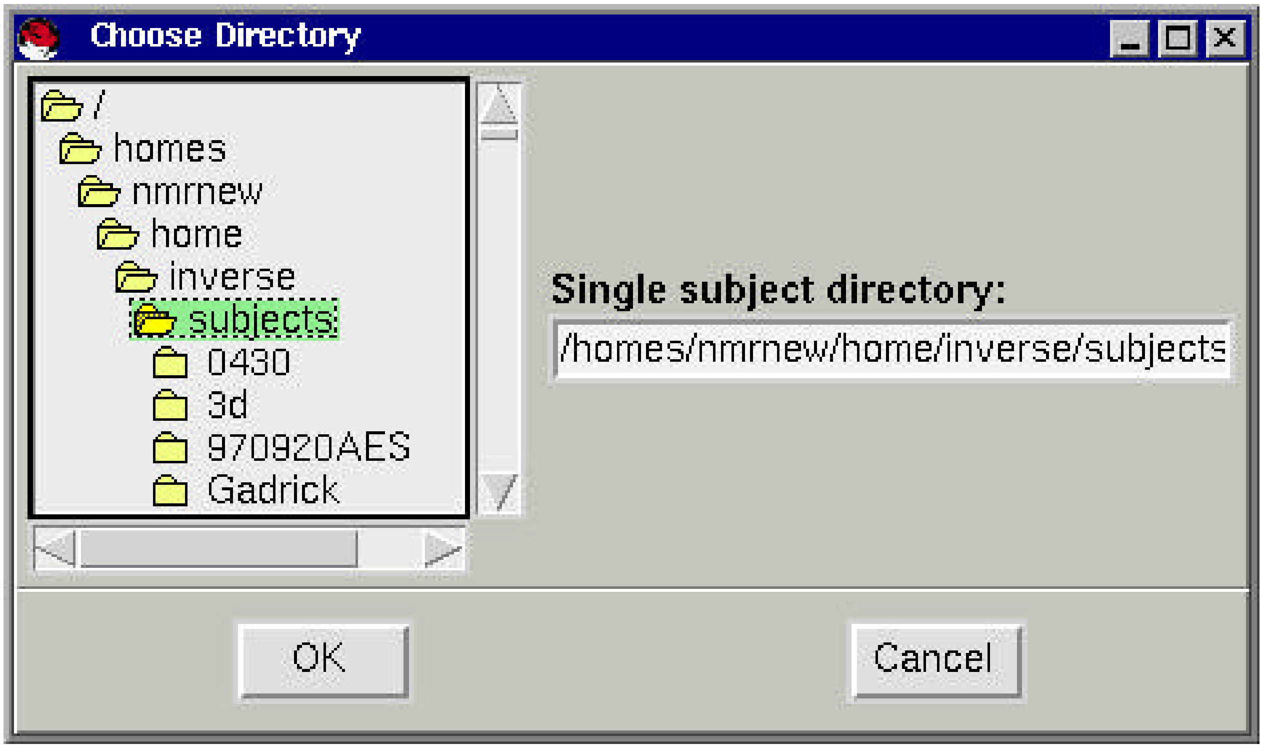
<name>
- bem
- label
- morph
- mpg
- mri
  - T1
  - brain
  - filled
  - orig
  - tmp
  - transforms
  - wm
- rgb
- scripts
- surf
- tiff
- tmp

If the subject does exist, you will be asked if you want to move the current subject from <name> to <name>~.



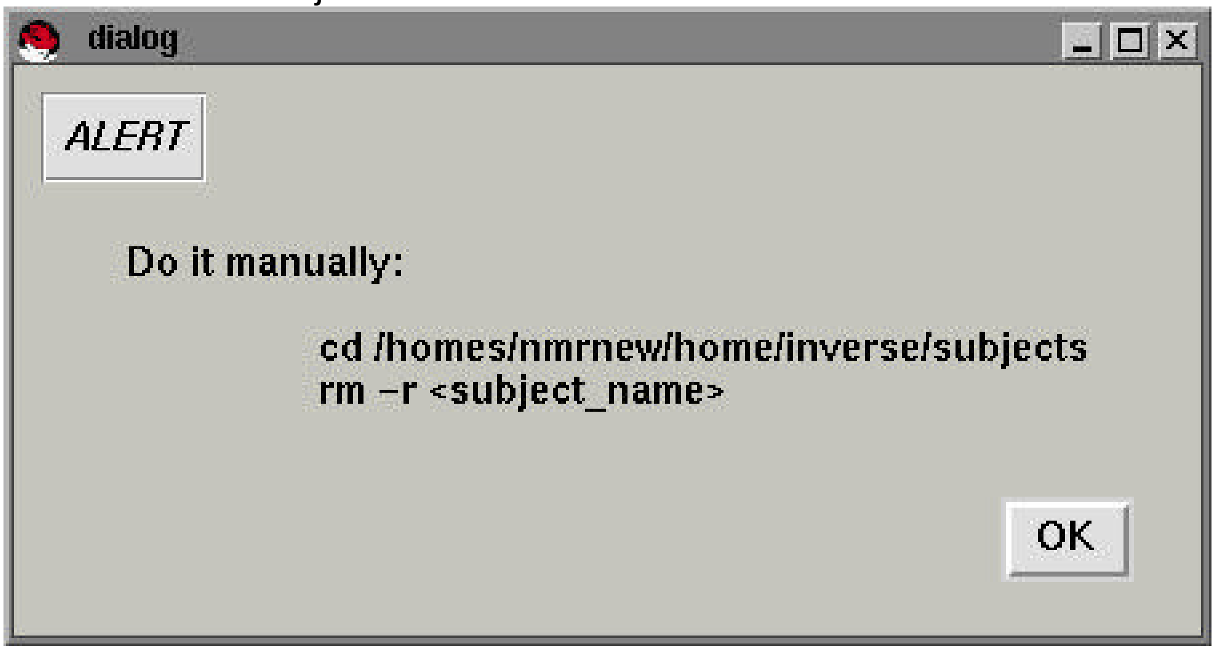
Open Subject

Opens an existing subject. Either select the subject directory on the left or enter the subject directly into the field on the right.



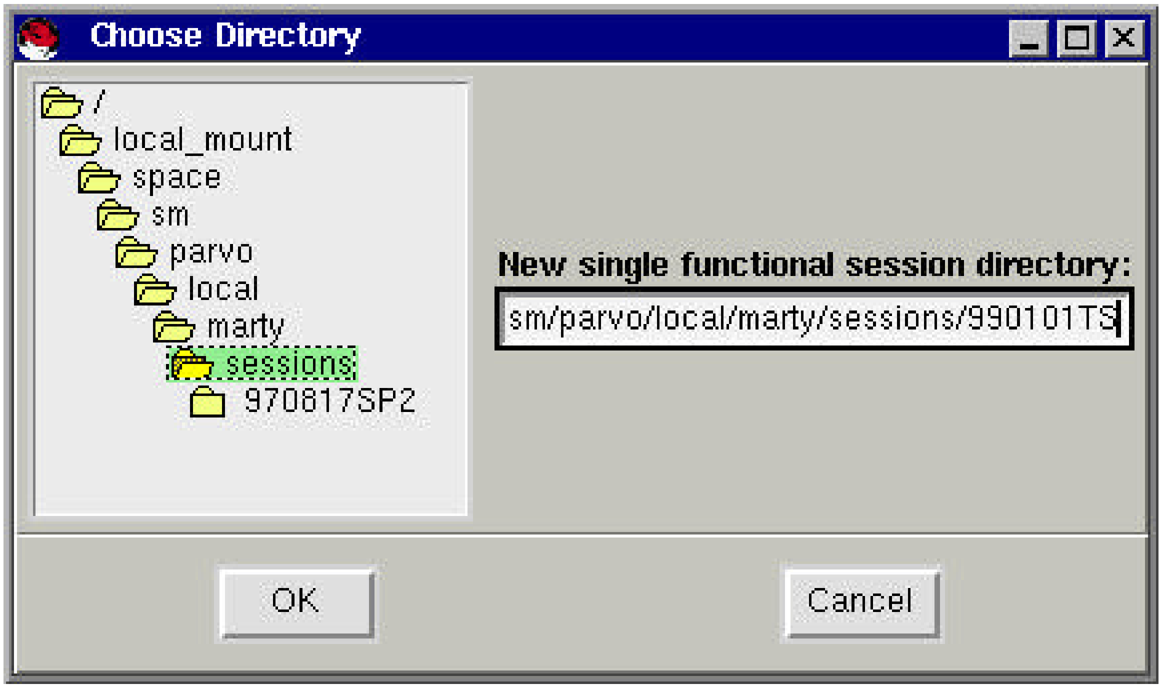
Delete Subject

Deletion of the subject is manual.

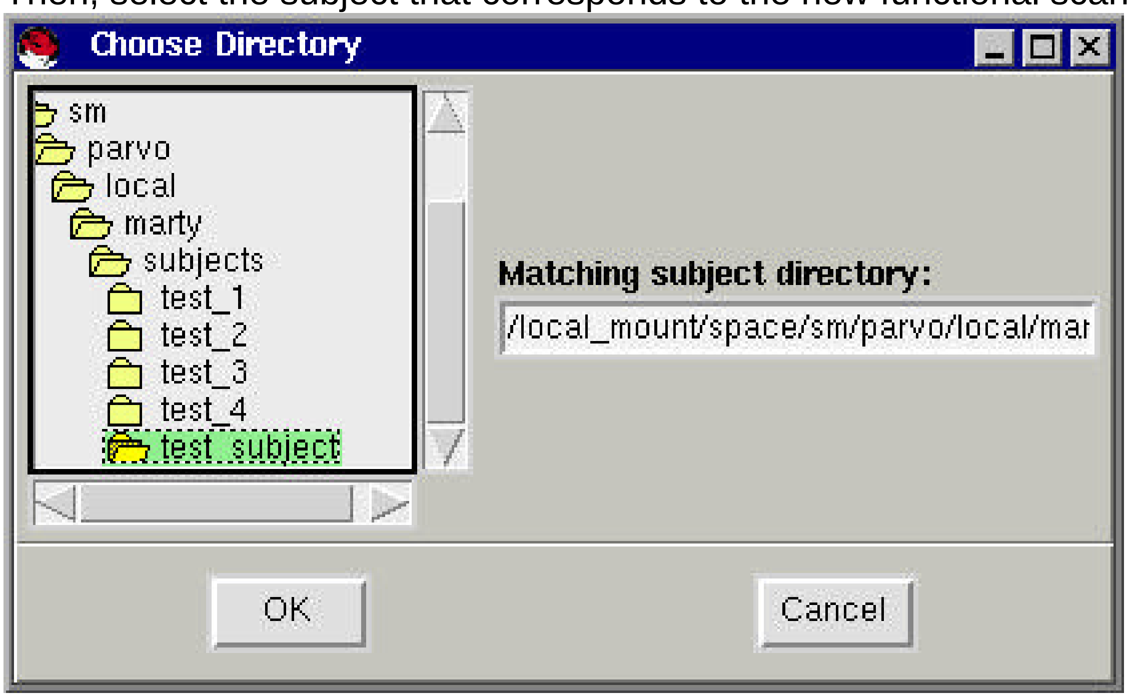


New Functional

Creates the necessary directories for a new functional scan. Specify the name of the new functional directory in the highlighted window. The suggested naming convention is <2year><2month><2day><2-3initial>.



Then, select the subject that corresponds to the new functional scan.

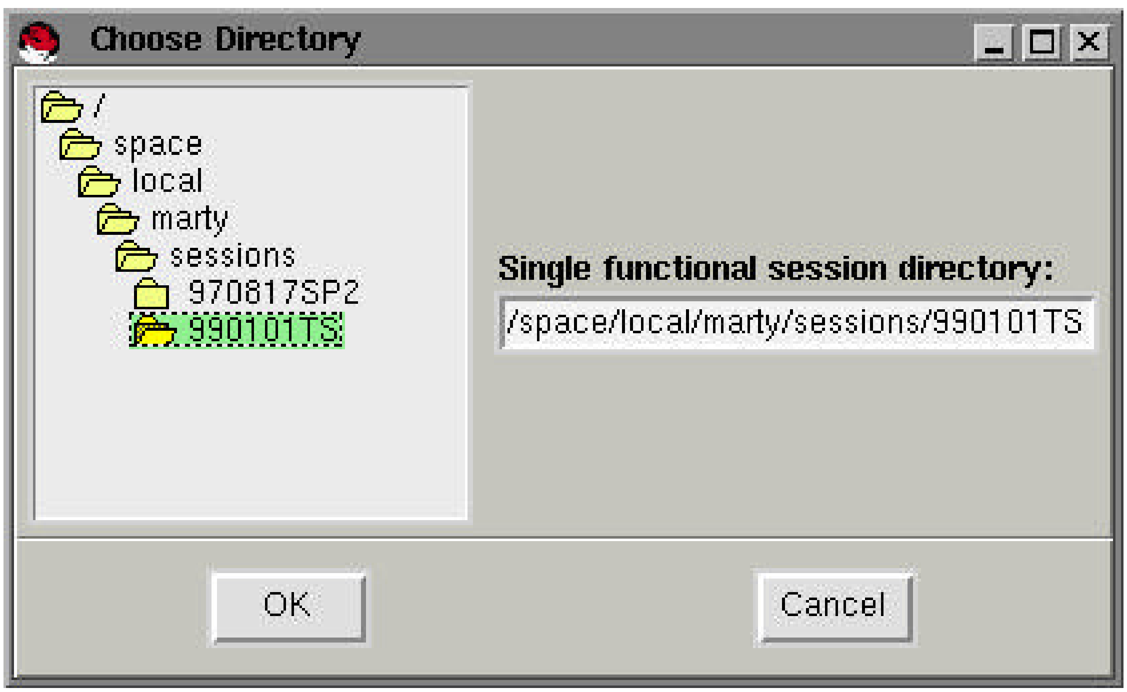


The following directories are created:

<functional scan>
  image
    mpg
    rgb
    scripts

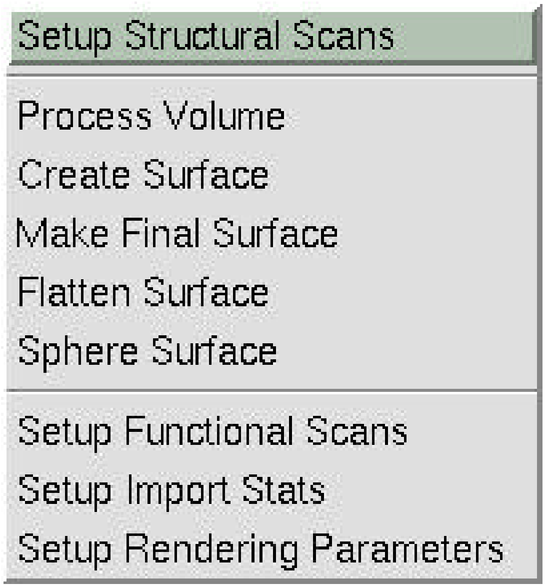
In addition the file “name” is created in the new functional directory and contains the name of the subject.

Open Functional
Opens an existing functional scan. Either select the functional directory on the left or enter the functional directory directly into the field on the right.



Quit
Quits the **csurf** interface.

Tools Menu – Structural Commands



The Tools pull down menu can also be accessed by pressing **Alt-T**.

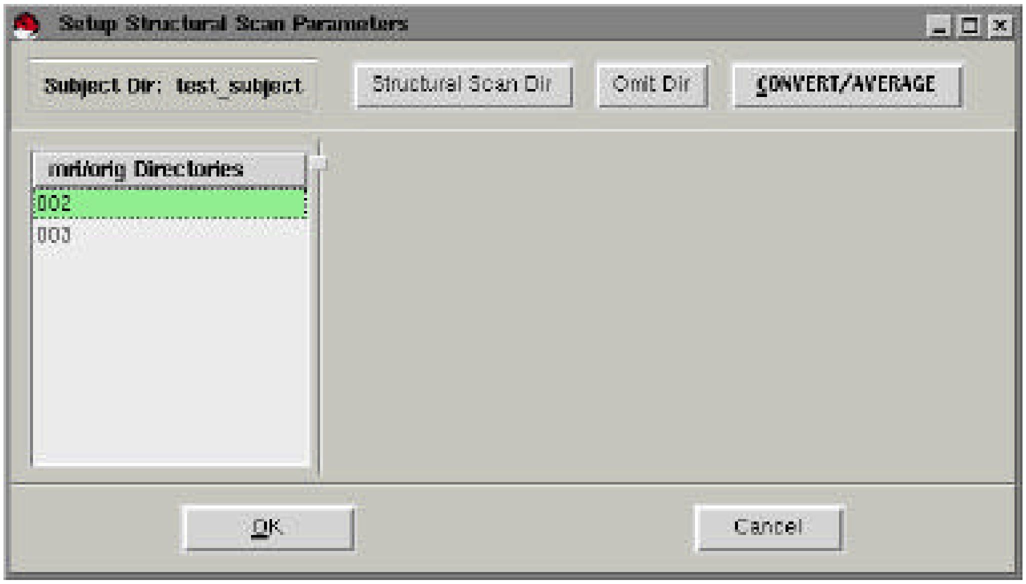
The first six commands on the **Tools** menu are used to

setup the structural MRI data directories
process the structural data (performed once)
create the cortical surface
inflate the cortical surface
flatten the cortical surface
morph the surface into a sphere

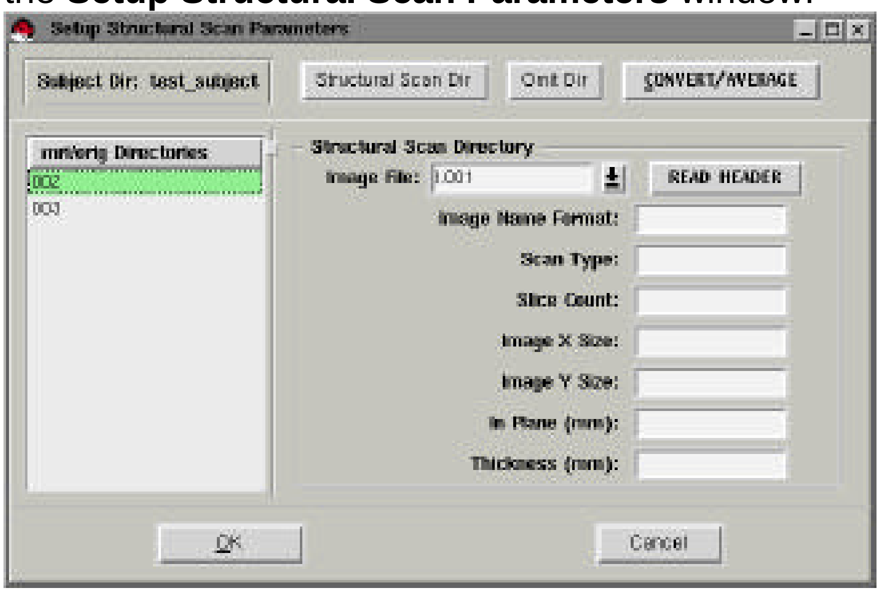
Setup Structural Scan(s)

Converts (and averages if there are multiple acquisitions) the MRI data into the 256 coronal slices, 256 x 256 voxels in plane with1 mm^3 resolution. Currently supported formats are: SPM/analyze, AFNI, Siemens, and GE.

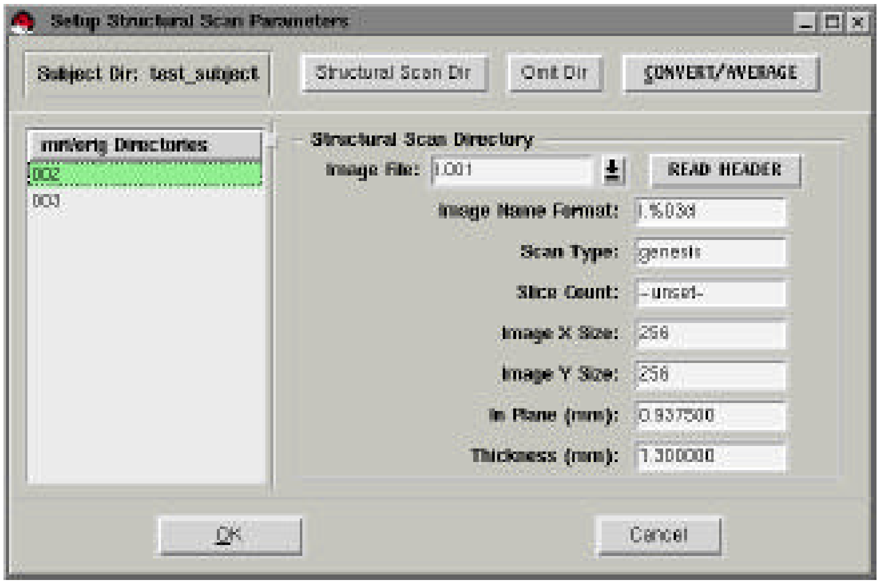
Each structural acquisition must be in its own directory in the **mri/orig** directory.



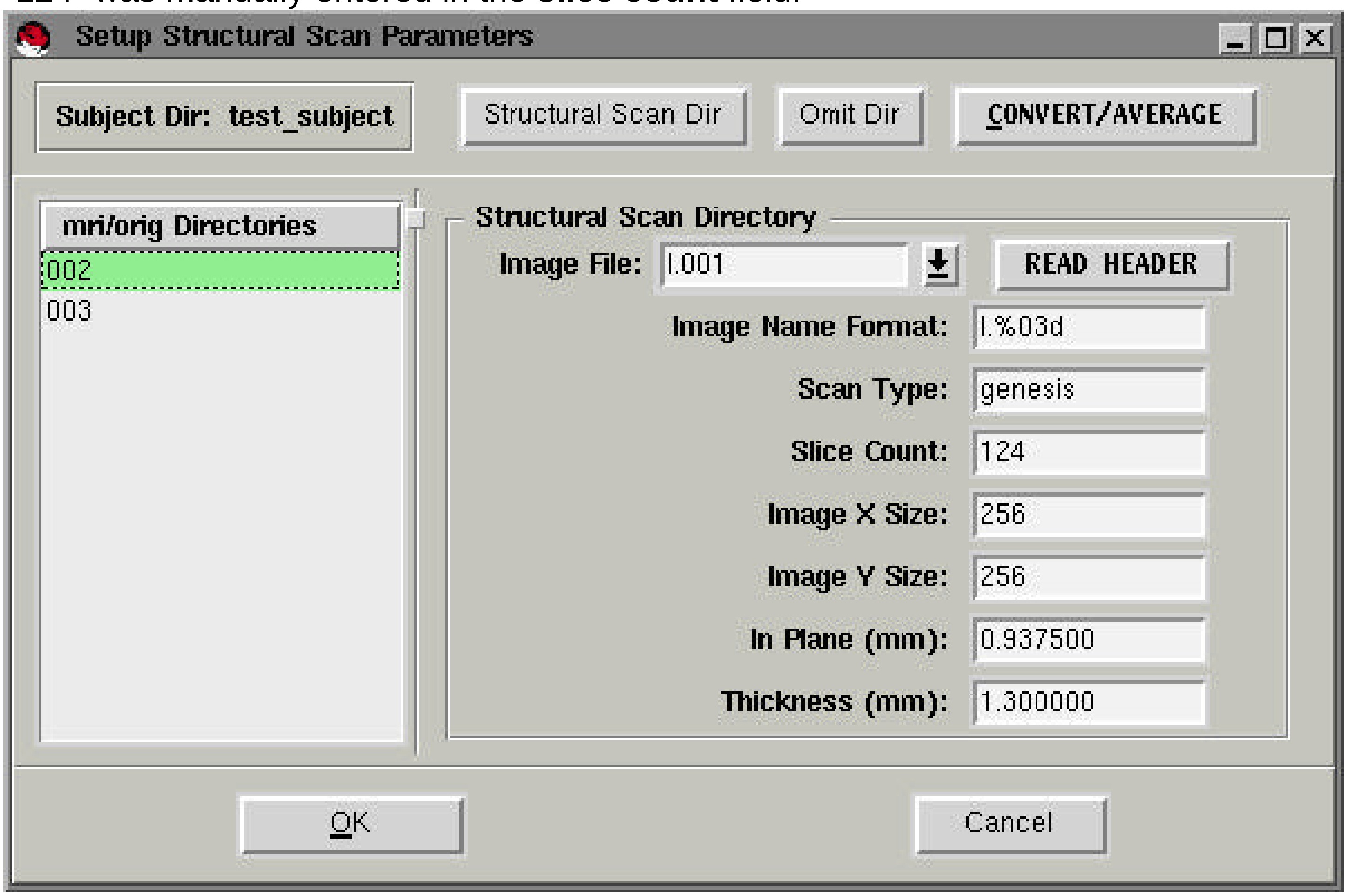
Select the first structural acquisition and press **Structural Scan Dir**. This will expand the **Setup Structural Scan Parameters** window:



Select the first image file in the **Image File** field and press **READ HEADER** to automatically determine the scan parameters. If there are any errors, manually enter them in the field.



In this particular example, the **slice count** was not determined by **READ HEADER**, so “124” was manually entered in the **slice count** field.



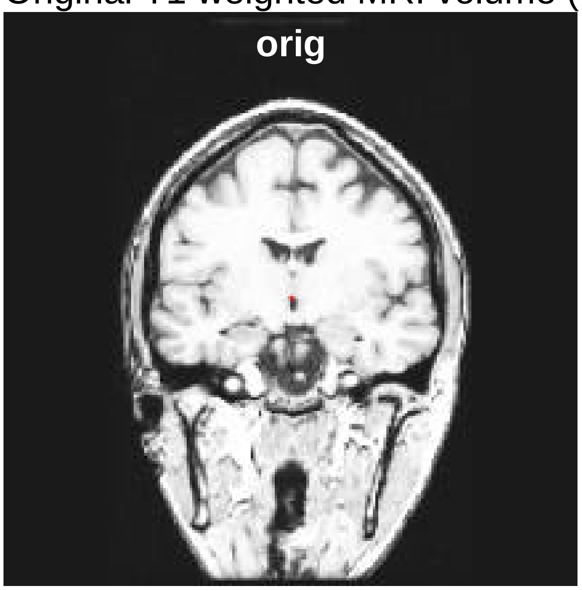
For each structural acquisition, repeat the process of selecting the acquisition on the left window, pressing **Structural Scan Dir**, and then pressing **READ HEADER**.

If you want to omit an acquisition that was previously selected, reselect the directory and press **Omit Dir**.

Press **Convert/Average** to average the acquisitions and convert the MRI volume in the 256 coronal slices.

The output files of the setup are:
    images: $SUBJECTS_DIR/$name/mri/orig/COR-???

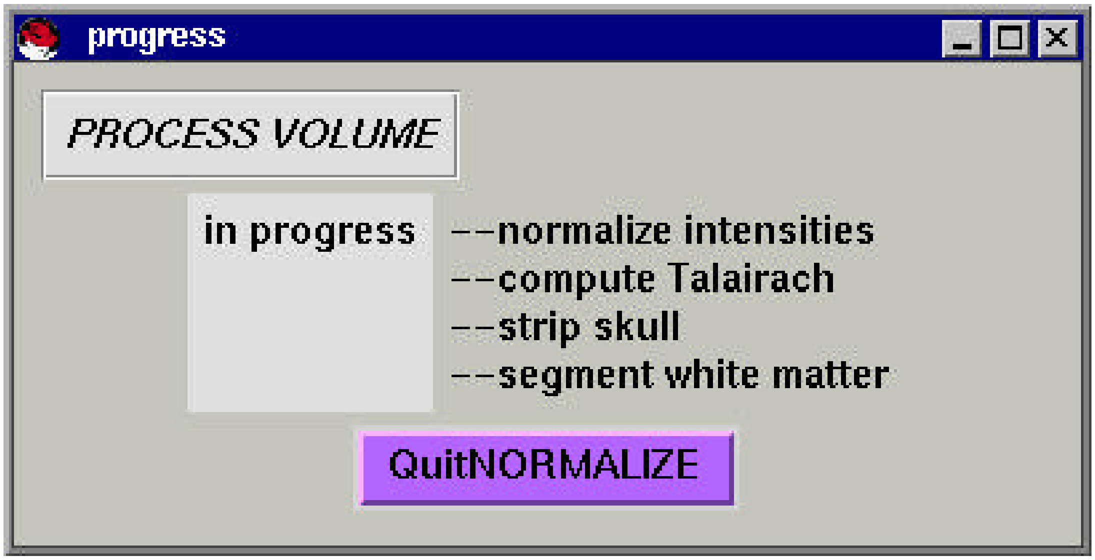
Original T1 weighted MRI volume (orig volume)



Process Volume

Starts a four-part background process to prepare the 3-D MRI data to make a surface. While the volume is being processed, the csurf interface can still be accessed. The process can be canceled by pressing the **Quit** button that is highlighted in purple.

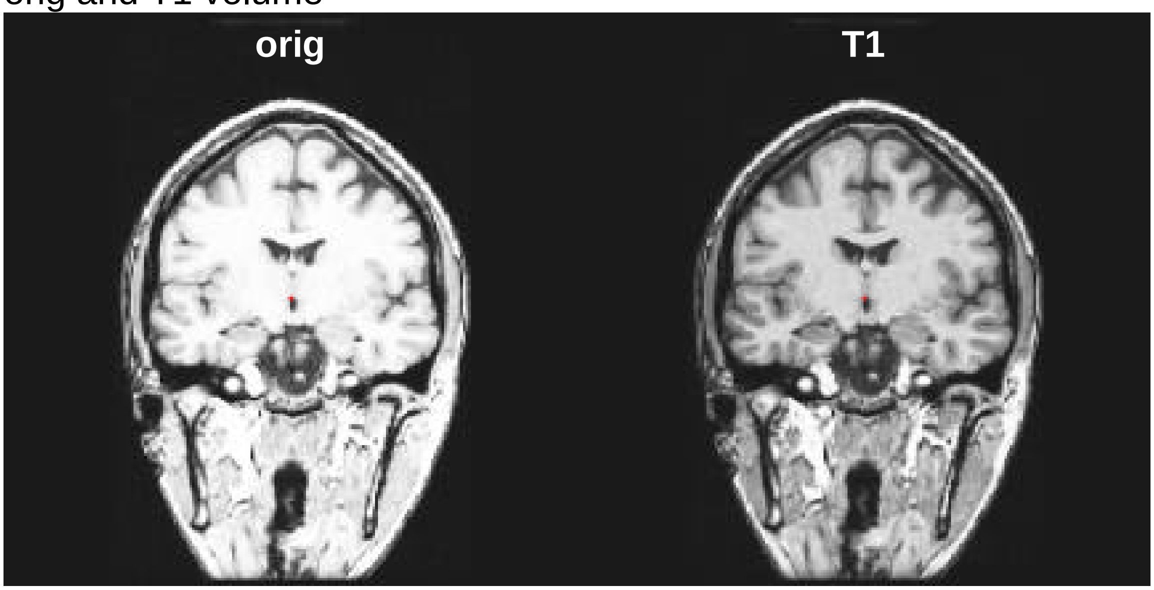
Part 1: Normalize Intensities



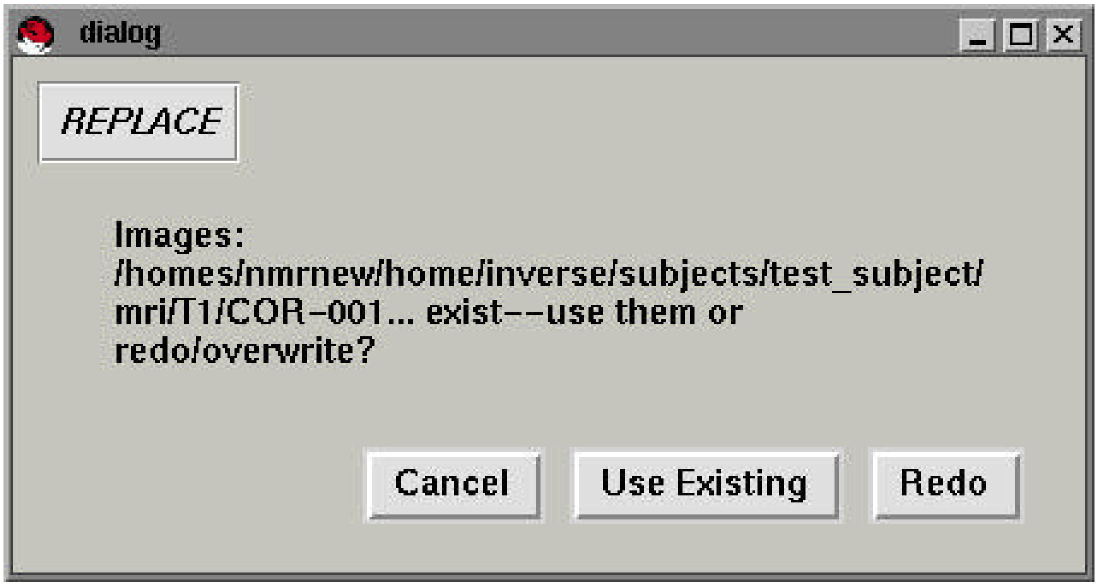
The intensity normalization procedure removes variations in intensity due to magnetic susceptibility artifacts and RF-field inhomogeneities

The output files written by the normalization procedure are:
images: $SUBJECTS_DIR/$name/mri/T1/COR-???

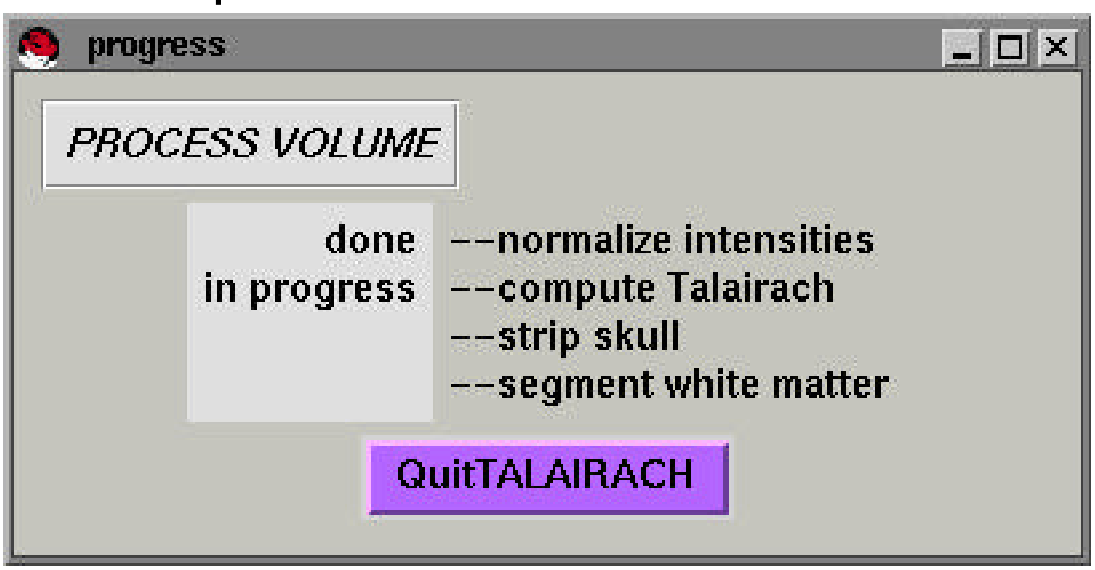
orig and T1 volume



If the T1 volume (T1/COR-??? files) already exists, you can either use the existing files (**Use Existing** button) and continue with the process, or redo the intensity normalization (**Redo** button). If **Don't Ask Overwrite** (under the **Preferences** menu) is checked, the intensity normalization is automatically redone.



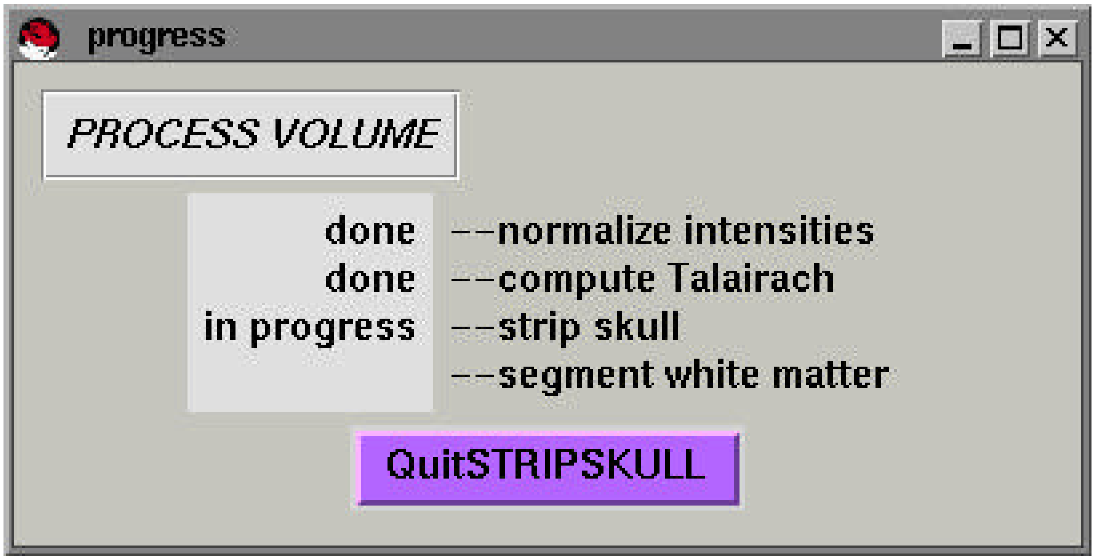
Part 2: Compute Talairach



Computation of the Talairach transformation matrix is currently not supported. If a transformation matrix in the Montreal Neurological Institute format (xfm) is available, Talairach coordinates will be displayed by **medit** and **surfer**. The transformation matrix should be in the following file:

$SUBJECTS_DIR/$name/mri/transforms/talairach.xfm

Part 3: Strip Skull

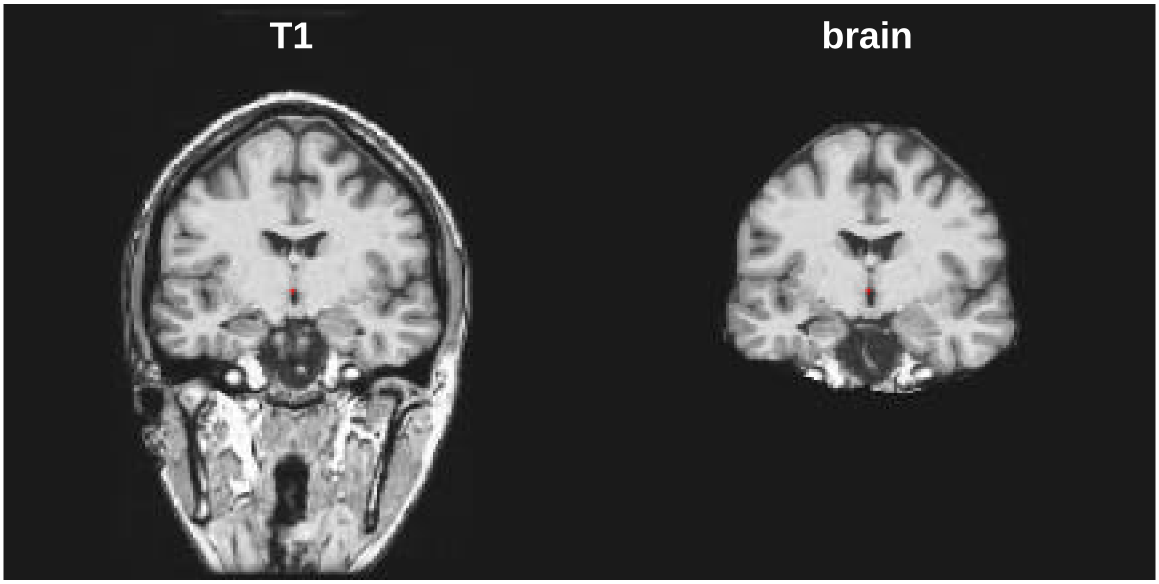


A supertessellated icosahedron is shrunk onto the normalized data set. At each point on the surface a 'core sample' is used to search for the black laminae that correspond to the inner and outer tables of bone in a T1-weighted scan, while at the same time, preserving neighbor relations. Once this surface is found, it is used to strip the skull off the T1 data set.

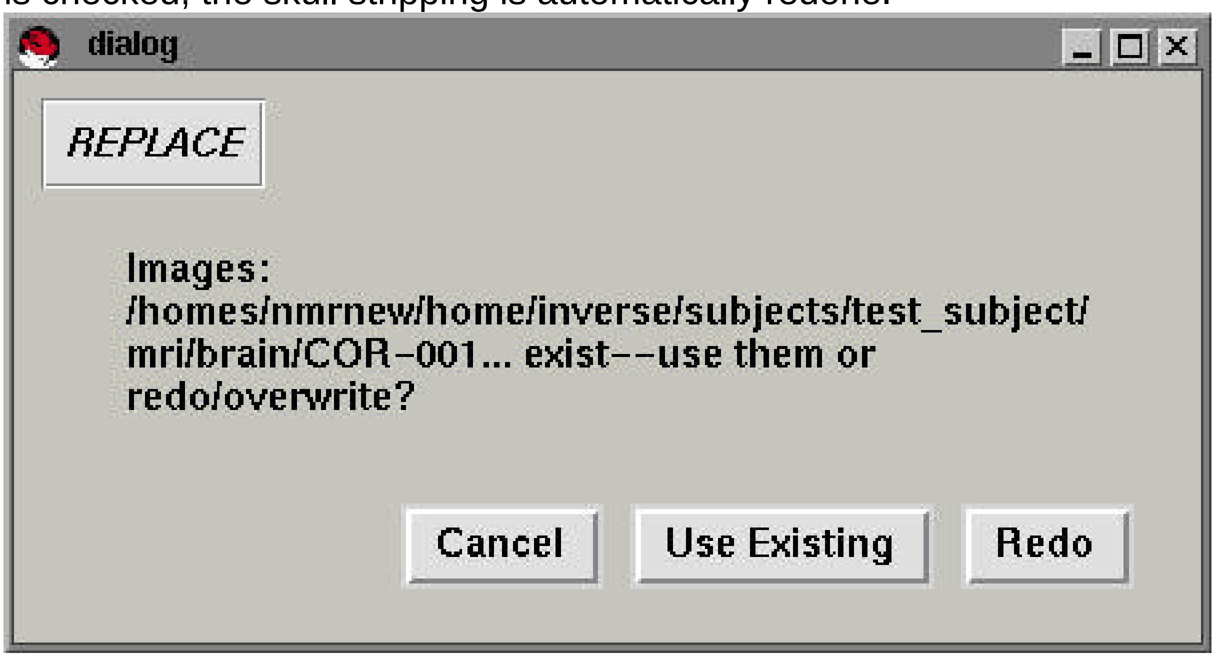
The output files written by this procedure are:

shrinksurface: $SUBJECTS_DIR/$name/bem/brain.tri
images: $SUBJECTS_DIR/$name/mri/brain/COR-???

T1 and brain volume

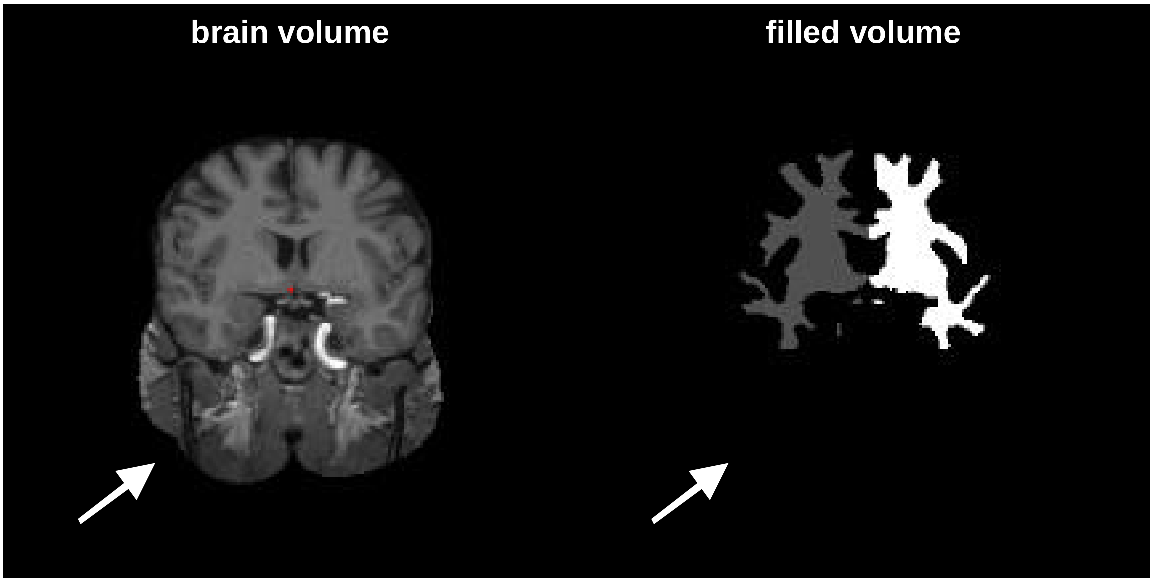


If the brain volume (brain/COR-??? files) already exists, you can either use the existing files (**Use Existing** button) and continue with the process, or redo the intensity normalization (**Redo** button). If **Don't Ask Overwrite** (under the **Preferences** menu) is checked, the skull stripping is automatically redone.

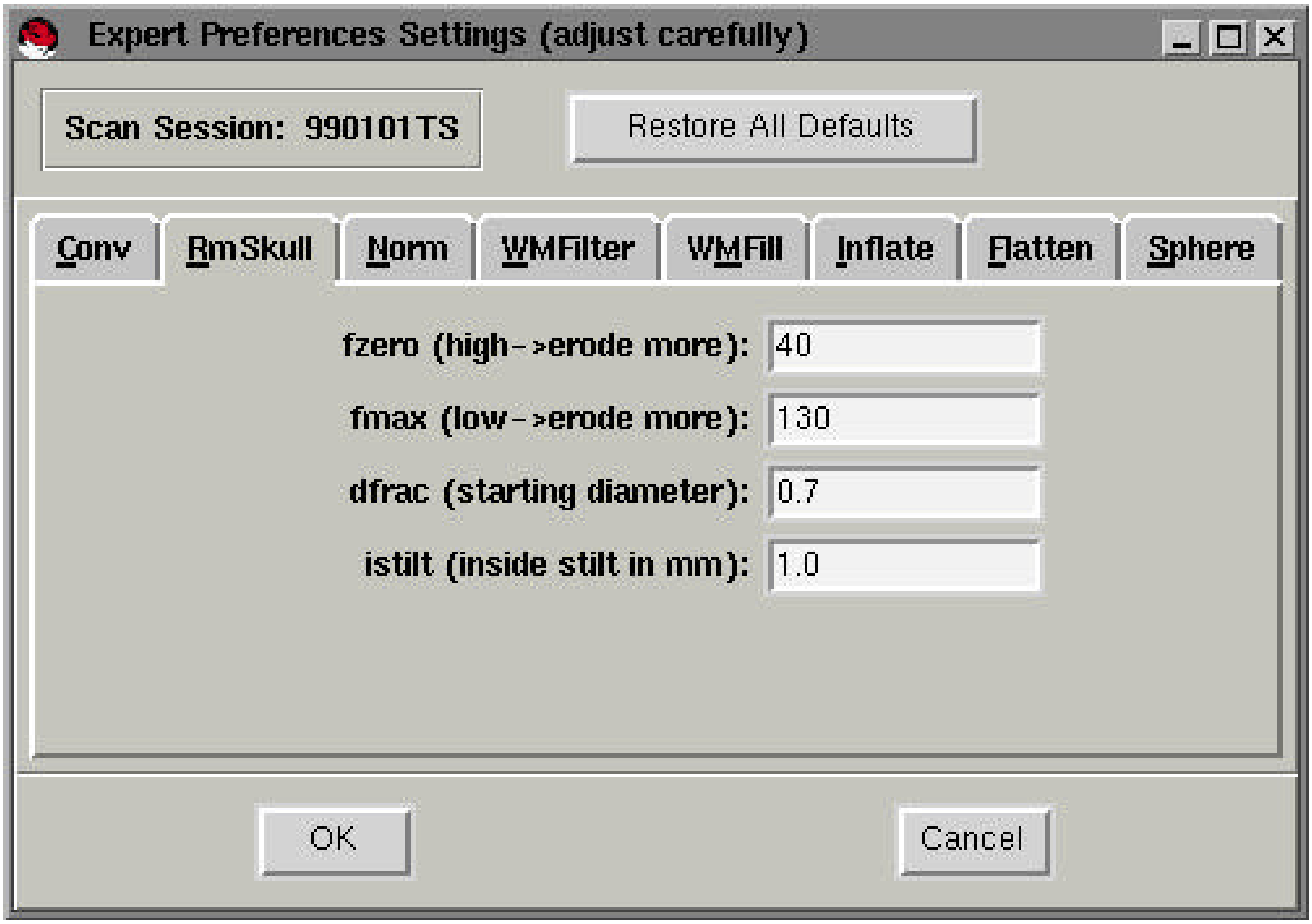


Failure of the skull stripping

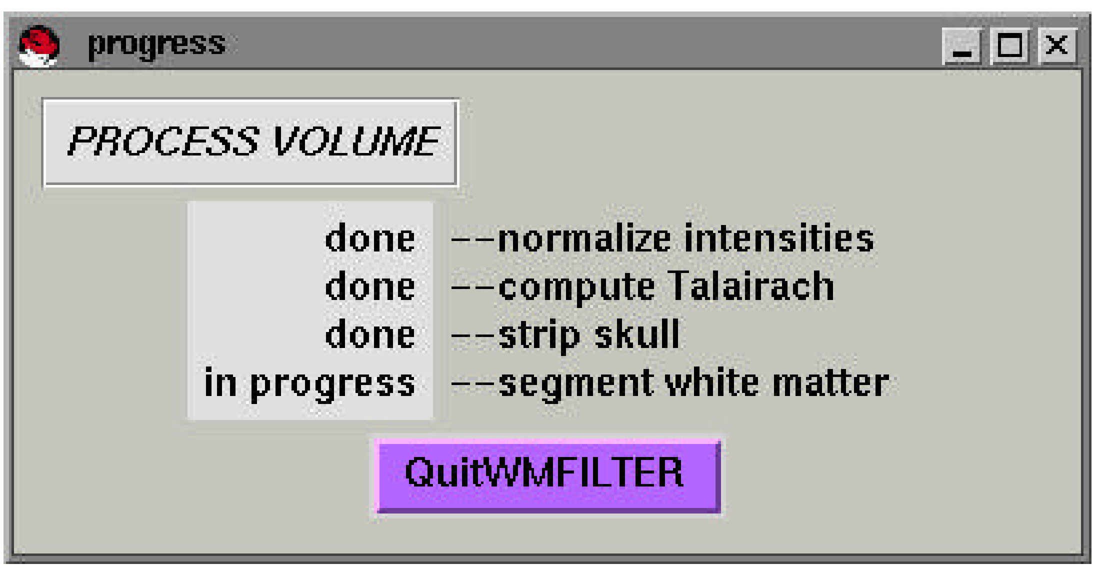
In some cases, the skull stripping either removes too little skull or removes some of the brain parenchyma. First, examine the **filled** volume. In some cases where the skull stripping has left a substantial amount of neck, the subsequent segmentation and filling has removed the extra neck.



If the filled volume has unwanted skull and neck structures, or actual brain has been removed, select **Expert Preferences** under the **Preferences** menu. Select the **Stripskull** tab. If there are extra structures (e.g. unwanted skull and neck) in the filled volume, increase the **fzero** parameter and decrease the **fmax** parameter in small increments (5 is good choice). It brain has been removed, decrease the **fzero** parameter and increase the **fmax** parameter in small increments (5 is good choice). Rerun **Process Volume** from the **Tools menu**. You do not have to redo the intensity normalization (i.e. select **Use Existing**).



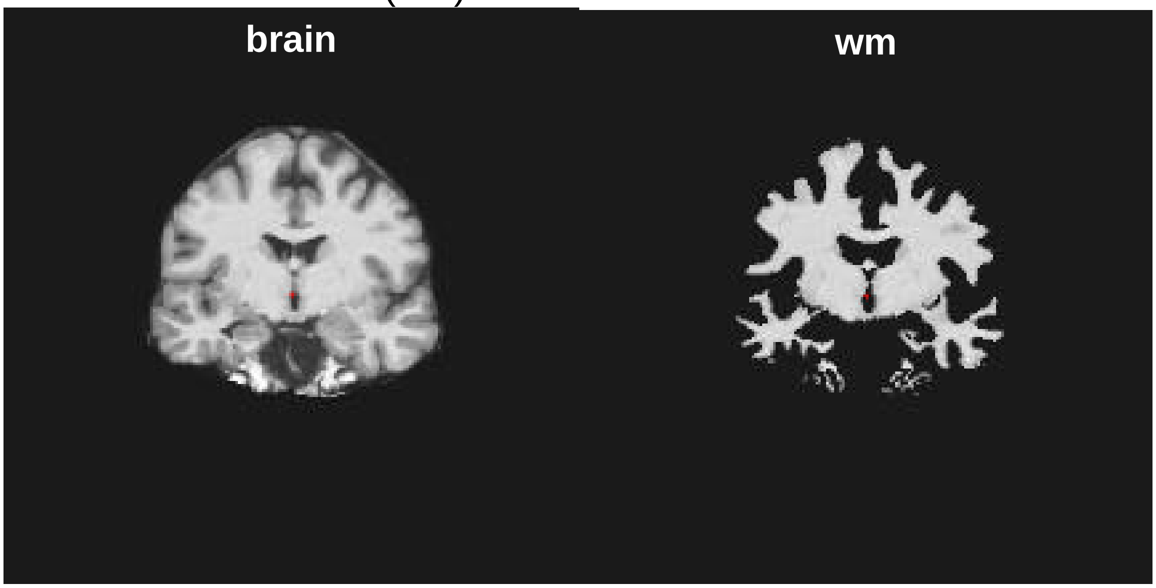
Part 4: White Matter Filter



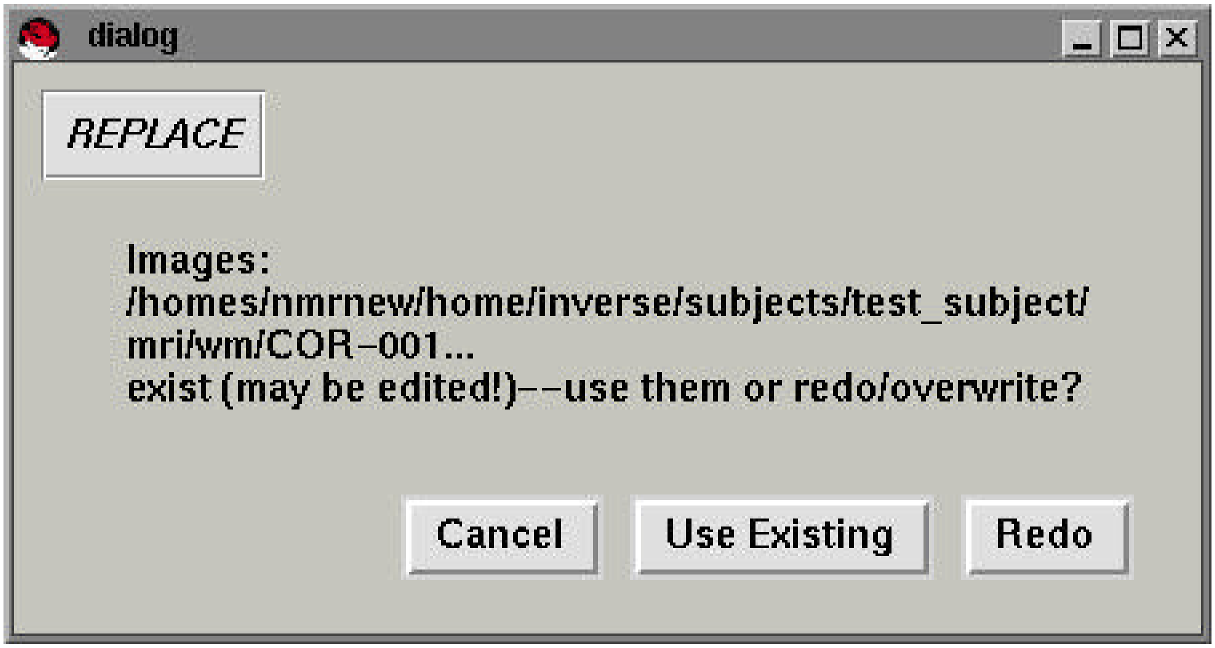
An oriented filter is applied to the stripped data set to 'floss' and 'spackle' small defects. Filter coefficients are computed for each 5x5x5 volume in order to determine which plane is most likely to be parallel to the cortical surface, and then a modified median filter is applied.

The output files written by this procedure are:
images: $SUBJECTS_DIR/$name/mri/wm/COR-???

brain and white matter (wm) volume



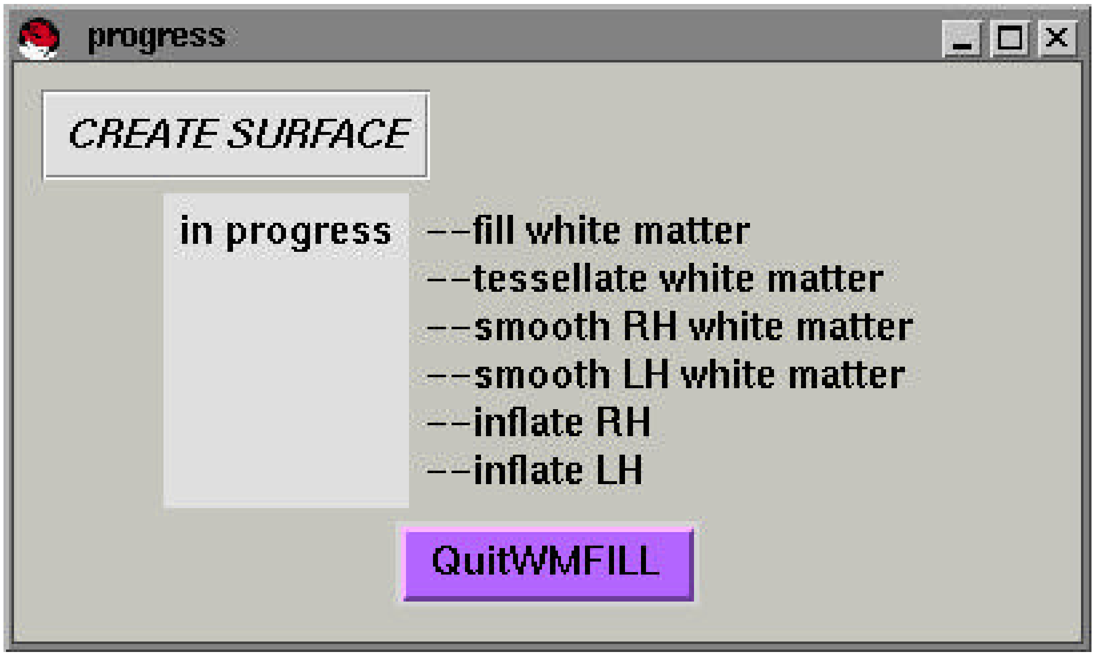
If the white matter volume (wm/COR-??? files) already exists, you can either use the existing files (**Use Existing** button) and continue with the process, or redo the intensity normalization (**Redo** button). If **Don't Ask Overwrite** (under the **Preferences** menu) is checked, the white matter segmentation is automatically redone.



Create Surface

Starts a six-part background process to create the left and right hemisphere cortical surfaces. If necessary, you can create the left or right hemispheres alone. While the surface is being generated, the csurf interface can still be accessed. The process can be canceled by pressing the **Quit** button that is highlighted in purple.

Part 1: Fill White Matter

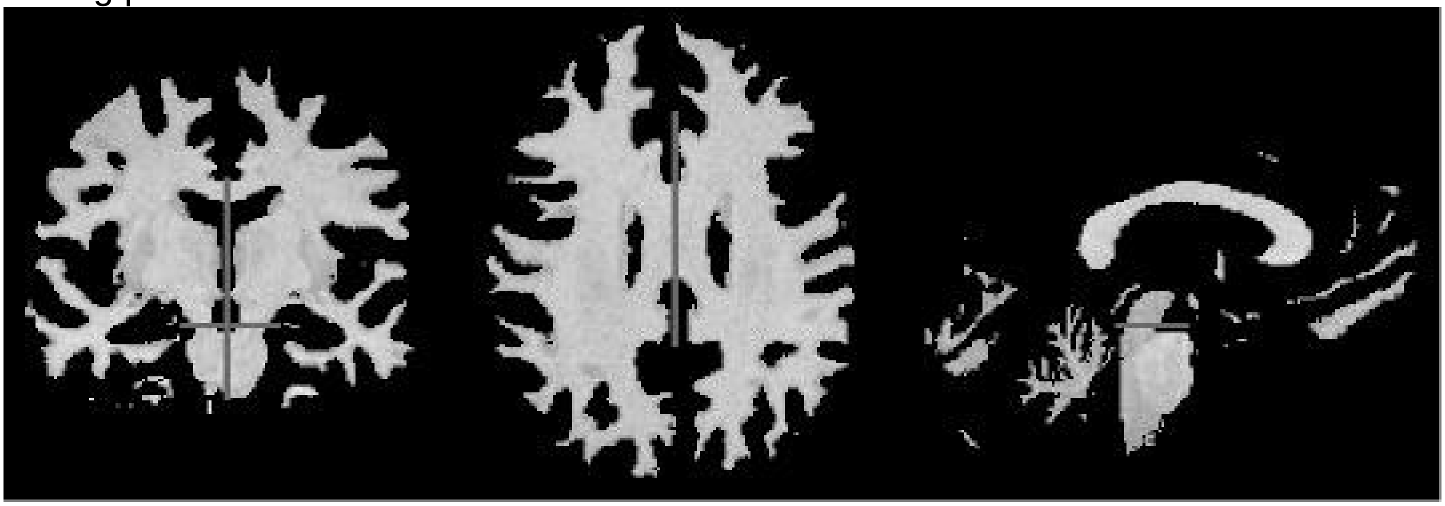


First, automatically finds the cutting planes to 1) separate the left and right hemispheres and 2) prevent the surface from going down into the brain stem.
Then, starts a 3-D region growing process from a starting point within the white matter in order to generate the starting point for a surface.
Right hemisphere voxels are assigned a value of 80 (gray). Left hemisphere voxels are assigned a value of 255 (white).

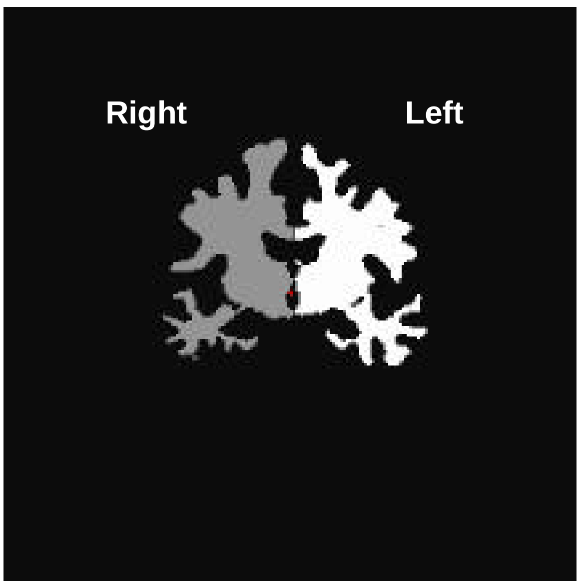
The output files written by this procedure are:

images: $SUBJECTS_DIR/$name/mri/filled/COR-???

cutting planes

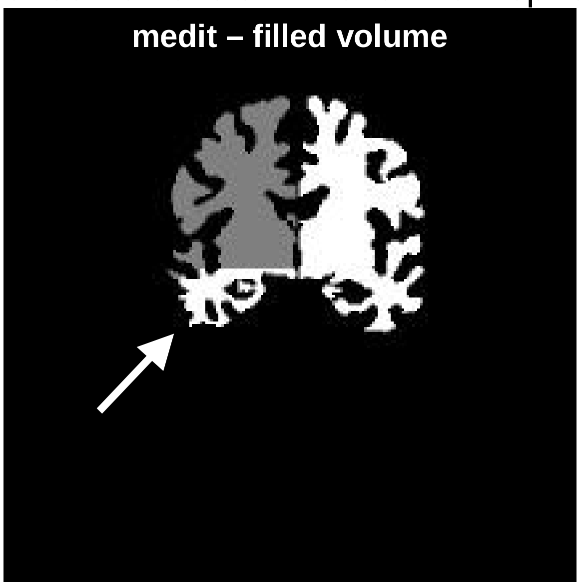


filled volume

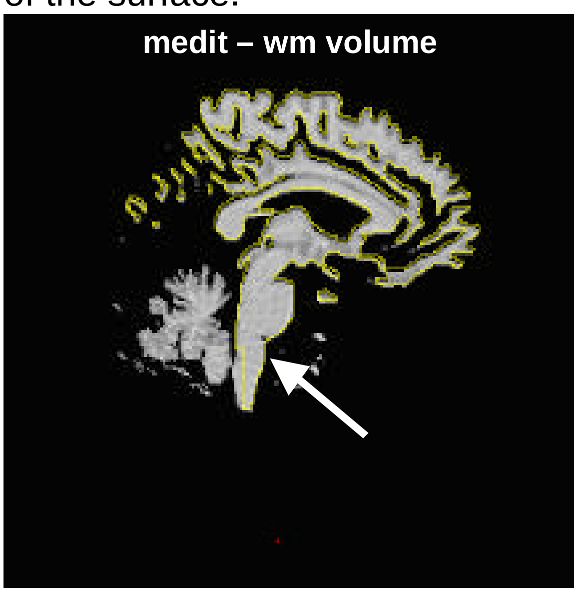


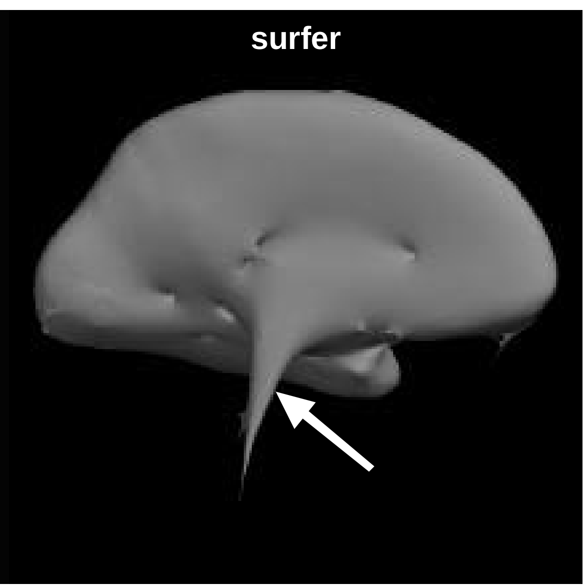
Failure of the automatic cutting planes:

In some cases, the cutting planes are not found correctly. This can be seen in **medit** as incorrectly colored voxels, e.g. some voxels in the right hemisphere are white (255) or some voxels in the left hemisphere are gray (80).



This can also be seen in **surfer** as a large protrusion in the center of the medial aspect of the surface.



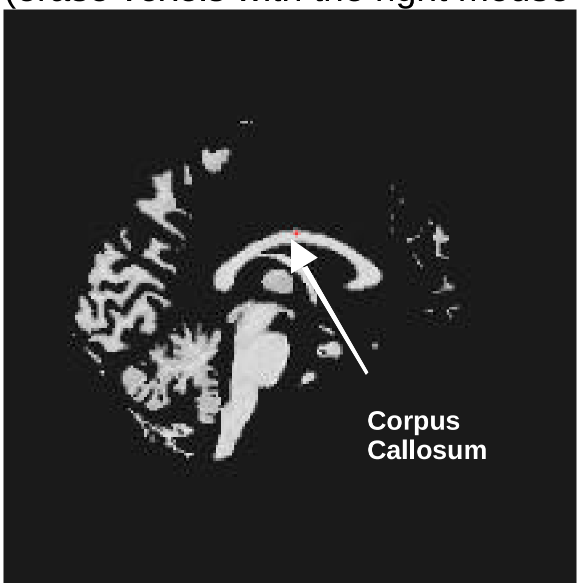


Manually defined cutting planes

The following instructions describe how to manually define the two cutting planes.

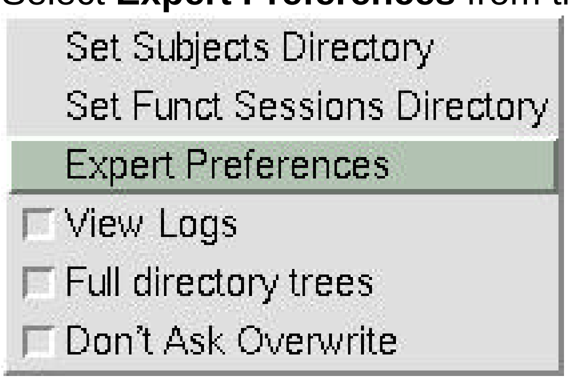
Left-Right Hemisphere separation (Corpus Callosum)

Using **medit**, find a sagittal slice in the **wm** volume in which the corpus callosum is disconnected. This slice should be near the mid-line. If you cannot find a slice with the corpus callosum disconnected, manually erase voxels to separate the corpus callosum (erase voxels with the right mouse button in medit).

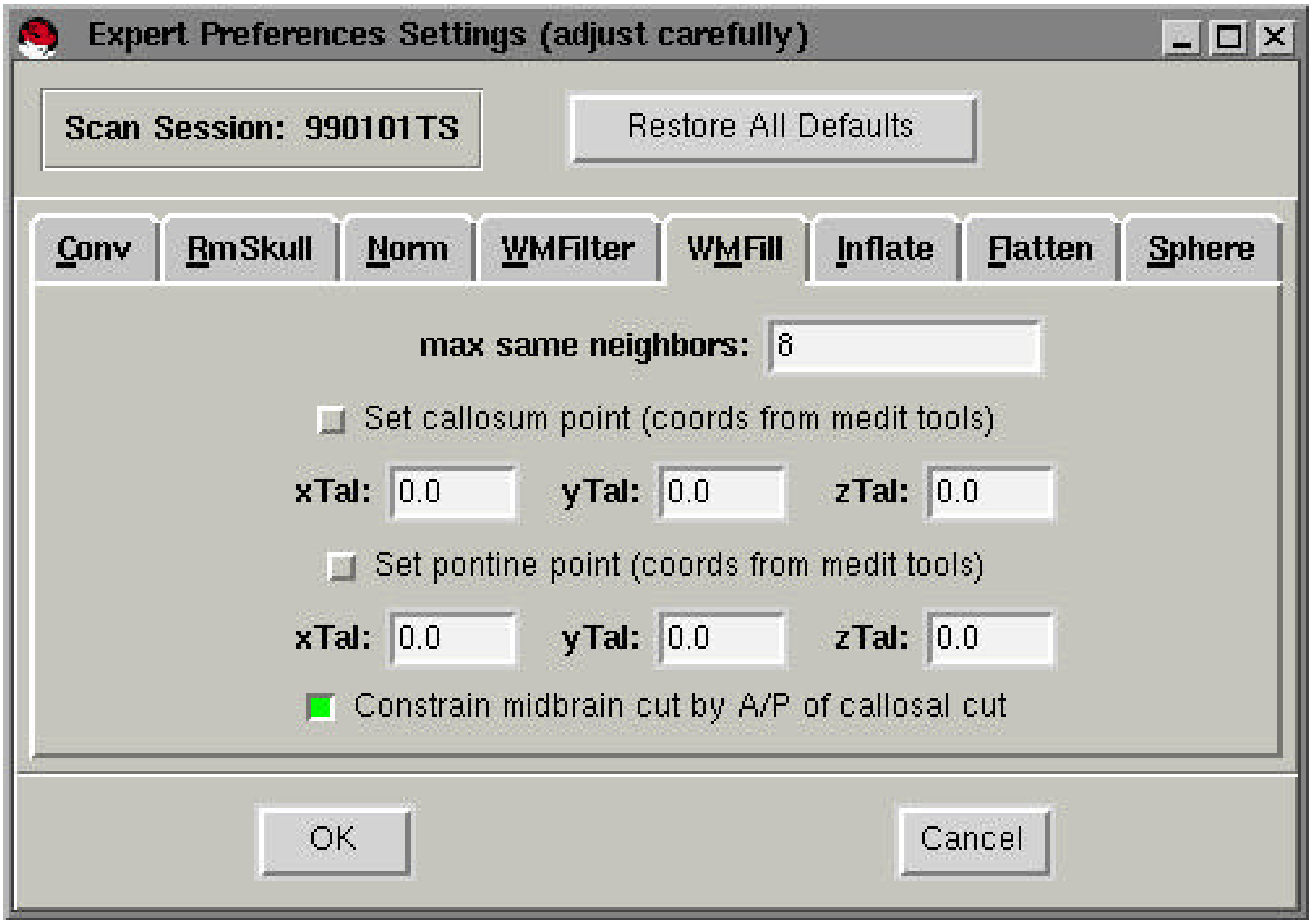


Select a point (left mouse button) in the corpus callosum to get the Talairach coordinates. Be sure that **View Logs** is selected. The Talairach coordinates will be printed in the **csurf** window under the **medit** bar.

Select **Expert Preferences** from the **Preferences** menu.



In the **Expert Preferences** window, select the **WMFill** tab.



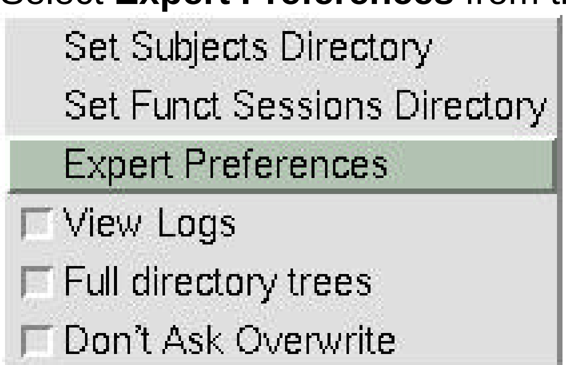
Enter the Talairach coordinates of the corpus callosum into the **WMFill** window and press the button next to "**Set callosum point**." If the Talairach transformation matrix is not available, enter the x, y, and z locations output by **medit**.

Pons:

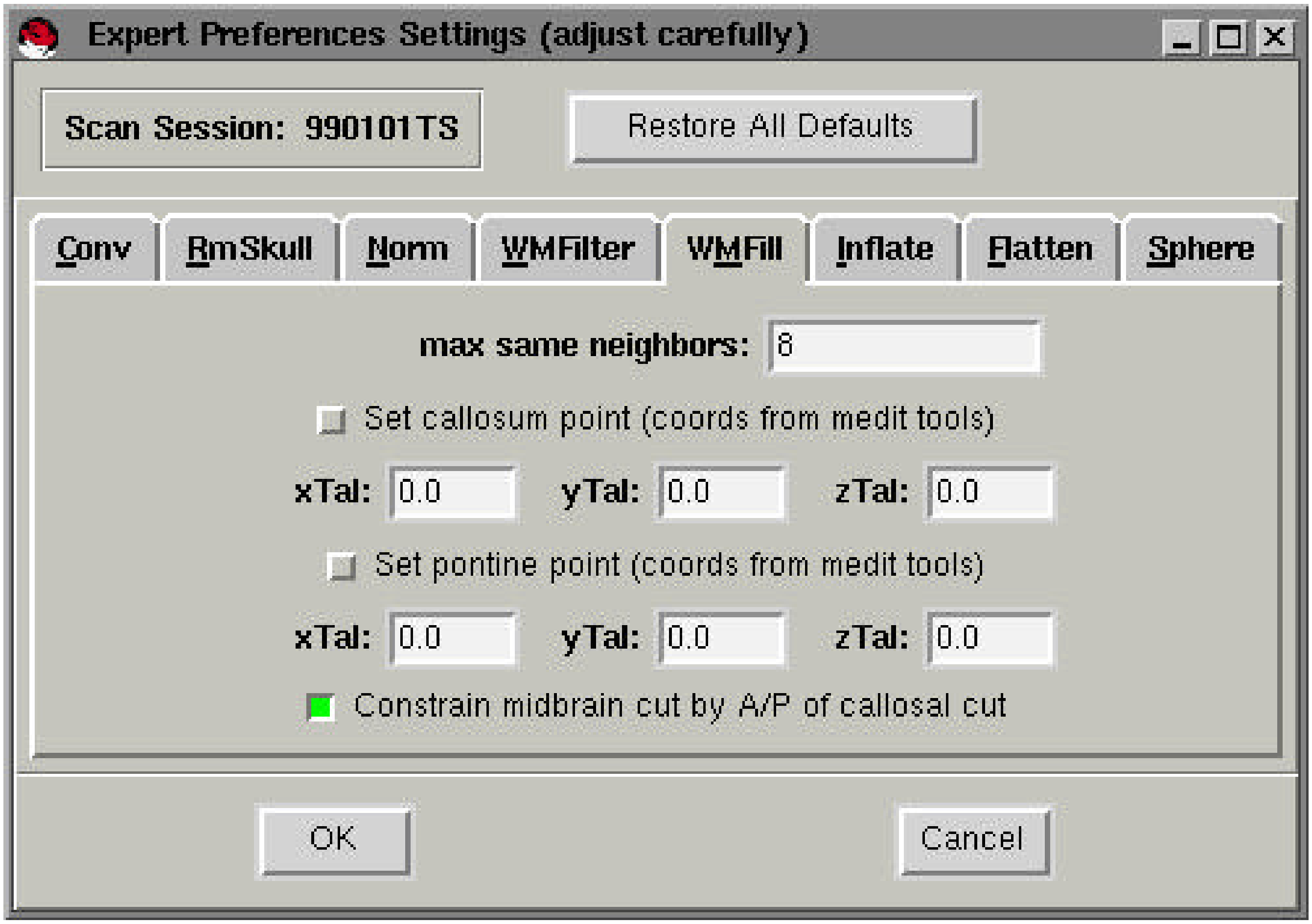
Using **medit**, find a horizontal slice in the **wm** volume in which the brainstem is disconnected from the rest of the brain. This slice should be near the top of the pons. If you cannot find a slice with the mid-brain disconnected, manually erase voxels to separate the mid-brain (erase voxels with the right mouse button in medit).

Select a point (left mouse button) in the brainstem to get the Talairach coordinates. Be sure that **View Logs** is selected. The Talairach coordinates will be printed in the **csurf** window under the **medit** bar.

Select **Expert Preferences** from the **Preferences** menu.

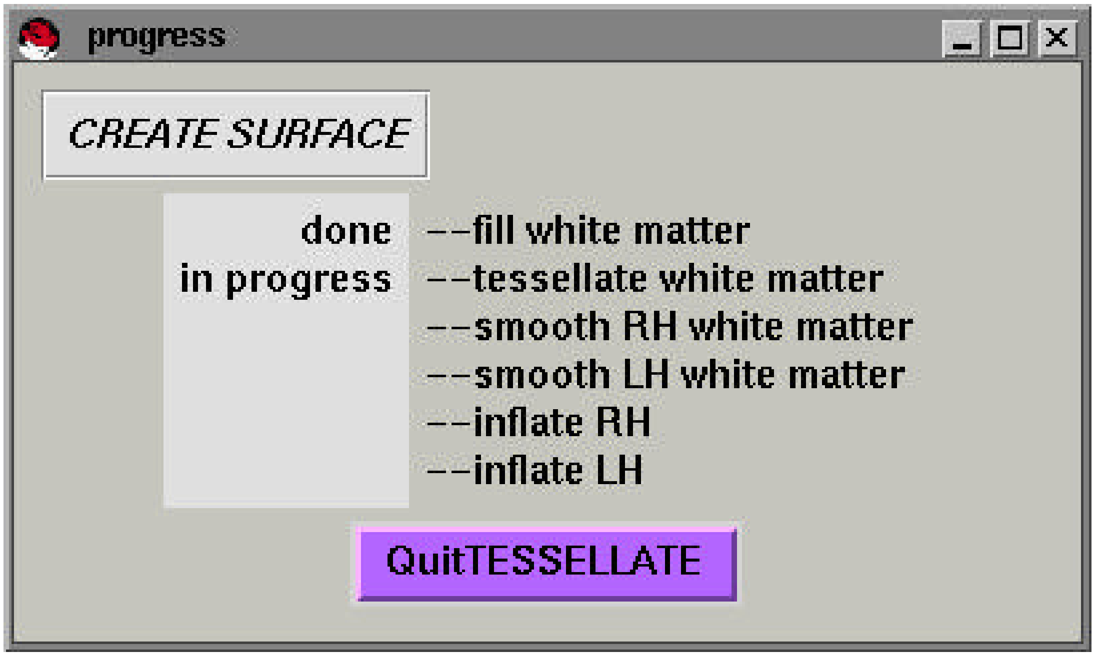


In the **Expert Preferences** window, select the **WMFill** tab.



Enter the Talairach coordinates of the pons into the **WMFill** window and press the button next to "**Set pontine point**." If the Talairach transformation matrix is not available, enter the x, y, and z locations output by **medit**.

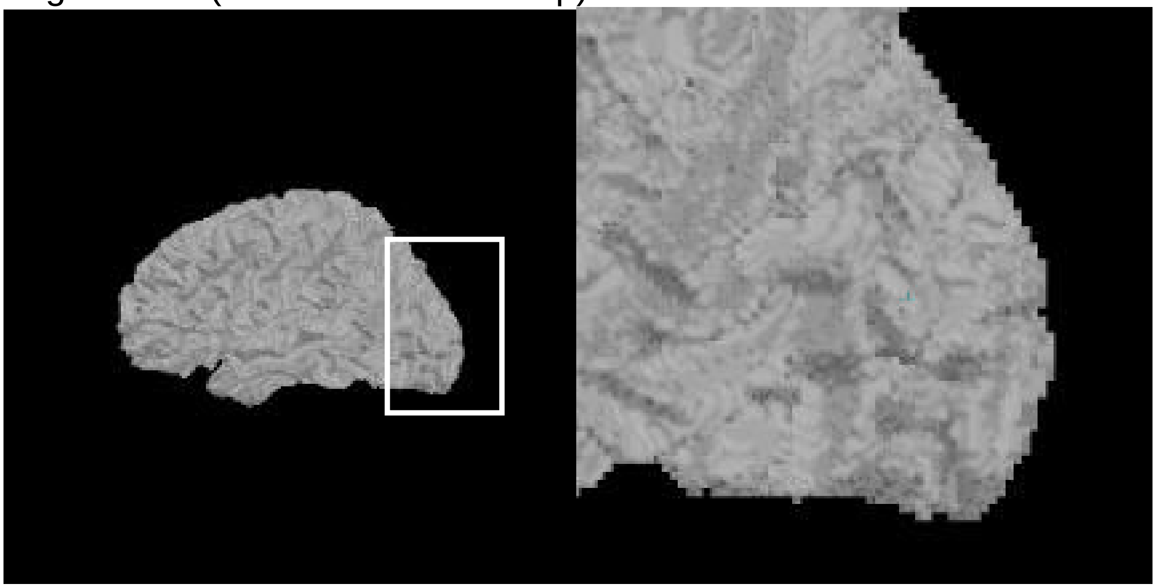
Part 2: Tessellate White Matter



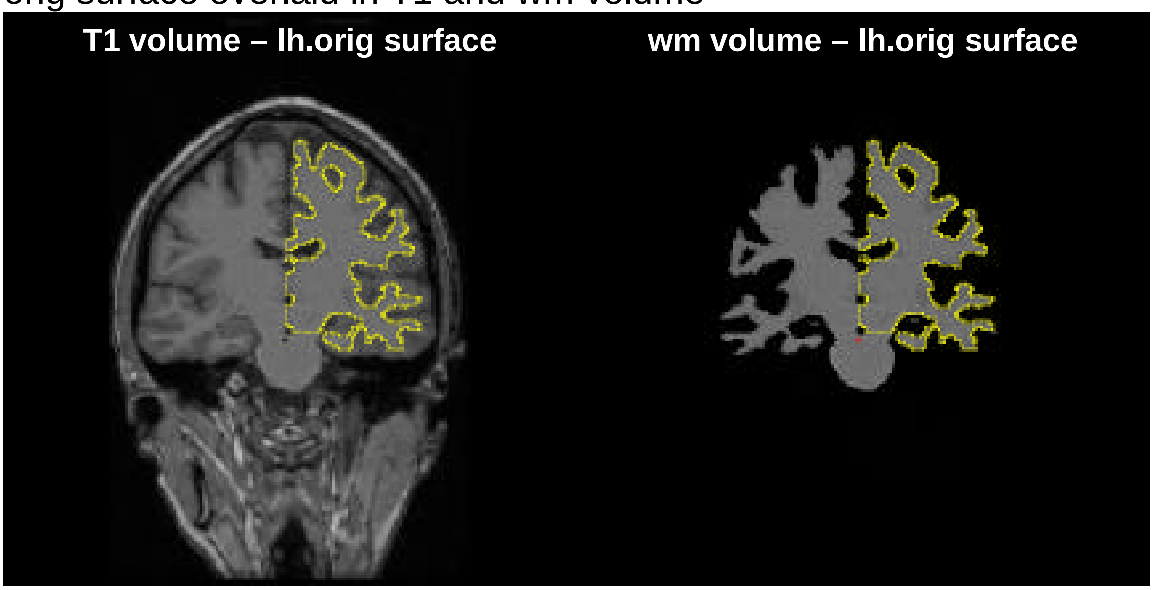
Connects the surfaces of the filled white matter voxels into a continuous surface.

The output files written by this procedure are:
surface: $SUBJECTS_DIR/$name/surf/rh.orig
surface: $SUBJECTS_DIR/$name/surf/lh.orig

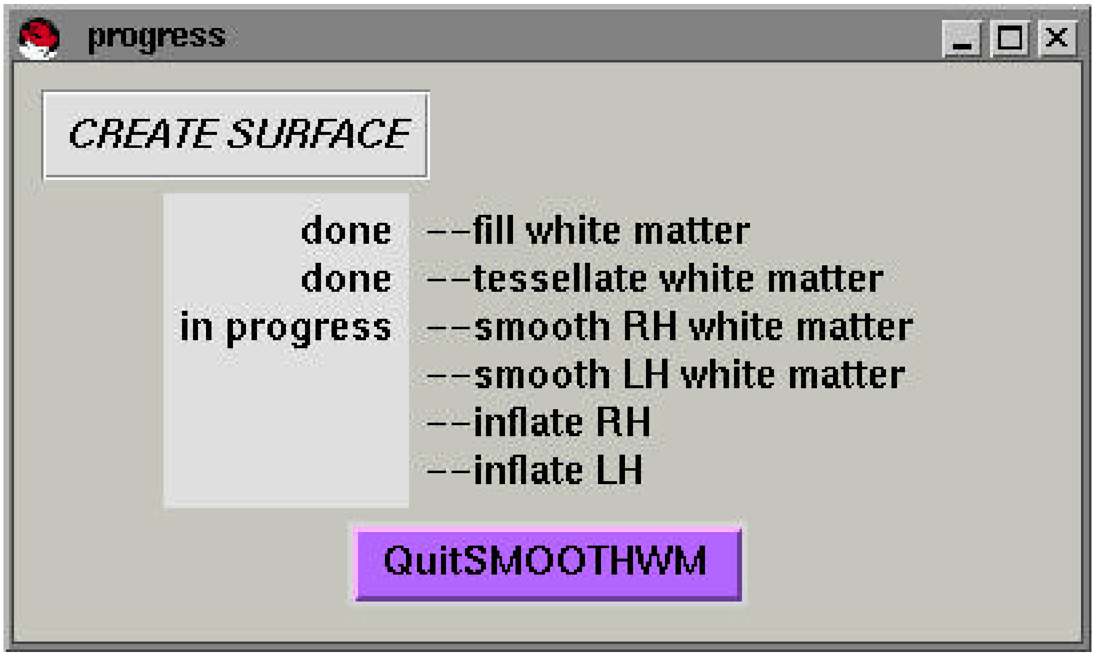
orig surface (full view and close up)



orig surface overlaid in T1 and wm volume



Part 3: Smooth Right Hemisphere White Matter



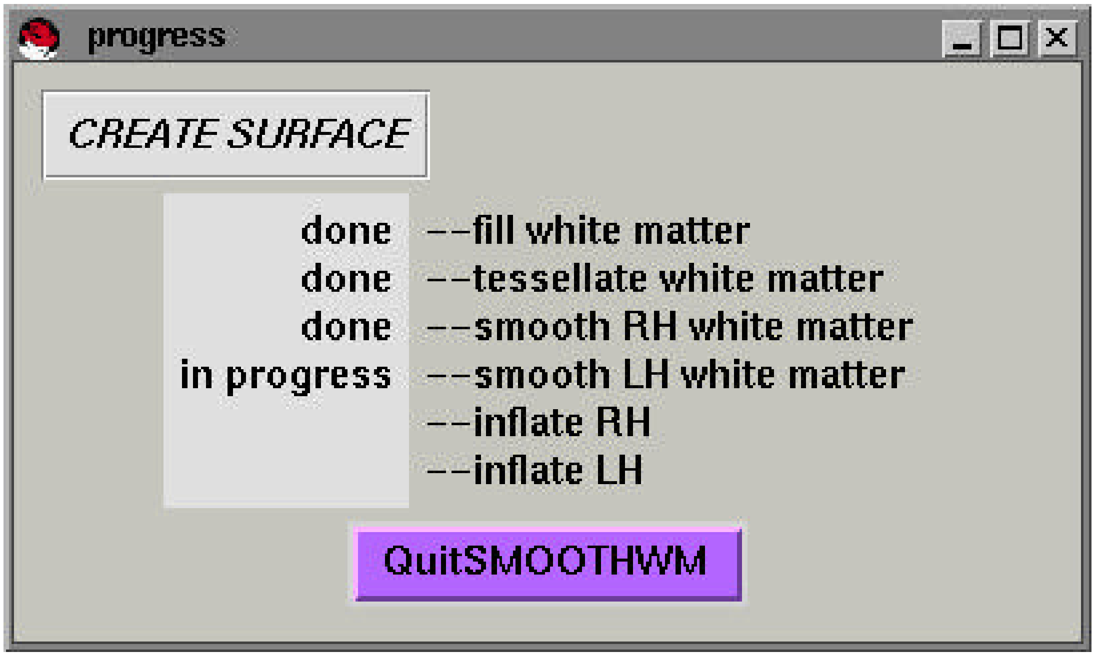
The reconstructed white matter surface of the right hemisphere (rh.orig) is smoothed. The mean curvature is computed from the smooth surface.

The output files written by this procedure are:

surface: $SUBJECTS_DIR/$name/surf/rh.smoothwm

curv: $SUBJECTS_DIR/$name/surf/rh.curv

Part 4: Smooth Left Hemisphere White Matter



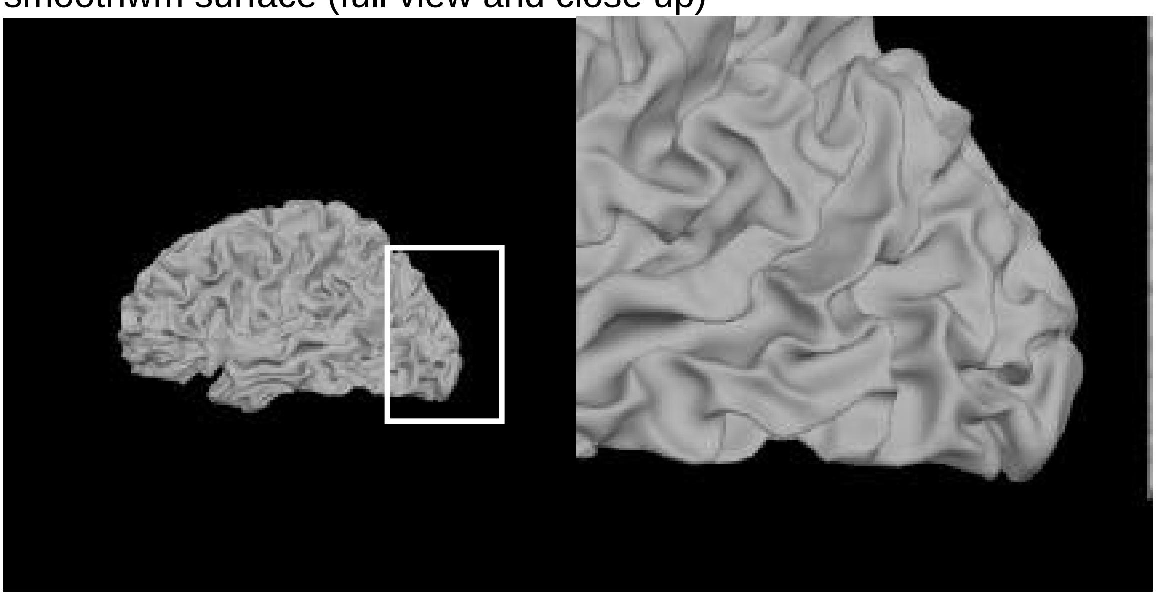
The reconstructed white matter surface of the left hemispheres (lh.orig) is smoothed. The mean curvature is computed from the smooth surface.

The output files written by this procedure are:

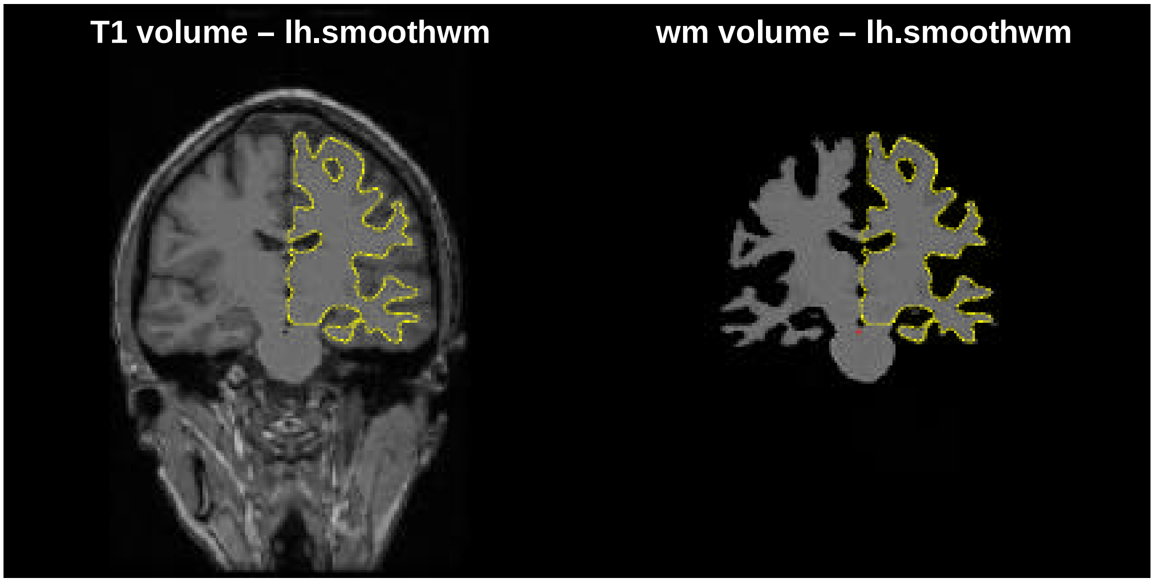
surface: $SUBJECTS_DIR/$name/surf/lh.smoothwm

curv: $SUBJECTS_DIR/$name/surf/lh.curv

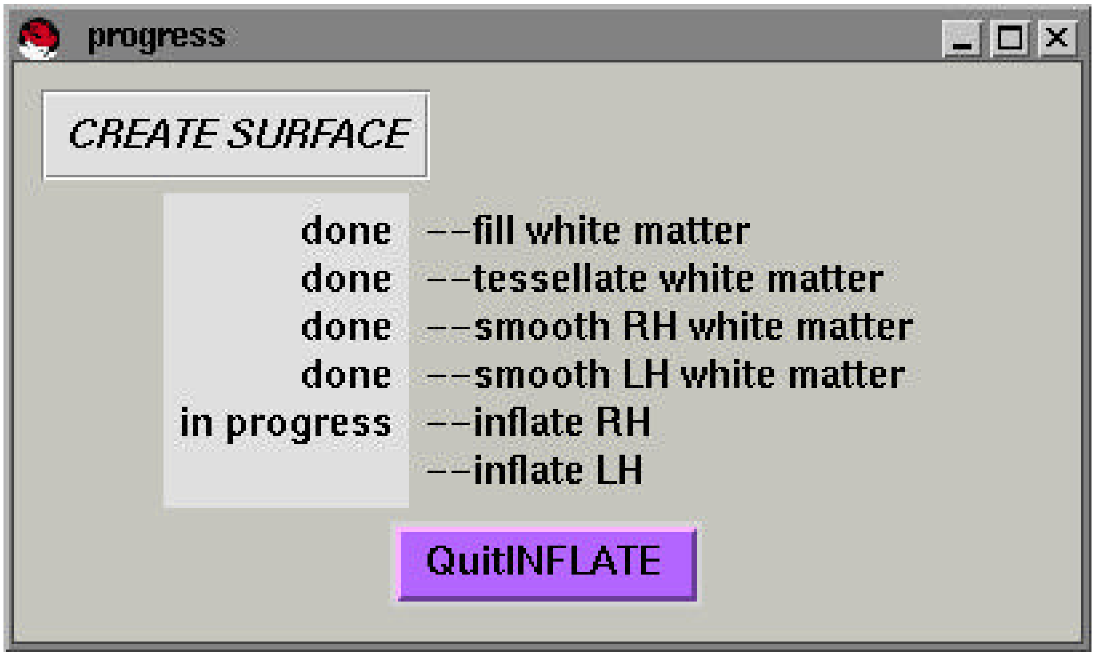
smoothwm surface (full view and close up)



smoothwm surface overlaid in T1 and wm volume



Part 5: Inflate Right Hemisphere

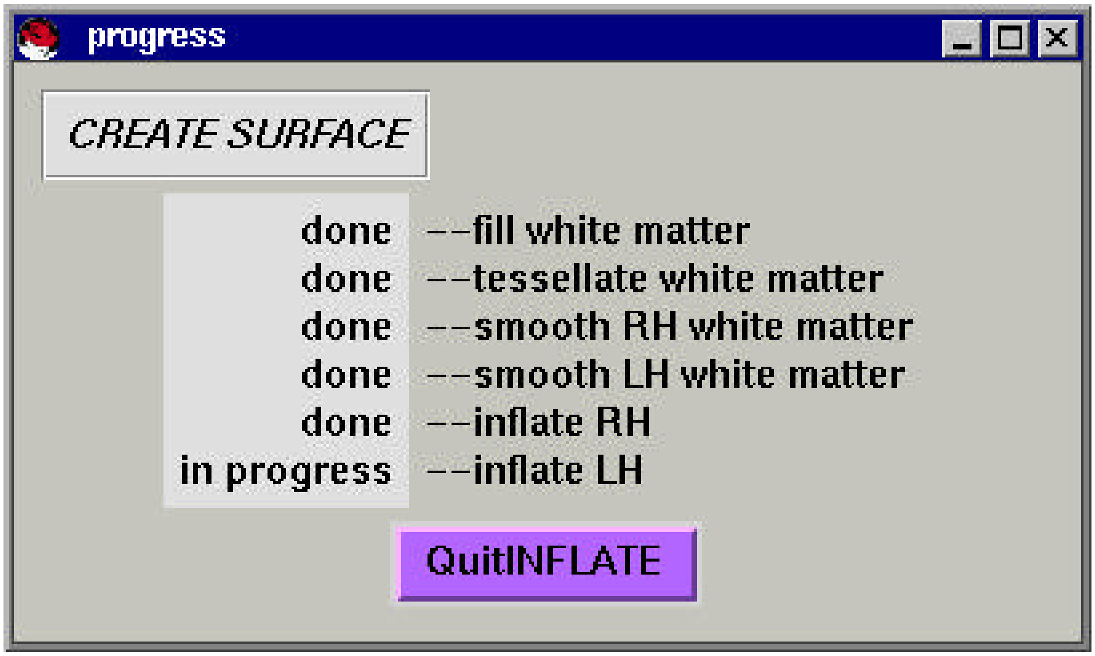


Inflates the right hemisphere surface (rh.smoothwm) while attempting to minimize metric distortion.

The output file written by this procedure is:

surface: $SUBJECTS_DIR/$name/surf/rh.inflated

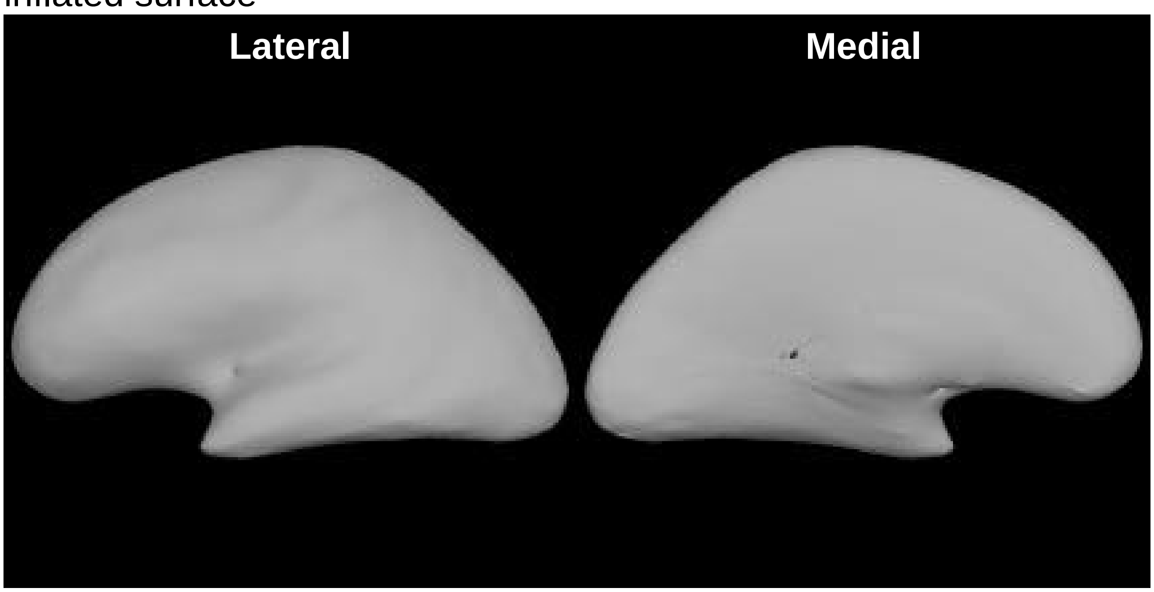
Part 6: Inflate Left Hemisphere



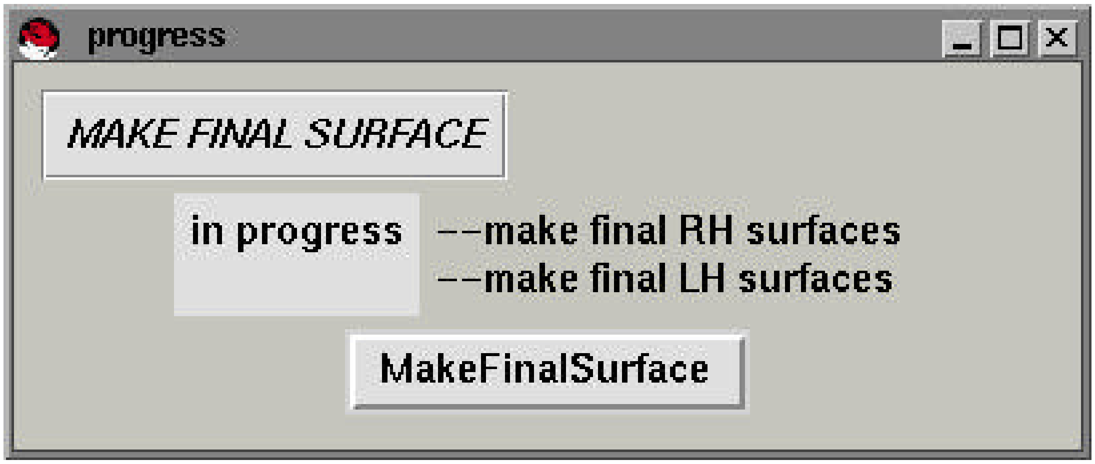
Inflates the left hemisphere surface (lh.smoothwm) while attempting to minimize metric distortion.

The output file written by this procedure is:
surface: $SUBJECTS_DIR/$name/surf/lh.inflated

inflated surface



Make Final Surface



Starts a two-part background process to create the final left and right hemisphere cortical surfaces (after removal of topological defects). **Make Final Surface** should only be run once the surface editing is complete (i.e. the surface is topologically correct.

Part 1: Make Final Right Hemisphere Surfaces

The output files written by this procedure are:

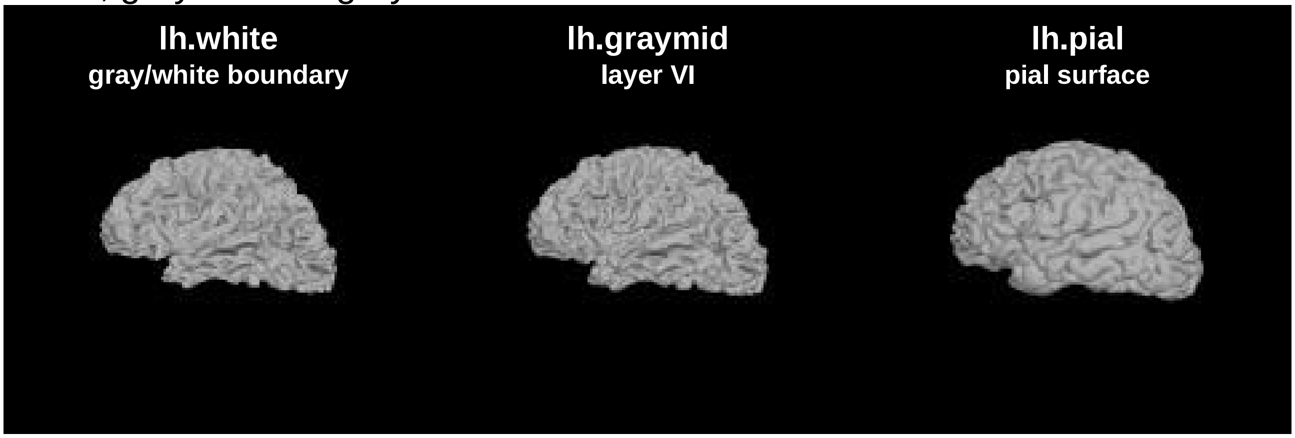
surface: $SUBJECTS_DIR/$name/surf/rh.white
surface: $SUBJECTS_DIR/$name/surf/rh.graymid
surface: $SUBJECTS_DIR/$name/surf/rh.pial

Part 2: Make Final Left Hemisphere Surfaces

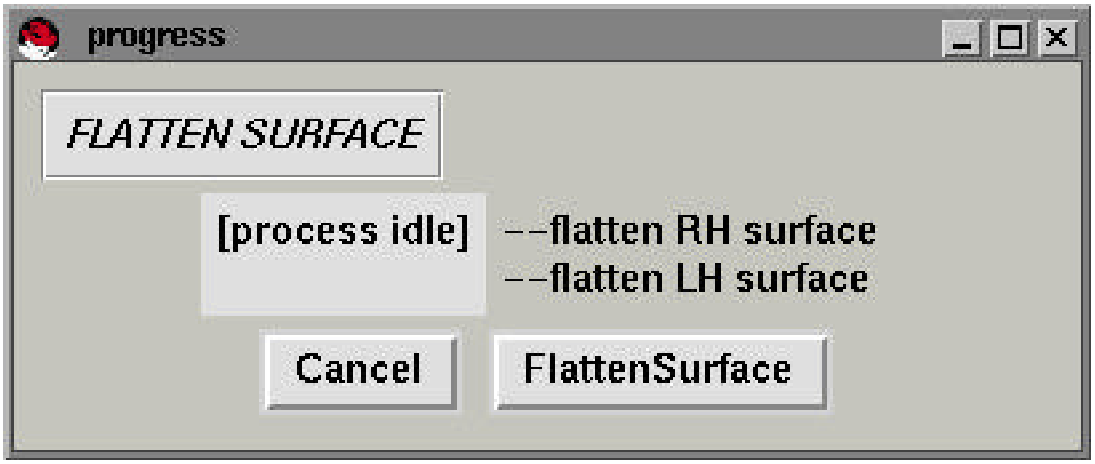
The output files written by this procedure are:

surface: $SUBJECTS_DIR/$name/surf/lh.white
surface: $SUBJECTS_DIR/$name/surf/lh.graymid
surface: $SUBJECTS_DIR/$name/surf/lh.pial

White, graymid and gray surfaces



Flatten Surface



Starts a two-part background process to create the flattened left and right hemisphere cortical surfaces (or surface portions). Estimates geodesic distances on smoothwm surface representation and minimizes metric distortions of flattened representation. (16-30 hours per full surface, 12 hours for the occipit surface).

Part 1: Flatten Right Hemisphere Surfaces

The output file written by this procedure is:

surface: $SUBJECTS_DIR/$name/surf/rh.*.patch.flat

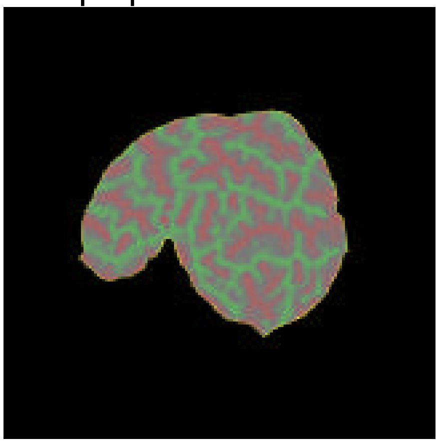
Part 2: Flatten Left Hemisphere Surfaces

The output file written by this procedure is:

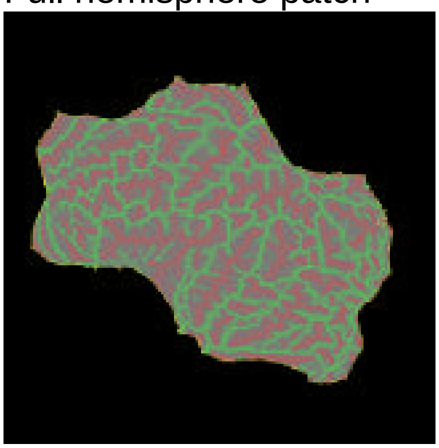
surface: $SUBJECTS_DIR/$name/surf/lh.*.patch.flat

Typical flattened patches:

Occipit patch



Full hemisphere patch



Sphere Surface

Starts a four-part background process to create the spherical left and right hemisphere cortical surfaces and then registers them with an average spherical cortical surface representation.

Part 1: Sphere Right Hemisphere Surface

The output file written by this procedure is:

surface: $SUBJECTS_DIR/$name/surf/rh.sphere

Part 2: Sphere Left Hemisphere Surface

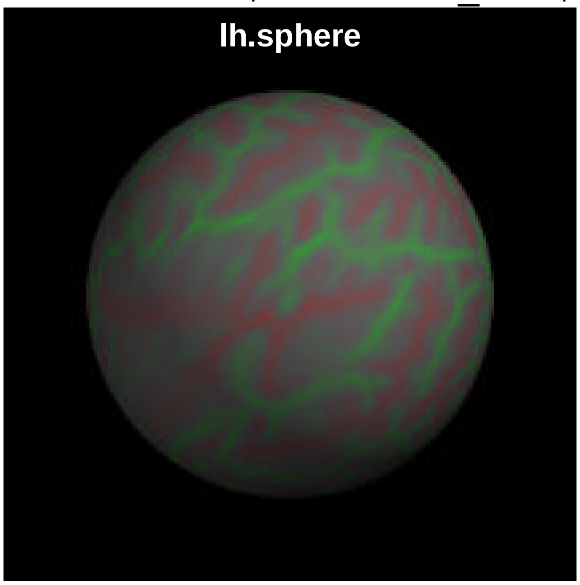
The output file written by this procedure is:

surface: $SUBJECTS_DIR/$name/surf/lh.sphere

Part 3: Register Right Hemisphere Surface

The output file written by this procedure is:

surface: $SUBJECTS_DIR/$name/surf/rh.sphere.reg



Part 4: Register Sphere Left Hemisphere Surface

The output file written by this procedure is:

surface: $SUBJECTS_DIR/$name/surf/lh.sphere.reg

Tools Menu – Functional Commands

The three functional commands under tools menu perform the following functions :

Conversion of the functional data into bshorts

Sampling of the statistical volume using the reconstructed surface (we refer to as "painting") which associates a statistical value with each vertex in the cortical surface

Rendering of the cortical surface that has been "painted" with the statistics. Additional controls are provided to vary the visual appearance or the rendering The statistical volume can also be viewed overlaid in the high resolution MRI volume.

Setup Functional Scans

Selects the functional directories and determines the resolution of the scans. Currently supported formats are: SPM/analyze, AFNI, and bfloat

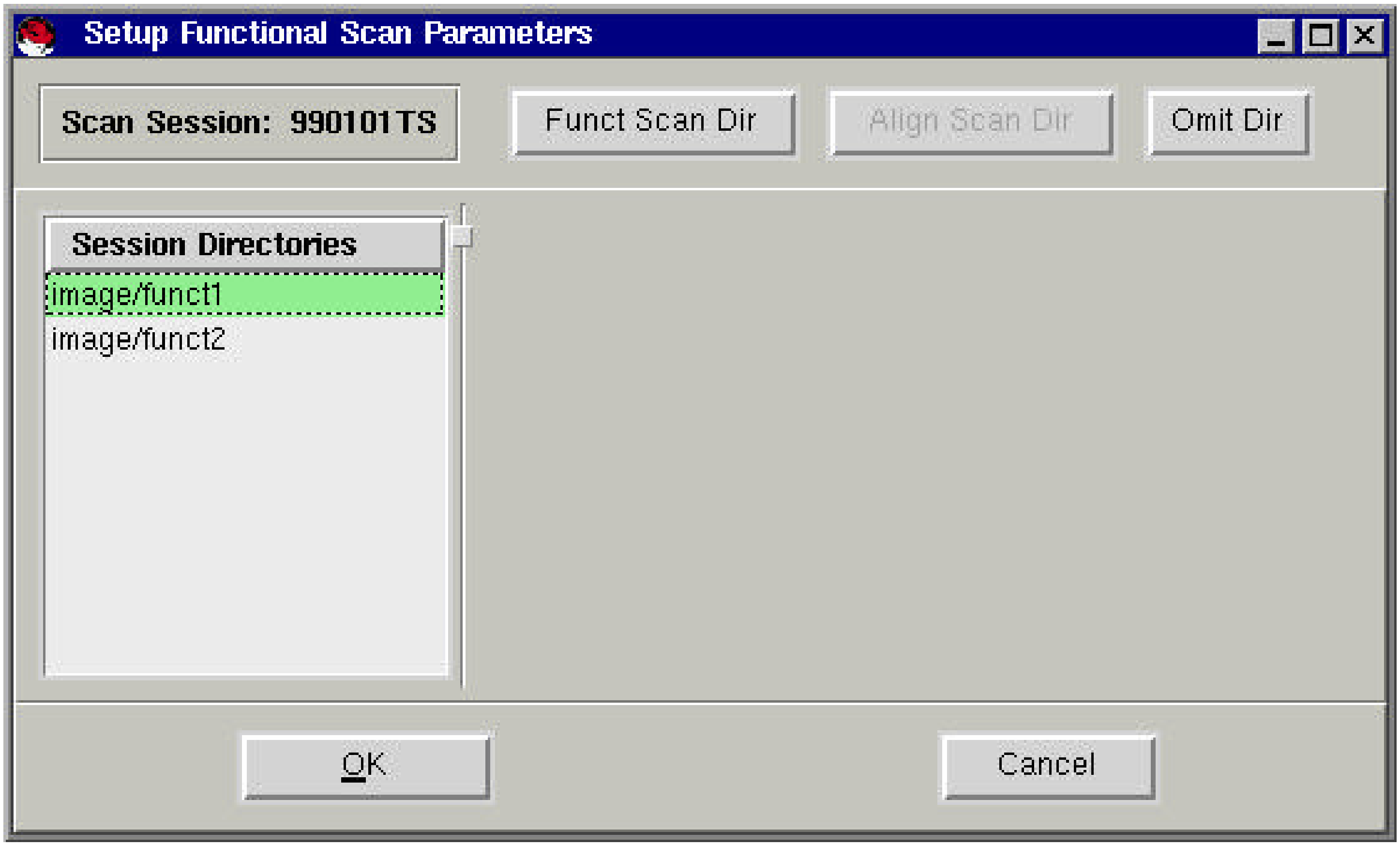
Each functional acquisition must be in its own directory in the **image** directory. Within each functional directory there must be a file **register.dat**.

register.dat

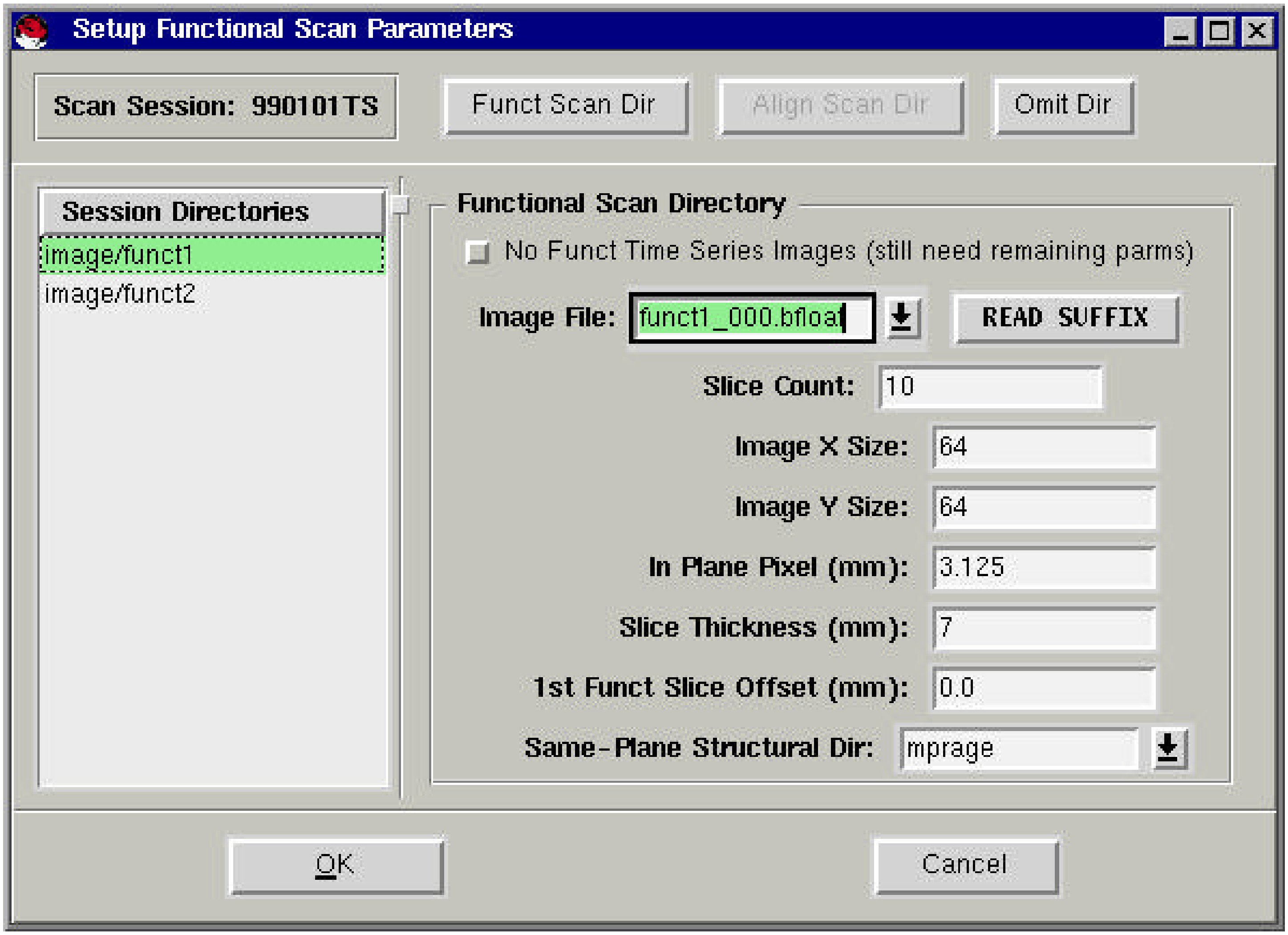
```
name_of_subject
in_plane_resolution(mm)
slice_thickness(mm)
brightness
transformation_matrix
```

Example

```
test_subject
3.125
7.000
0.100
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

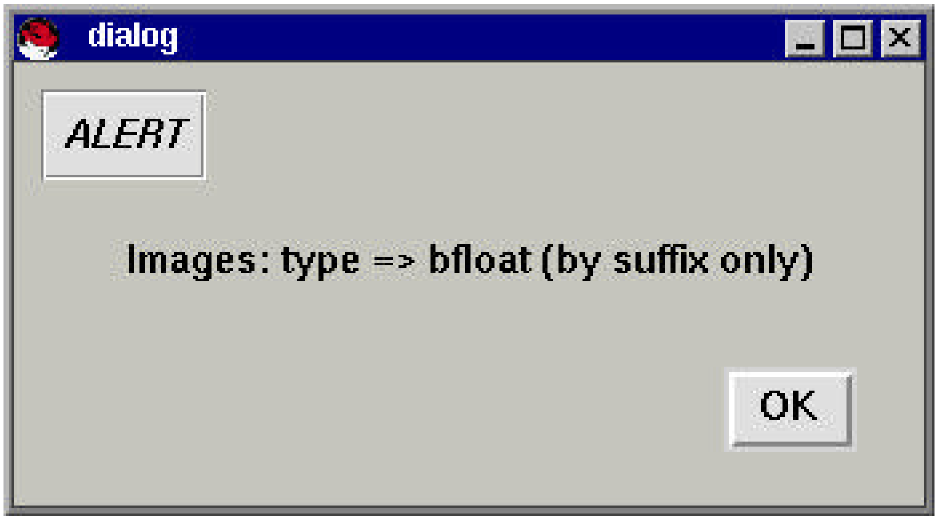


Select the first functional scan in the window on the left and press **Funct Scan Dir**.



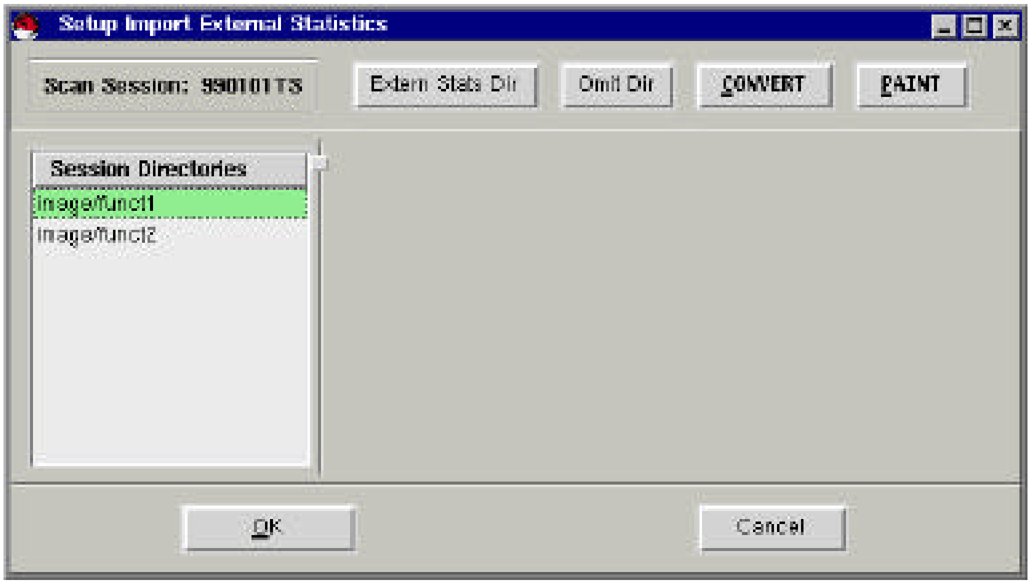
Manually correct any fields that were not correctly determined. If you want to omit a functional scan that was previously selected, reselect the directory and press **Omit Dir**.

The **READ SUFFIX** button simply determines the type of the scan based on the suffix.

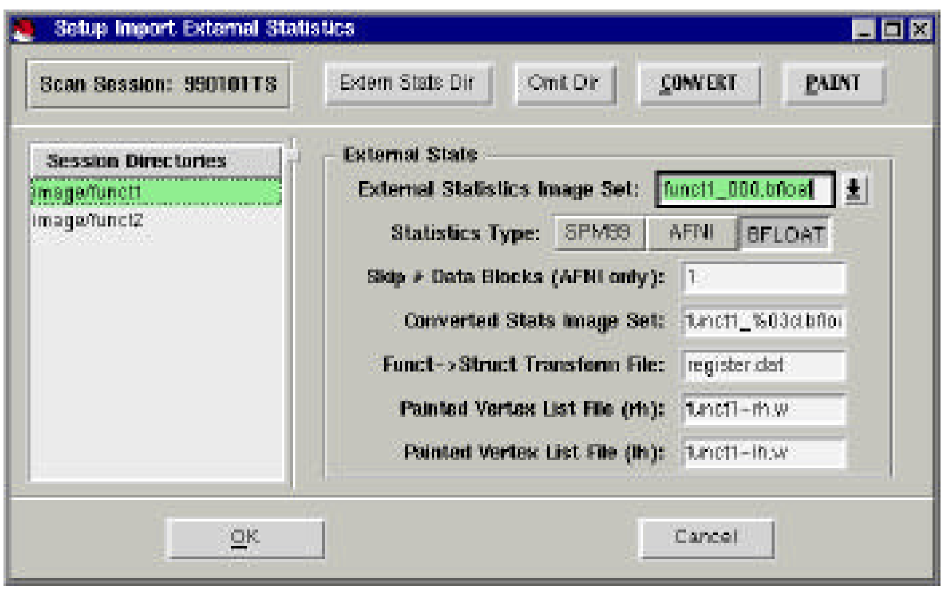


Setup Import Stats

Determines the filename structure for the imported statistical volume, converts the data to bshorts (if necessary), and samples the statistical volume using the **smoothwm** surface.



Select the external statistics directory in the left window and press **Extern Stats Dir**.



Description of each field:

Statistics Type: Select the type of statistical volume (either **SPM**, **AFNI**, or **BFLOAT**) with the appropriate button. If the statistic type is either SPM or AFNI, press **CONVERT** to convert to bshorts.

Skip # Data Blocks (AFNI only): For AFNI data, enter the number of data blocks to skip.

Converted Stats Image Set: Format string to describe the functional data

Funct->Struct Transform File: Default is **register.dat**. Do not edit.

Painted Vertex List File (rh): Defaults to <stem of the functional data>-rh.w

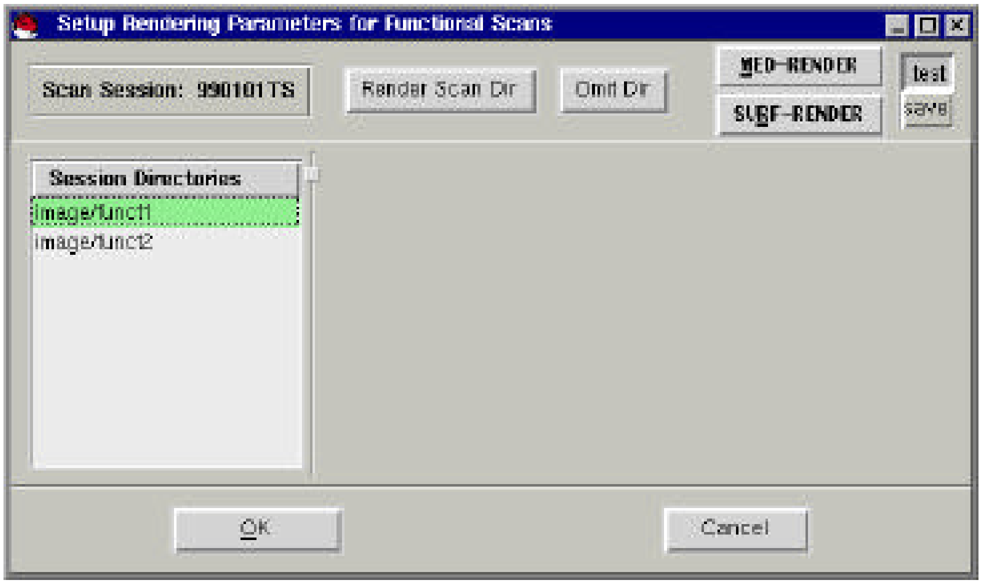
Painted Vertex List File (lh): Defaults to <stem of the functional data>-lh.w

To generate the statistical surface overlay (values only), press **PAINT**. This creates a list of vertices and the associated statistic sampled from the volume.

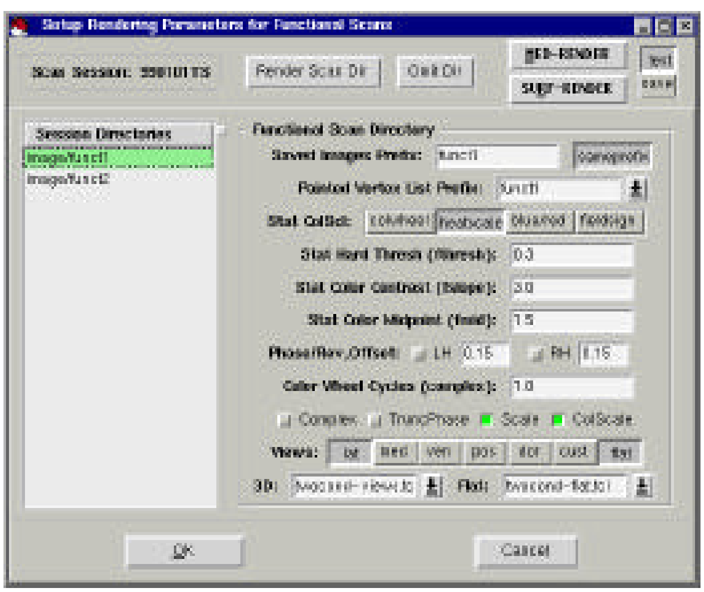
Repeat for each functional directory. If you want to omit a functional scan that was previously selected, reselect the directory and press **Omit Dir**.

Setup Rendering Parameters

Specify the rendering parameters for the overlaid statistical volume (previously selected using **Setup Import Statistics**). Provides both a volume overlay (using **medit**) and a surface overlay (using **surfer**).



Select the functional directory in the left window and press **Render Scan Dir**. Be sure that the functional directory that is selected in the left window matches the functional directory in the **scandir** field in the **csurf** window.



Description of each field:

Saved Images Prefix: Specifies the prefix for any saved images (rgb format). Images are saved in the **rgb** directory. Image are saved only if the **save** button is depressed. Default filename is <prefix>-<hemi>-<surface>.rgb.

Painted Vertex List Prefix: Must match the prefix specified in **Setup Import External Statistics**.

Stat ColScl: **heatscale** displays positive statistics in red and yellow and negative statistics in blue and green. **Colwheel**, **blue/red**, and **fieldsign** are currently unsupported (future functionality).

Stat Hard Thresh: Statistical threshold (for values below the threshold, the underlying curvature is displayed).

Stat Color Contrast: Color slope from **Stat Color Midpoint** (red/blue) to the maximum color (yellow/green). Maximum color (yellow, green) represents a statistical value of **Stat Color Midpoint** + 1/ **Stat Color Contrast**.

Stat Color Midpoint: Statistical value for full red/blue.

Phase/Rev, Offset: Currently unsupported (future functionality).

Color Wheel Cycles: Currently unsupported (future functionality).

Radiobuttons:

Complex: : Currently unsupported (future functionality).

TruncPhase: Only display positive values (red/yellow).

Scale: Display scale bar (1 cm).

ColScale: Display color scale bar.

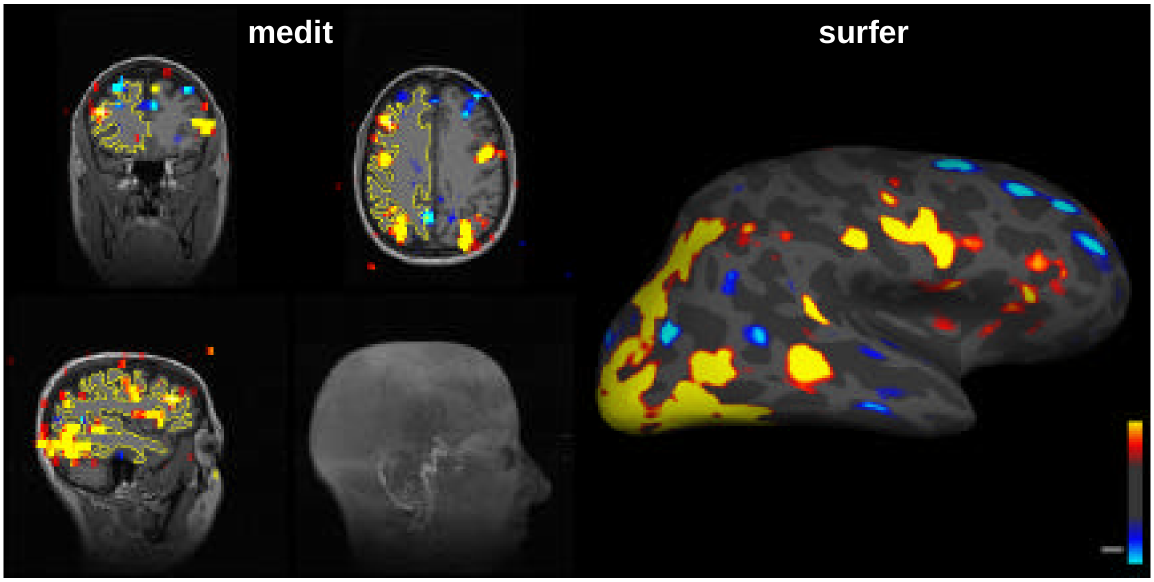
Pushbuttons:

Views: Specify which viewing orientations should be generated and saved. Can specify multiple orientations.

- **lat:** lateral
- **med:** medial
- **pos:** posterior
- **dor:** dorsal
- **cus:** custom
- **flat:** flattened patch

To view the statistical volume overlaid into the T1 volume, press **MED-RENDER**. To view the statistical volume painted onto a surface or flattened patch, press **SURF-RENDER**. Select the surface or patch in the **csurf** window (**surface** or **patch** field). To save the images generated by **SURF-RENDER**, press the **save** button. To view (without saving) the images generated by **SURF-RENDER**, press the **test** button.

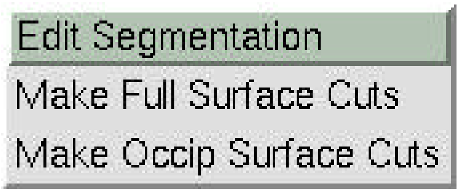
An example of the statistical overlay in the volume in 3 views (left) and the corresponding surface overlay (right) are shown below:



Repeat for the setup each functional directory. If you want to omit a functional scan that was previously selected, reselect the directory and press **Omit Dir**.

If you begin to experience errors with the rendering, exit **Setup Rendering Parameters** and begin again. Also be sure that the functional scan that you are rendering is also correctly listed in the **scandir** field in the **csurf** window.

Edit Menu



Edit Segmentation

Starts both medit (to edit the white matter volume) and surfer (to view the inflated cortical surface). Medit reads in the wm (white matter) volume with the orig surface overlaid and surfer reads in the inflated surface.

For a detailed description of how to fix topological defects, see the section titled "**Fixing Topological Defects in the Cortical Surface**."

Make Full Surface Cuts

Starts surfer using the inflated surface

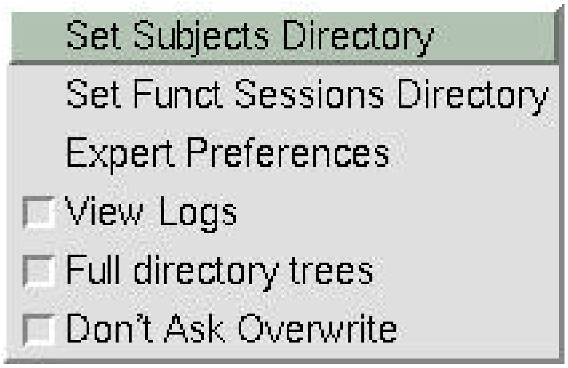
For a detailed description of how to make the full surface cuts, see the section titled "**Cutting the Full Surface**."

Make Occipit Surface Cuts

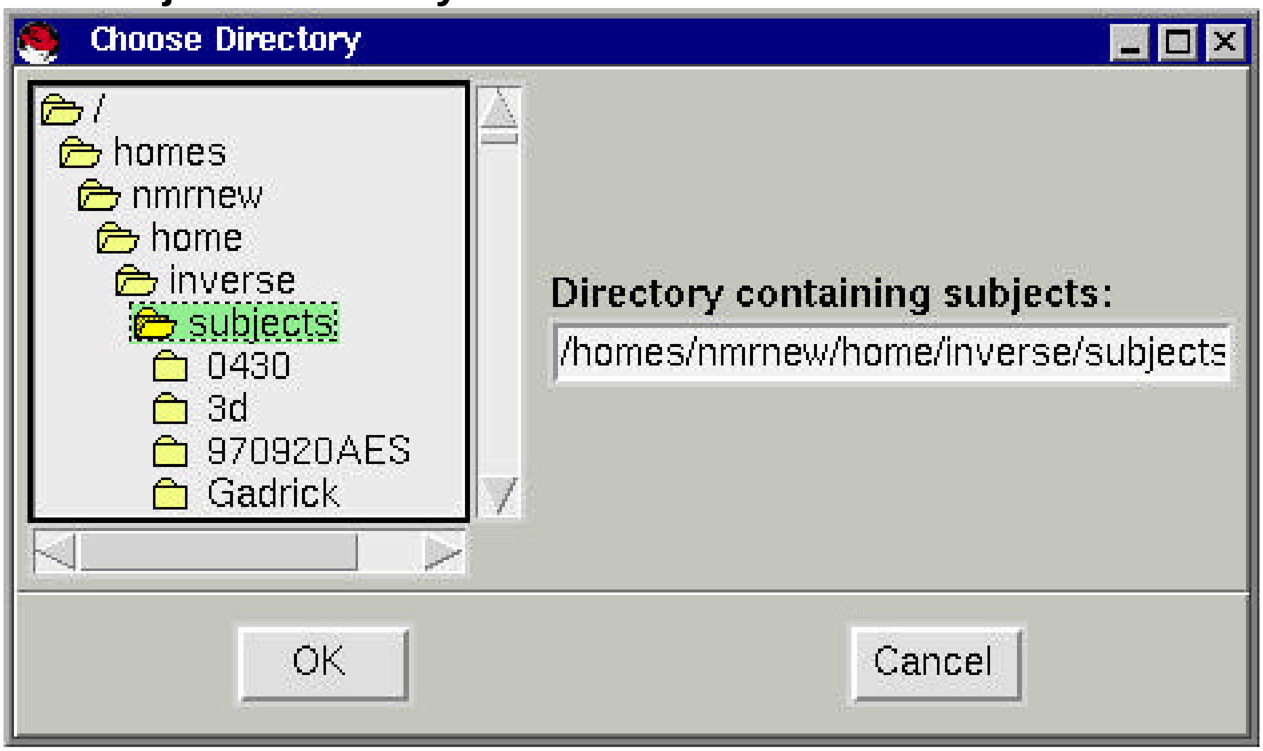
Starts surfer using the inflated surface

For a detailed description of how to make the occipit surface cuts, see the section titled "**Cutting the Occipit Surface**."

Preferences

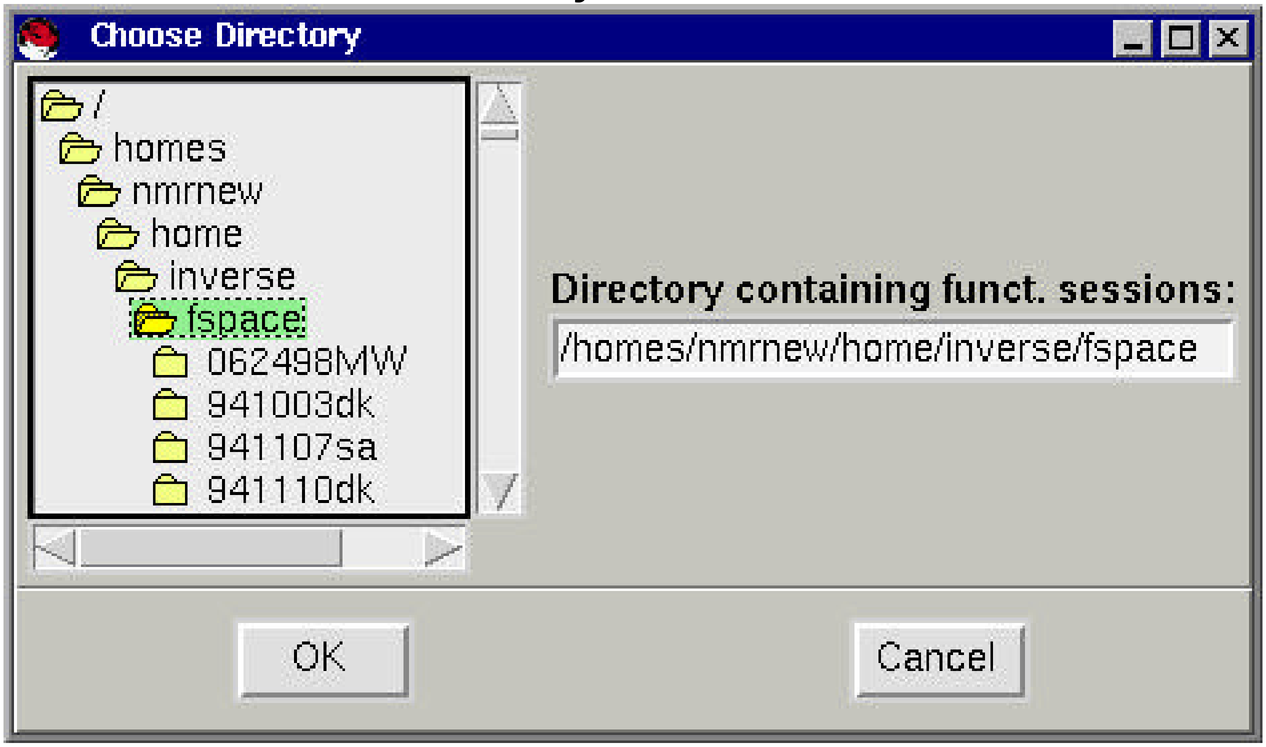


Set Subjects Directory



Sets the default directory for the subjects' structural data, volumes and surfaces.

Set Funct Sessions Directory

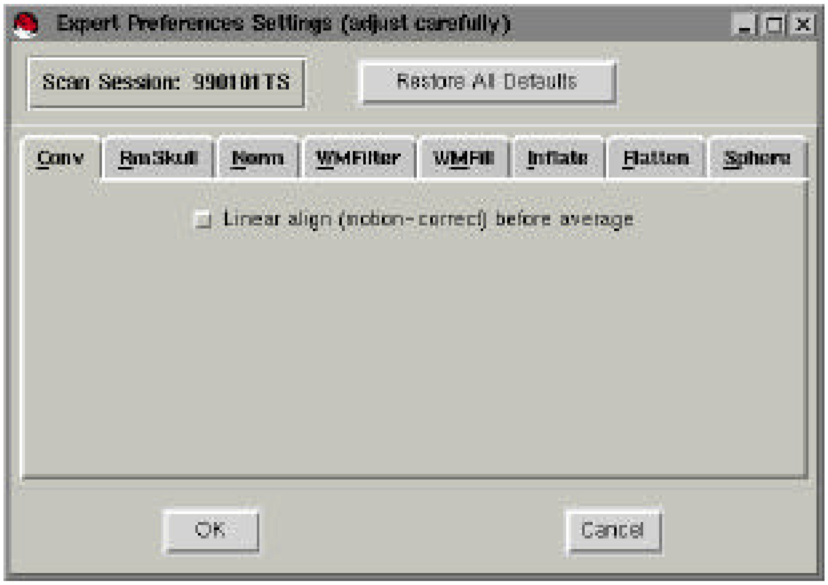


Sets the default directory for the subjects' functional data.

Expert Preferences

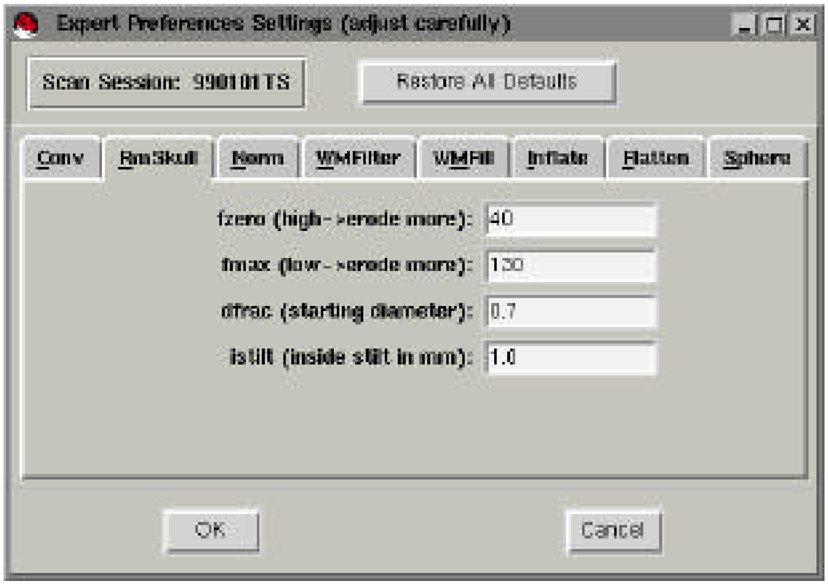
Provides additional parameters which can be adjusted if necessary.
To return to the original parameters, press **Restore All Defaults**.

Conv



Not supported. Do not modify.

Rmskull



These parameters are used in the skull stripping part of **Process Volume**.

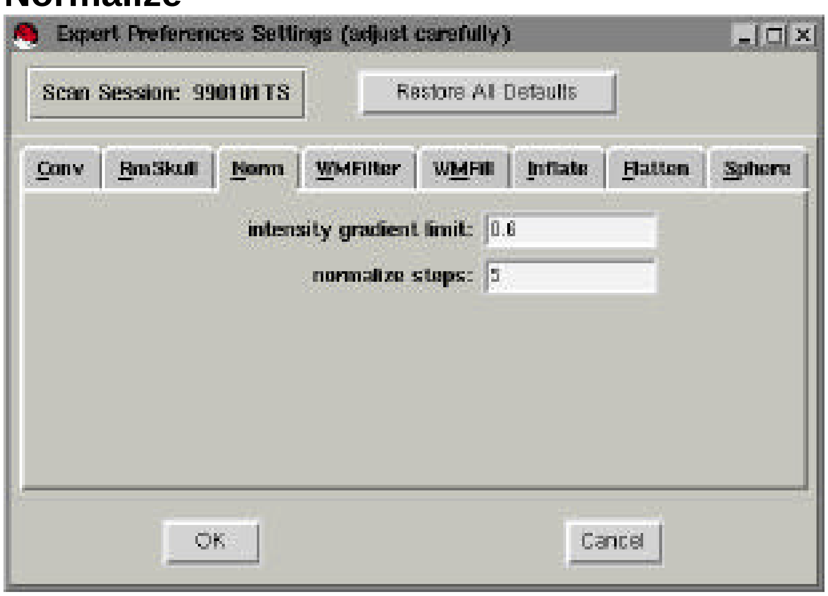
fzero: Minimum value while estimating outer brain surface. Larger values will strip more.

fmax: Maximum value for while estimating outer brain surface. Smaller values will strip more.

dfrac: Initial size of estimated outer brain surface.

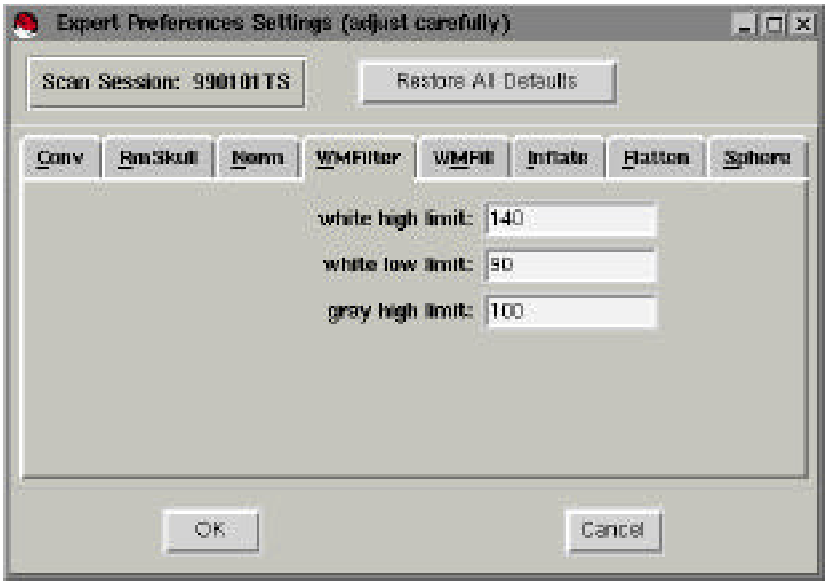
Istilt: Minimum distance of outer brain surface from outer brain.

Normalize



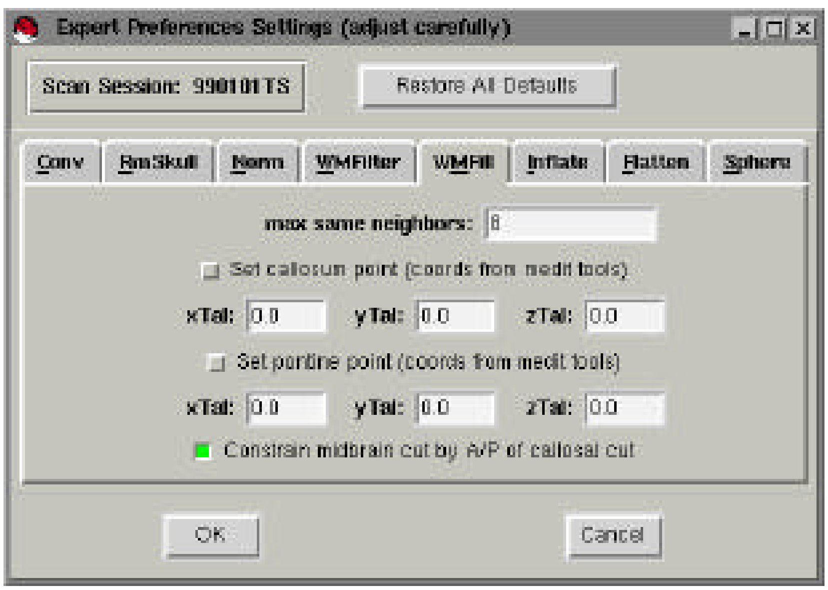
Not supported. Do not modify.

WMFilter



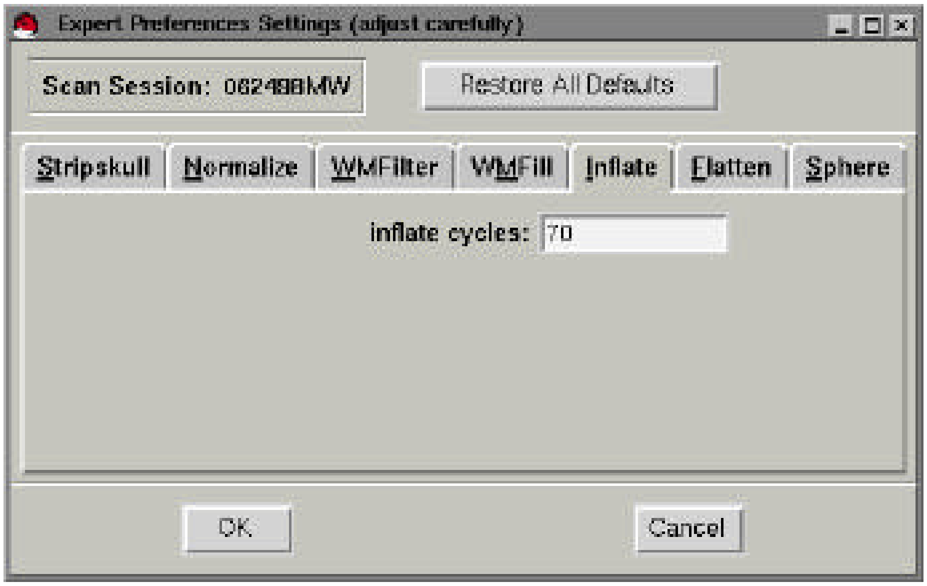
Not supported. Do not modify.

WMFill



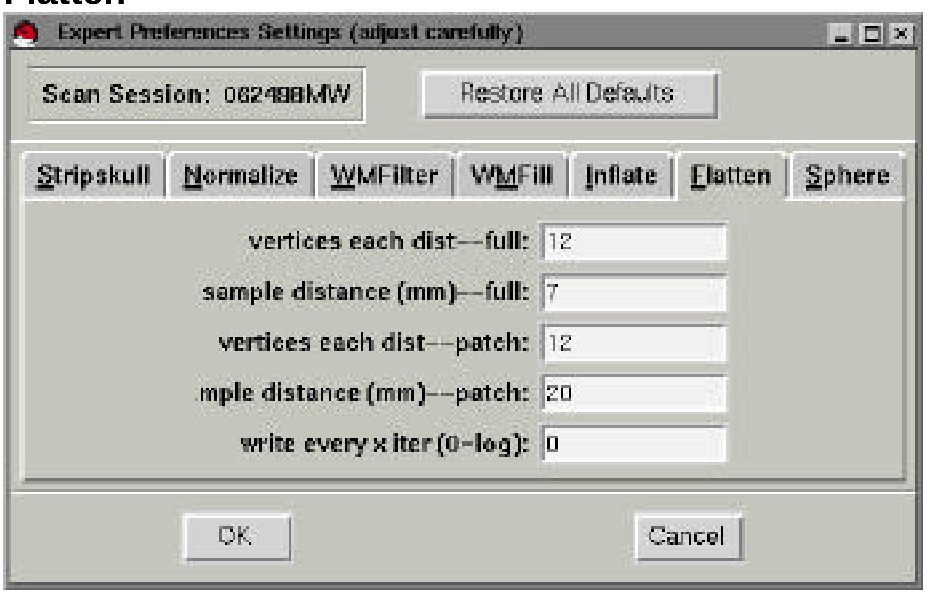
The coordinates specified here are used in **Part 1: Fill White Matter** of **Create Surface**. If the cutting planes fail, enter the Talairach coordinates of the corpus callosum or pons into the **WMFill** window and press the button next to "**Set callosum point**" or "**Set pontine point**." If the Talairach transformation matrix is not available, enter the x, y, and z locations output by **medit**.

Inflate



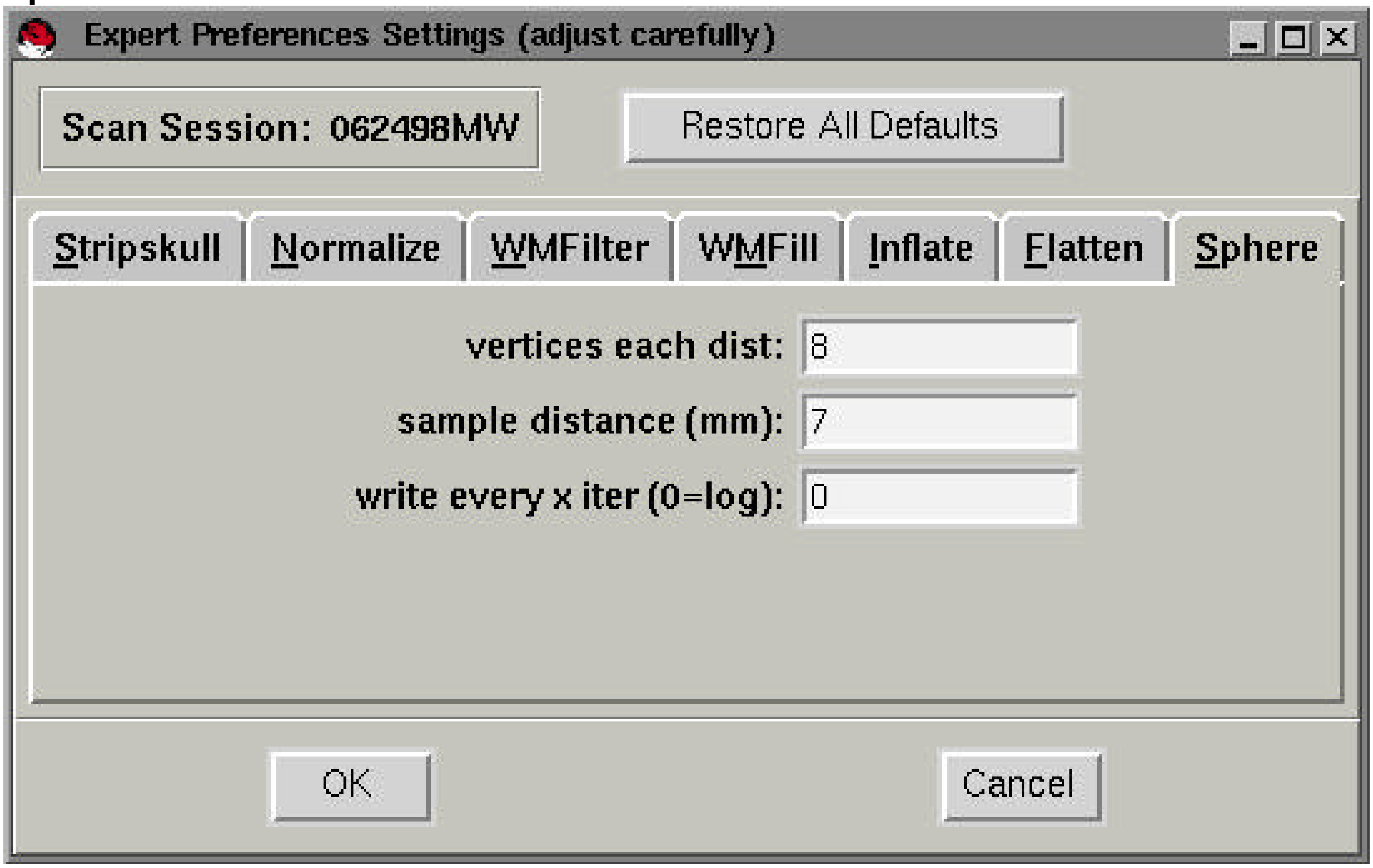
This parameter is used during the cortical surface inflation in **Create Surface**. Fewer **inflate cycles** will result in a less smooth surface.

Flatten



These parameters are used during the flattening of the cortical surface in **Flatten Surface**.
vertices each dist: number of vertices to examine
sample distance: distance limit to sample vertices.
write every x iter: writes out an intermediate surface at each **x** iterations
Fewer vertices and smaller sample distances will result in more metric distortion, but will require less memory and run faster.

Sphere



These parameters are used during the spherical morphing of the cortical surface inflation in **Sphere Surface**.

vertices each dist: number of vertices to examine

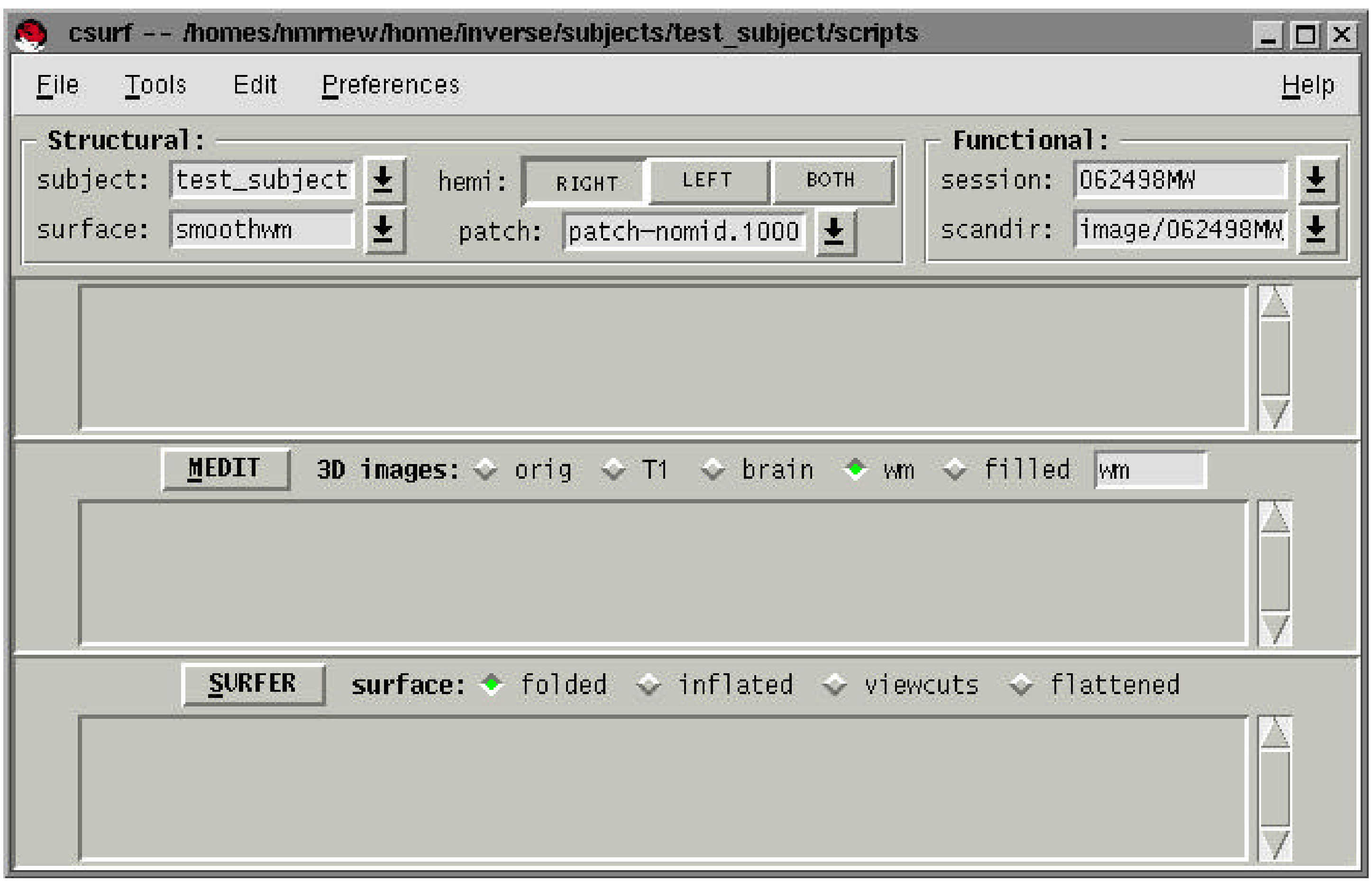
sample distance: distance limit to sample vertices.

write every x iter: writes out an intermediate surface at each of the specified iterations

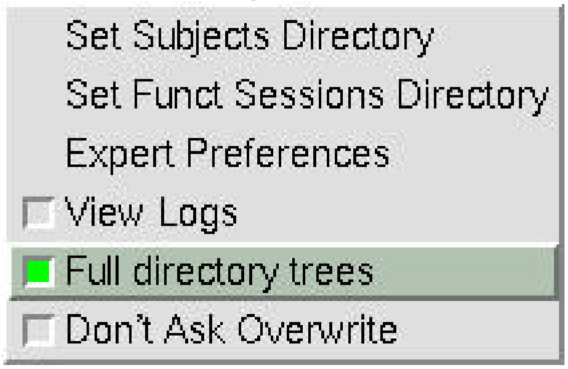
Fewer vertices and smaller sample distances will result in more metric distortion, but will require less memory and run faster.

View Logs

Enlarges the main **csurf** window to display 3 additional log windows. The top window shows the text output of processes run through the **csurf** interface (except for **medit** and **surfer**). For example, output for the various steps in **Process Volume** are displayed in the top window. Text output for medit and surfer are shown in the middle and lower windows, respectively. For example, the coordinates of a selected point (MRI and Talairach coordinates, when available) are displayed in these windows.

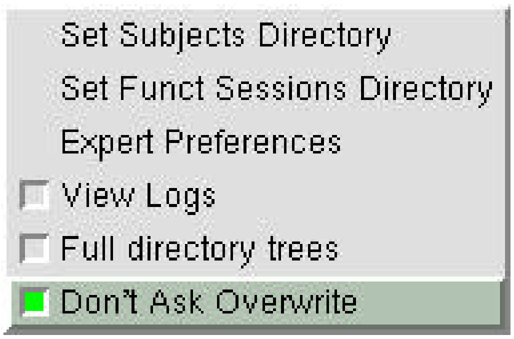


Full directory trees



When selecting directories, displays entire directory trees. This option can be quite slow for large file systems.

Don't Ask Overwrite



Automatically regenerates the volumes and surfaces created by the selections under the **Tools** menu. This is equivalent to selecting **Redo** in the Replace dialog window.

Fixing topological defects in the cortical surface

There are various causes of topologic defects in the reconstructed cortical surface. Certain defects are the result of typical anatomy (e.g. white matter in the tail of the hippocampus, the fornix, the occipital horn of the lateral ventricle). Others result from "errors" in the white matter segmentation. The topological defects are fixed by manually editing the **wm** (white matter) volume. The defects can be fixed by either erasing white matter or filling in white matter. The correct fix will depend on the specific defect. Some examples of defects and how to fix them are below.

In **medit**, read in the **T1** volume into the second volume:

Enter "**T1**" into the field **2nd imagedir**
Press **READ** to read in the **T1** volume into the second volume
To change volumes displayed, use the **COMPARE** button This will allow you to compare the segmented volume (**wm**) with the original MRI volume (**T1**) to determine how the defect needs to be fixed.

Select a point (**LEFT-CLICK**) on the cortical surface (**surfer** display window) near the topological defect.

1. Save the location of the point with **SEND** in the **surfer** window
2. Go to the point in the volume with **GOTO PNT** in the **medit** window

Some other helpful **medit** commands:
Fill (set voxel value to 255) – **MIDDLE-CLICK**
Erase (set voxel value to 1) – **RIGHT-CLICK**
Increase/Decrease the brush size for filling/erasing – **rad** field
For small defects use **rad** = 0, For most others **rad** = 1 is good.
Use 3D brush (edits out of visible plane) – **3Dbrush** radiobutton
Use care with the 3D brush because you are editing pixels you can't see
Save volume after changes are made – **SAVEIMG**
Toggle surface overlay – **surface** radiobutton

Some general hints

If the wormhole is not clearly visible at first in the initial slice view, change the slice plane before moving the cursor; it is often possible to see it more clearly this way.

To see black edits: contrast ($fsquash) = 200
midpoint ($fthresh) = 0.01

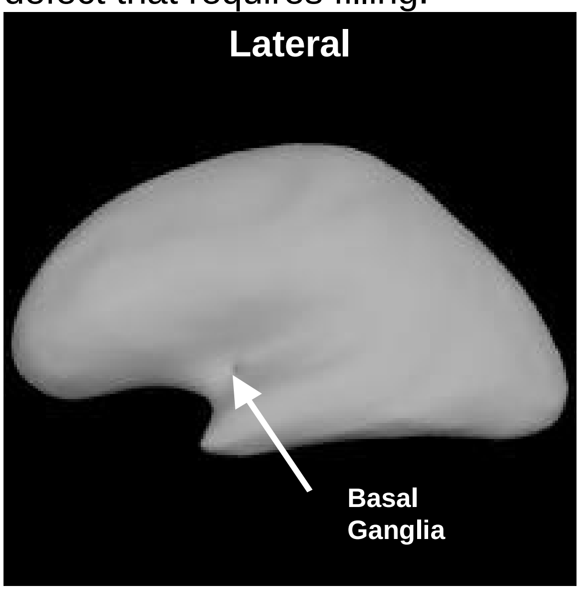
When selecting vertices on a wormhole, you may have to try several times until you get a vertex that has an edge in the wormhole itself. This is because (1) the body of the wormhole may be partly under the surface, and (2) clicking only selects the vertex nearest the viewer.

Touchup extensive edits in another plane to neaten the inevitably ragged edges due to the difficulty of editing the same location in successive slices.

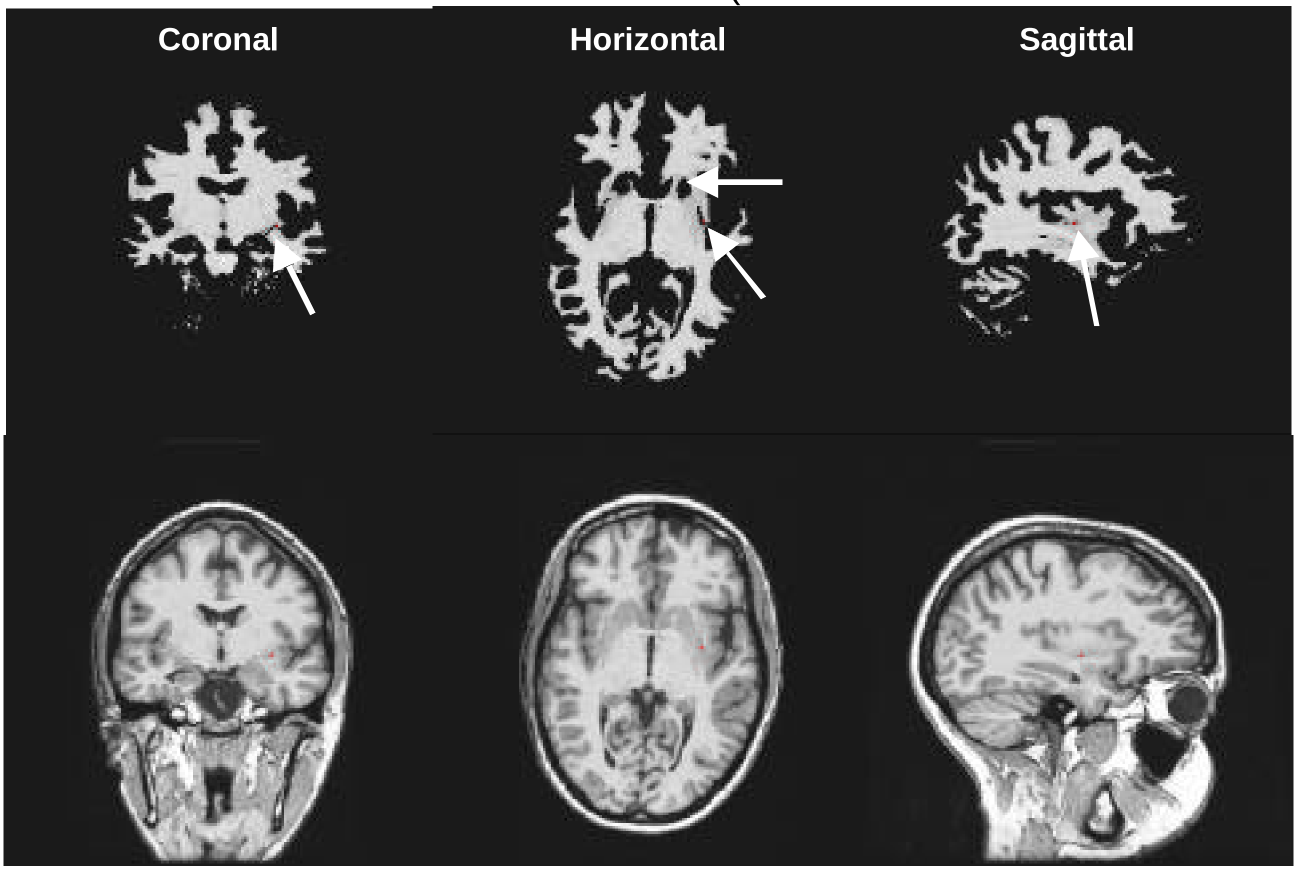
The following examples of defects arise from the typical anatomy. The procedure used to fix these defects will be similar to fixing other defects that arise.

Basal ganglia (caudate, putamen)

Results in a “hole” that links the lateral and medial surface. This is an example of a defect that requires filling.



The cause for the defect can be seen below(**wm** volume above and **T1** volume below)

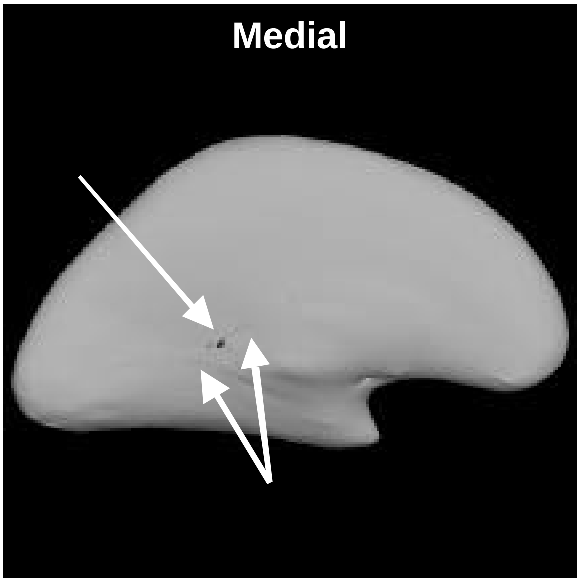


After filling:

This requires fixing in numerous slices. This particular defect is best fixed in the sagittal view. (Note: other kinds of defects may be best seen/fixed in another viewing orientation.)

Central defects on the medial surface:

There are two anatomic causes for most of the defects (numerous small handles) in the middle of the medial surface: 1) white matter by the tail of the hippocampus and 2) the fornix.

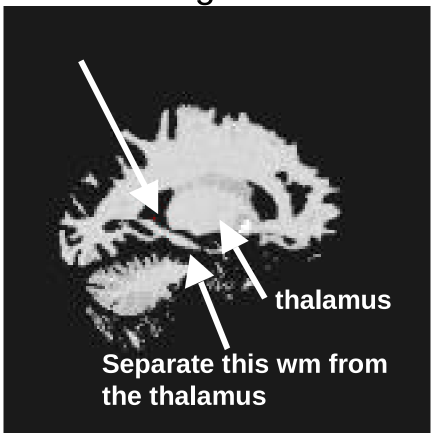


In both cases, we fix the topological defects by erasing the causative structure. Since we are only interested in the a representation of the cortical surface, loss of these structures does not overly concern us.

White matter by the tail of the hippocampus.

This defect is best viewed in the sagittal view

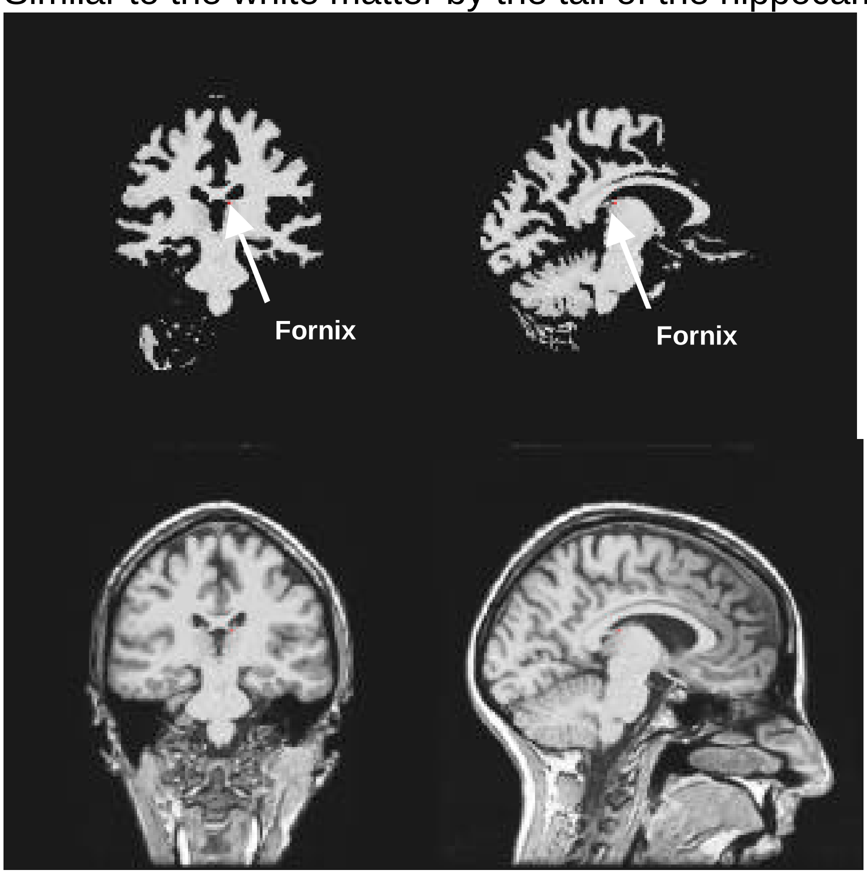
After erasing



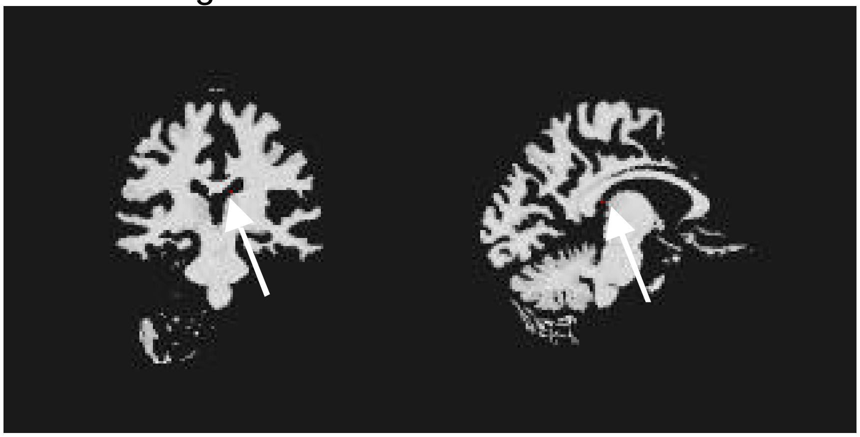
Also be sure that the white matter near the parahippocampus is separated from the thalamus (i.e. no residual hippocampus connecting the two areas indicated by the two lower arrows).

Fornix

Similar to the white matter by the tail of the hippocampus, the fornix should be erased.

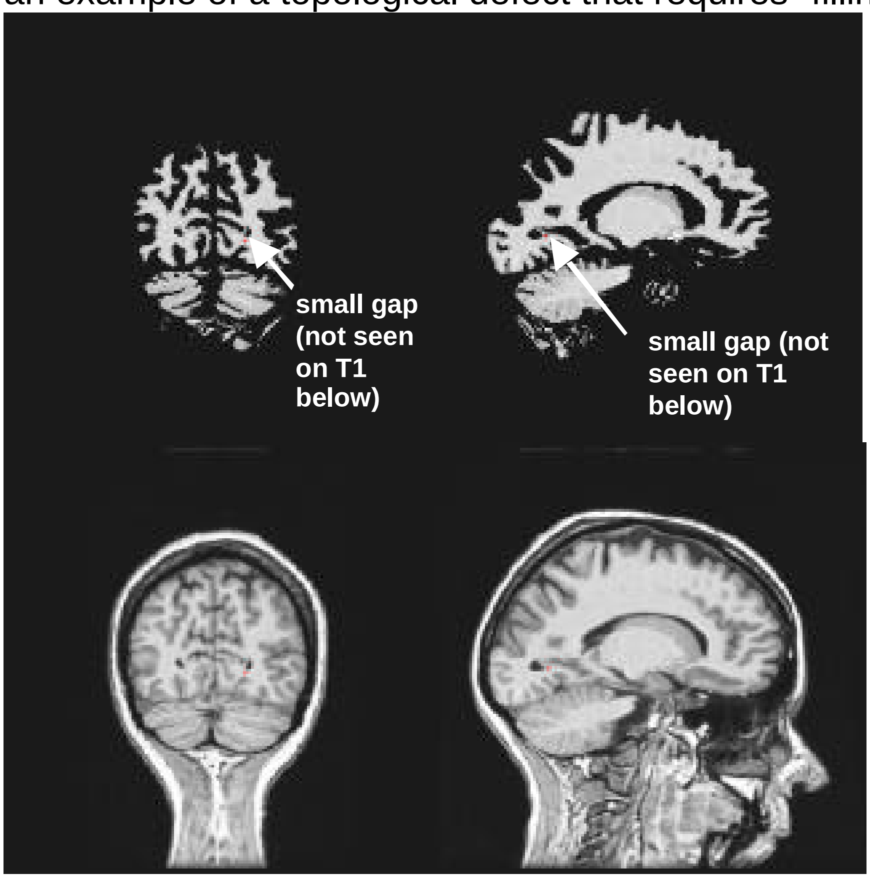


After erasing the fornix:

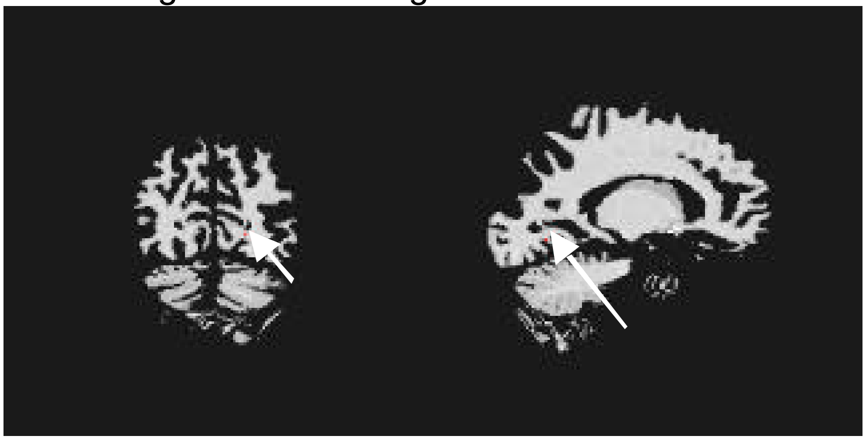


Occipital horn of the lateral ventricle

The last example of normal anatomy causing topological defects is the occipital horn of the lateral ventricle. The white matter is very thin in between the calcarine fissure and the occipital horn of the lateral ventricle. Often this region of white matter (due to partial volume effects) has a very low intensity and is lost during segmentation. This is an example of a topological defect that requires “filling” in the lost white matter.

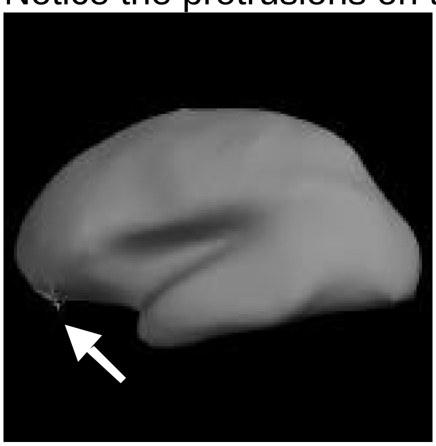


After filling in the “missing” white matter:

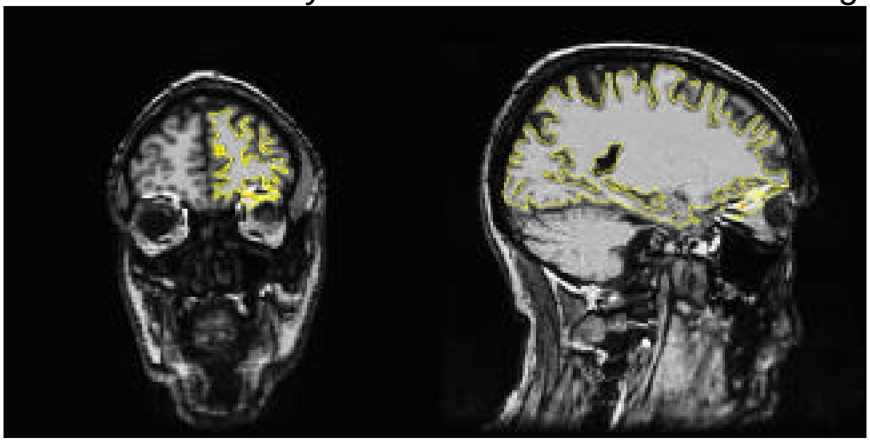


Escape into orbits

There is very little CSF between orbitofrontal cortex and the orbits, and the white matter fill may escape into the tops of the orbits, which sometimes remain in the 'brain' image set even after removing the skull. To fix this defect remove the remnants of the orbit.
Notice the protrusions on the surface.



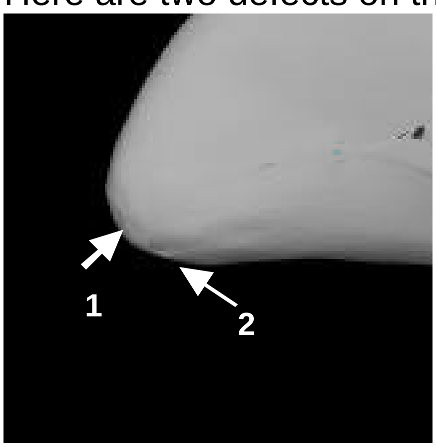
The defect is easily seen in the volume with the orig surface overlaid.



Some other examples of topological defects and how to fix them.

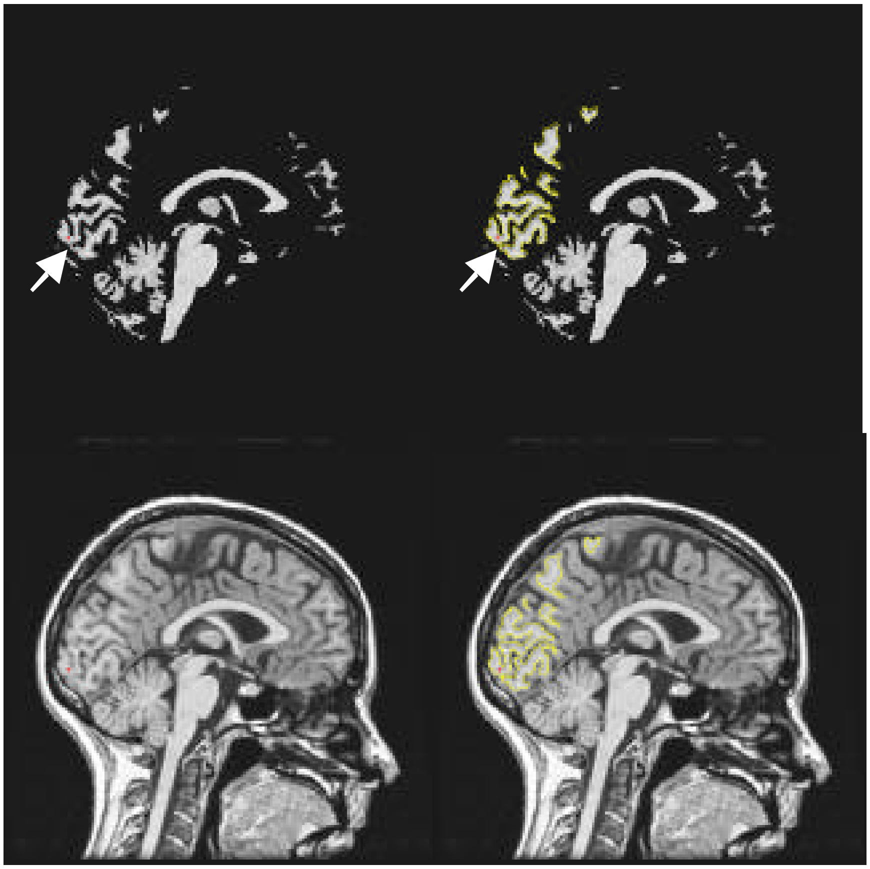
For these other defects (that are not consistent across subjects), it is often helpful to examine the surface (<hemisphere>.orig) overlaid onto the **wm** or **T1** volume to determine the cause of the defect. For very small defects, it may be difficult to see the defect with surface overlay. The surface overlay is toggled with the **surface** radiobutton in the **medit** window.

Here are two defects on the cortical surface:



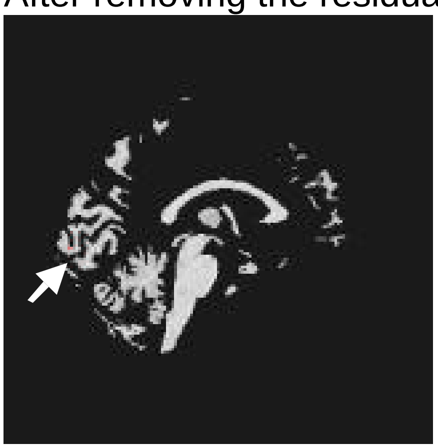
First, select a point on the surface near defect 1 and save the location with the **SEND** button in the **surfer** window. Then go to the location in the volume using the **GOTO PNT** button in the **medit** window. Often this will not bring you immediately to the defect, but by examining the nearby slices, you can usually quickly find the problem.

For defect 1:



Here, the defect is caused by too much gray matter being left by the segmentation. This essentially connects two adjacent strands of white matter, creating a “bridge” or “handle” on the surface.

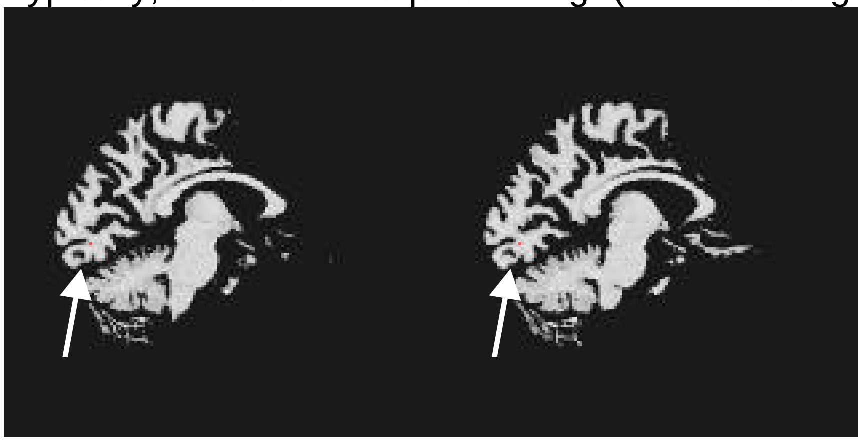
After removing the residual gray matter:



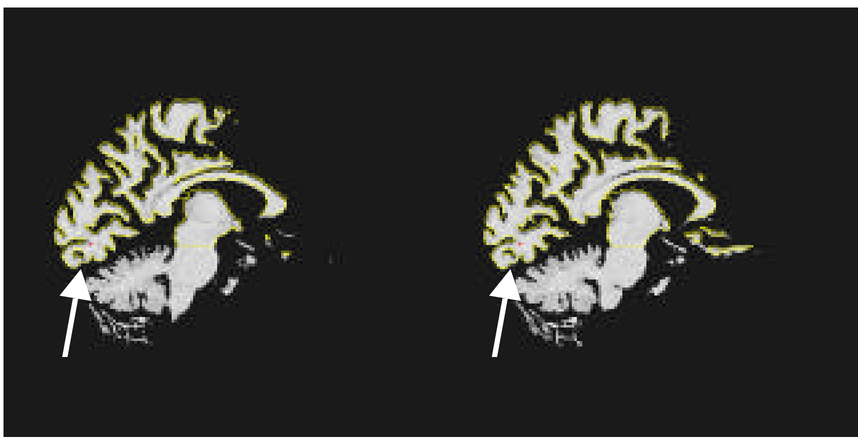
For defect 2, there is a similar situation. There is gray matter that is creating a bridge between two strands of white matter.

Shown below are two adjacent slices. The defect can be seen in both slices.

Typically, defects will require “fixing” (either erasing or filling) in more than one slice.



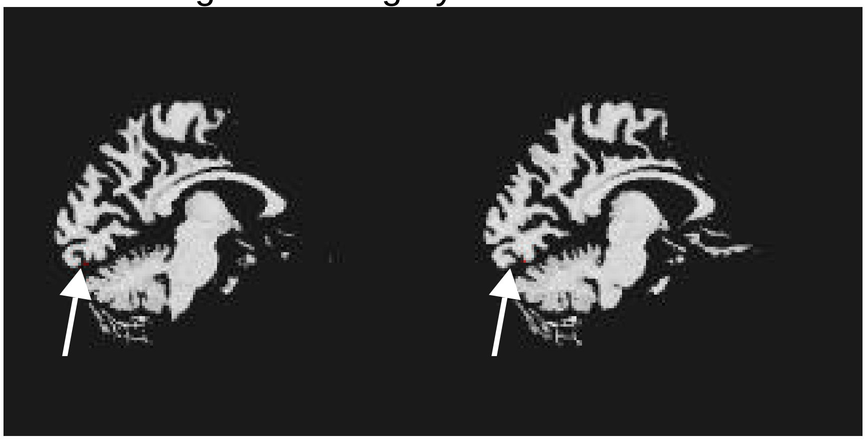
White matter volume



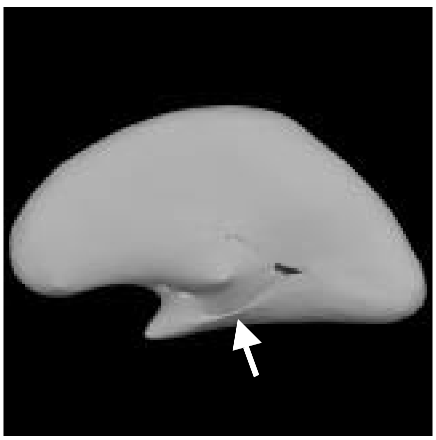
White matter volume with surface

T1 volume

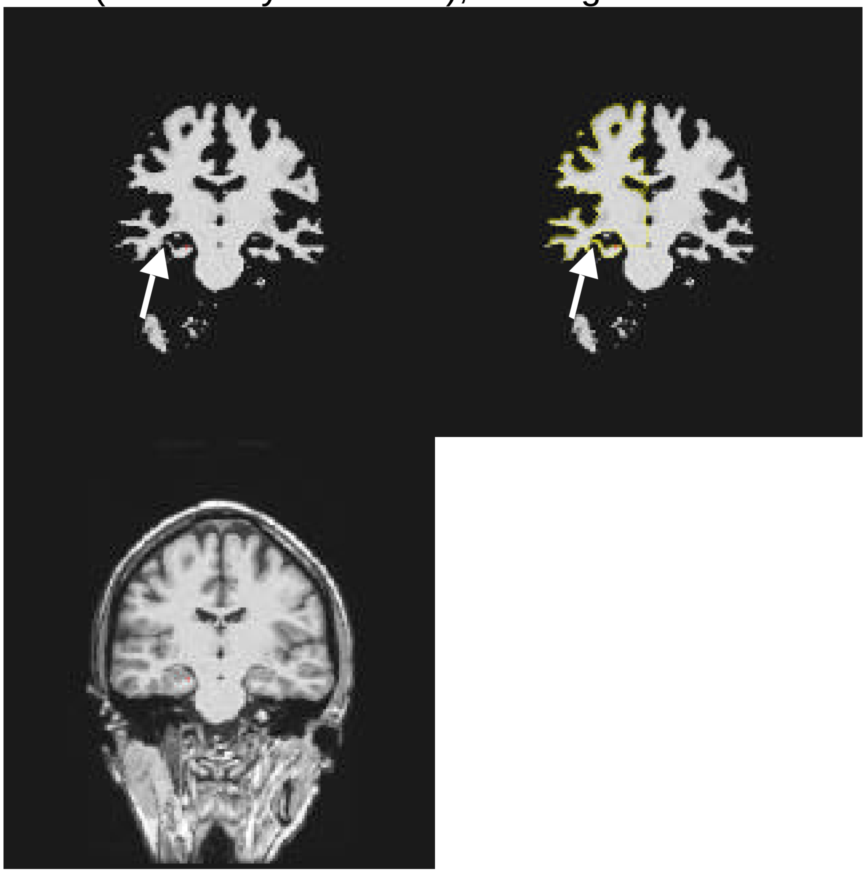
After erasing the extra gray matter:



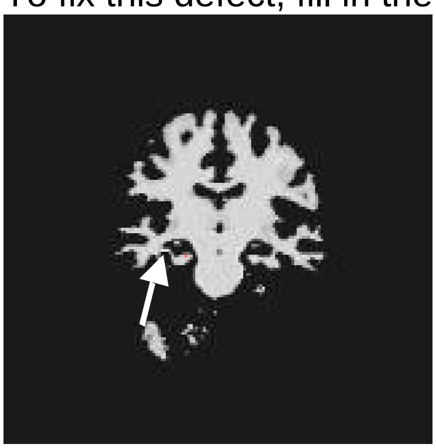
One final example is shown here. This handle looks quite similar to the two previous defects.



This defect arises because the white matter near the hippocampus is very thin. In this case (marked by the arrow), the segmentation has lost some of the white matter.



To fix this defect, fill in the missing white matter voxels:



After fixing defects in the **wm** volume, **Create Surface** under the **Tools** menu is repeated.

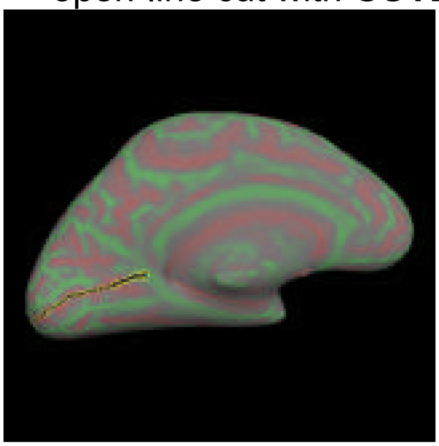
Cutting the Occiput Surface

The occiput patch represents the occipital pole with a cut down the calcarine fissure. The procedure is as follows:

1. make calcarine sulcus relaxation cut

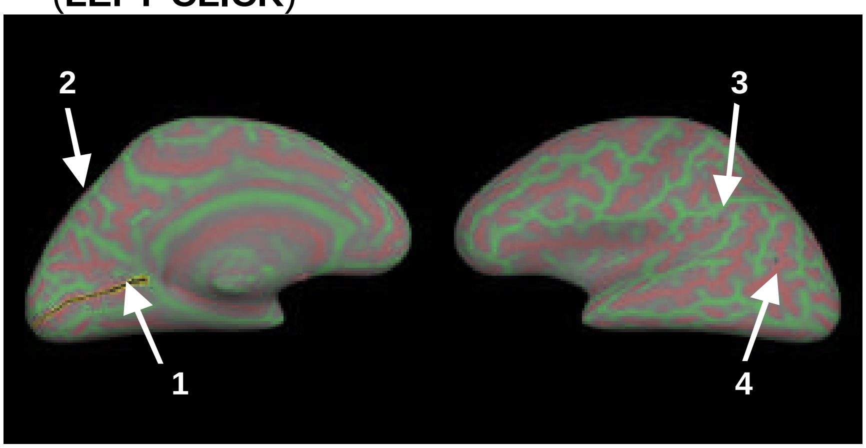
    select points (**LEFT-CLICK**) along the calcarine fissure beginning at the occipital pole and moving anterior.

    open-line cut with **CUTLINE**



2. specify cutting plane

    select three points (labeled 1, 2, and 3 below) to define the cutting plane, and a fourth point (labeled 4 below) to specify which portion of the surface to keep (**LEFT-CLICK**)



3. make the planar cut

   CUTPLANE button

4. save patch

   specify the patch name in the **patch** field and press **WRITE**

   The default names for the occipit cortical surfaces are:

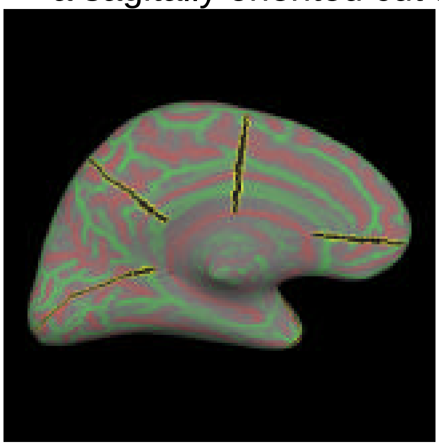
   rh.full.patch.3d

   lh.full.patch.3d

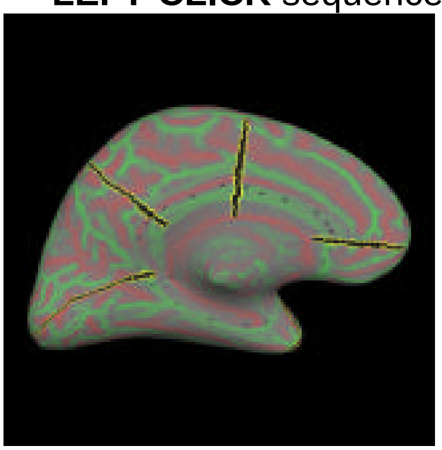
Cutting the Full Surface

The full surface represents an entire hemisphere without the mid-brain (middle of the medial surface).

1. Make 5 relaxation cuts.
   For each cut, select the points (**LEFT-CLICK**) and then press **LINE**.
   a cut down the calcarine (same cut as for the occiput surface)
   three equally spaced radial cuts on the medial surface
   a sagitally oriented cut around the temporal pole

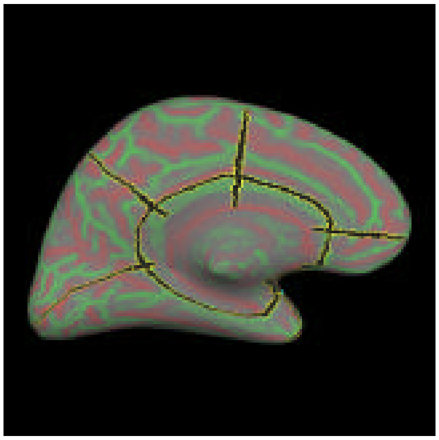


(1) encircle midline region (corpus callosum and mid-brain structures) to remove:
   LEFT-CLICK sequence



(2) make closed-line cut

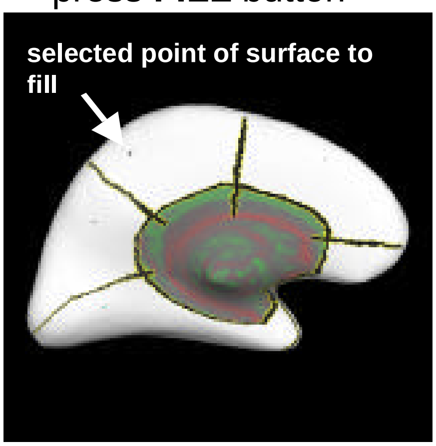
CUTAREA button

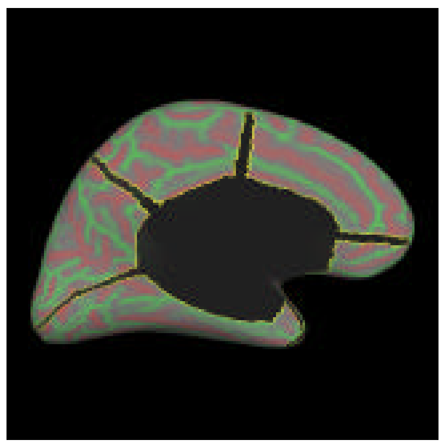


(3) mark and fill region to save

single **LEFT-CLICK** in region to save

press **FILL** button





(8) save patch

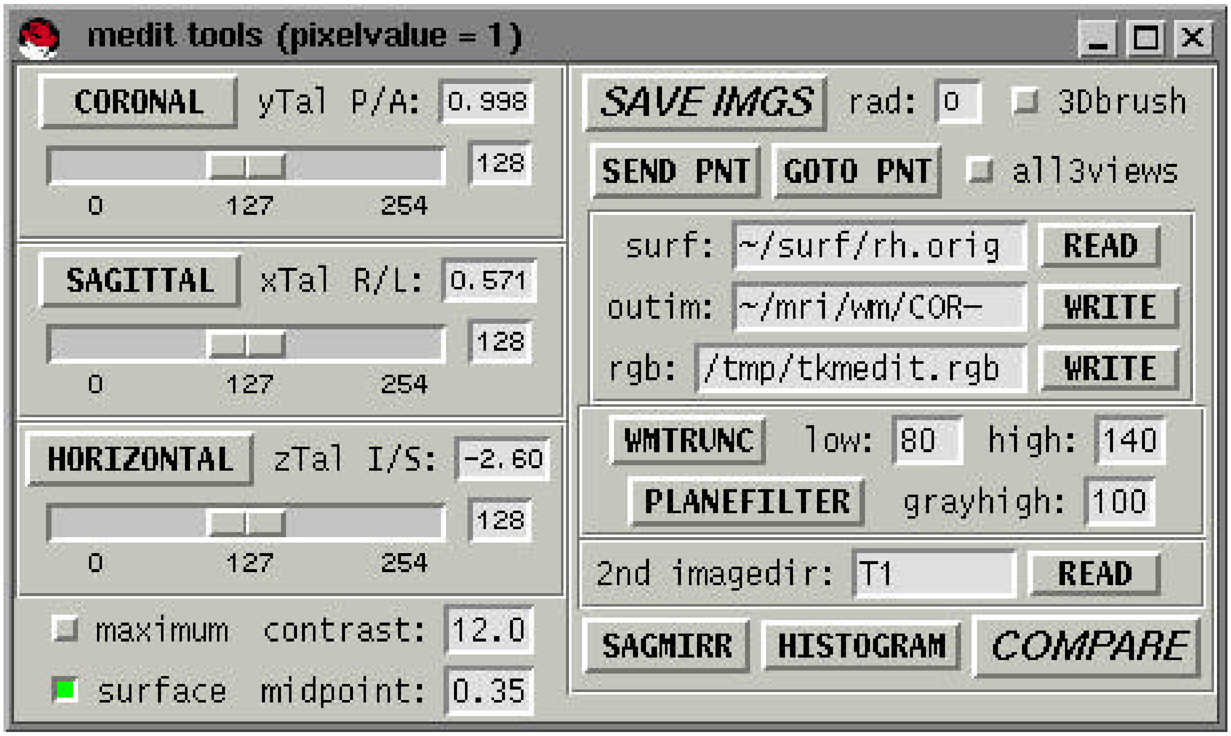
specify the patch name in the **patch** field and press **WRITE**

The default names for the full cortical surfaces with the midline removed are:

rh.full.patch.3d

lh.full.patch.3d

Medit



Medit reads one or more series of images and displays them in coronal, sagittal, horizontal, or all 3 planes. The images in the first (edit) buffer can be edited and saved. Images read into the second buffer cannot be edited. Edits applied while viewing the second buffer affect the first buffer (useful when edit buffer has segmented white matter and second buffer hasoriginal full T1 images).

Medit can also read in a surface and display the intersection of the surface with the current slice.

Viewing orientation

CORONAL button: switch to coronal view
slider: change coronal slice
field: enter coronal slice number
SAGITTAL button: switch to sagittal view
slider: change sagittal slice
field: enter sagittal slice number
HORIZONTAL button: switch to horizontal view
slider: change horizontal slice
field: enter horizontal slice number
all3views radiobutton: displays all three views with a maximum intensity projection

Display parameters

maximum radiobutton: maximum intensity projection
surface radiobutton: toggle surface overlay
contrast field: contrast
midpoint field: midpoint of intensity range
SAGMIRROR button: Mirror volume about sagittal plane (left-right mirror)

Read/Write

SAVE IMGS button: save images (COR files) in volume specified by **outim**
surf field: surface for overlay
outim field: volume to save with **SAVE IMGS** button
rgb field: name of rgb to save currently viewed slice

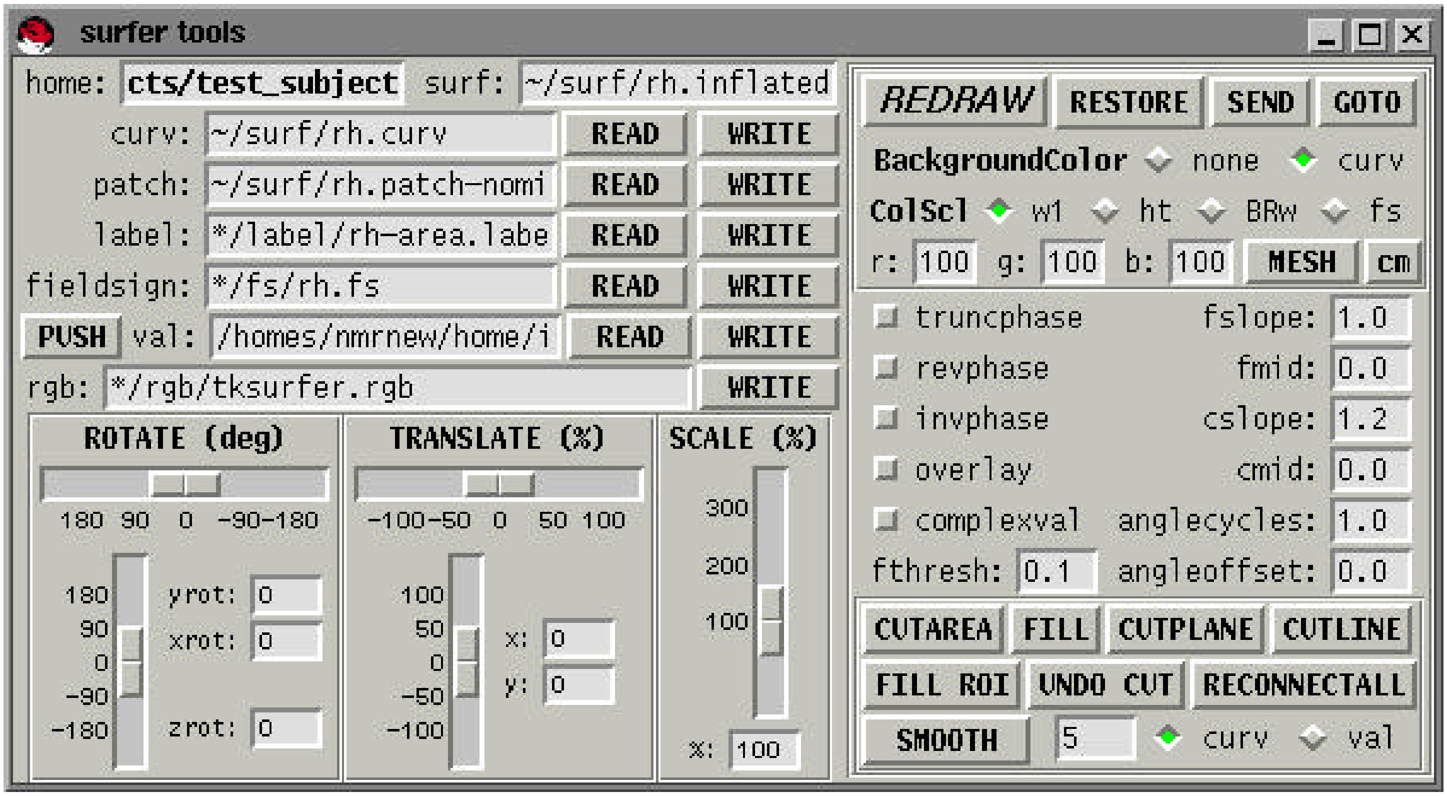
Editing

MIDDLE-CLICK	draw (set intensity to 255)
RIGHT-CLICK	erase (set intensity to 1)
rad field:	radius of editing brush
3Dbrush radiobutton:	toggle 3D/2D brush (2D brush only edits viewed slice)
SEND PNT button	save location of cursor for **surfer**
GOTO PNT button	goto location of point saved by **surfer**
2nd imagedir field:	name of 2nd volume to read in
READ button:	read in 2nd volume
COMPARE button:	toggle between two volumes (must first read in **2nd imagedir**)

Unsupported

WMTRUNC button
PLANEFILTER button
low field
high field
grayhigh field
HISTOGRAM button

Surfer



Surfer reads a surface file, and optionally, functional data, and then renders a view of the surface.

home field:	subject directory
surf field:	surface that is displayed. To view a different surface, enter the surface and press **RETURN**

READ/WRITE

curv field:	cortical curvature file
READ	read curvature file
WRITE	write curvature file
patch field:	cortical patch file
READ	read cortical patch
WRITE	write cortical patch
label field:	not supported - future functionality
READ	
WRITE	
fieldsign field:	Not supported. Future functionality
READ	
WRITE	
val field:	val file – same format as *.w files
READ	read val file
WRITE	write val file
rgb field:	rgb file
WRITE	write rgb file of current surfer window.

Viewing orientation

REDRAW	redraw screen – redraw is not always automatic
RESTORE	restore original viewing orientation
ROTATE	
slider (horizontal)	rotate view around vertical axis
slider (vertical)	rotate view around left-right axis
yrot field	rotate around y axis
xrot field	rotate around x axis
zrot field	rotate around z axis
TRANSLATE	
slider (horizontal)	translate left-right
slider (vertical)	translate up-down
x field	translate in x axis
y field	translate in y axis
SCALE	
slider (vertical)	scale larger-smaller
% field	scale in percent

Surface Display

BackgroundColor	
none radiobutton	do not display curvature information (uniformly gray brain)
curv radiobutton	display curvature information (specified in **curv** field)
MESH button	display actual vertices and mesh
r, g, b fields	set color of mesh
cm button	display scale bars (size scale bar = 1cm)

Functional Display

ColScl	color scale of functional data
w1 radiobutton	not supported - future functionality
ht radiobutton	yellow/red for positive values, blue/green for negative values
BRw radiobutton	not supported - future functionality
fs radiobutton	not supported - future functionality
truncphase	only display positive values (red/yellow).
revphase	not supported - future functionality
invphase	reverses positive and negative values
overlay	displays functional data when checked
complexval	not supported - future functionality
fthresh field	statistical threshold (for values below the threshold, the underlying curvature is displayed). This is equivalent to **Stat Hard Thresh** in **Setup Rendering Parameters**
fslope field	color slope from **fmid** (red/blue) to the maximum color (yellow/green). Maximum color (yellow, green) represents a statistical value of **fmid** + 1/ **fslope**. This is equivalent to **Stat Color Contrast**.
fmid field	statistical value for full red/blue. Equivalent to **Stat Color Midpoint**
cslope	contrast of curvature
cmid	mid-point of curvature colorscale
anglecycles	not supported - future functionality
angleoffset	not supported - future functionality
SMOOTH button	smooth curvature or values (functional data)
field	number of smooth iterations
curv	smooth curvature
val	smooth values (functional data)

Cutting Surface

CUTAREA	cuts a close line from selected **LEFT-CLICKs**
FILL	fills in a region from a selected **LEFT-CLICK** defined by cuts
CUTPLANE	planar cut. First 3 LEFT-CLICKs define plane, 4th **LEFT–CLICK** defines surface to keep
CUTLINE	cuts an open line from selected **LEFT-CLICKs**
FILL ROI	fills in a region defined by statistical values
UNDO CUT	undoes the most recent cut
RECONNECTALL	undoes all cuts

Example of a Typical Surface Reconstruction

This is a menu step by menu step example of a typical reconstruction including cutting flattened cortical patches. For more detail refer to the specific sections in the manual.

Under **File**, select **New Subject**
Under **Tools**, select **Setup Structural Scans**
Under **Tools**, select **Process Volume**
Under **Tools**, select **Create Surface**
Under **Edit**, select **Edit Segmentation**
Manually fix topologic defects using medit and surfer
Repeat steps 5-7 until all defects are fixed
Under **Tools**, select **Make Final Surface**
Under **Edit**, select **Make Full Surface Cuts**
Make and save full patch
Under **Edit**, select **Make Occip Surface Cuts**
Make and save occipit patch
Under **Tools**, select **Flatten Surface** for each patch that was cut
Under **Tools**, select **Sphere Surface**

Appendix 1: tcl/tk license

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

RESTRICTED RIGHTS: Use, duplication or disclosure by the government is subject to the restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause as DFARS 252.227-7013 and FAR 52.227-19.

Appendix 2: Tix license

Appendix 3: Mesa license

This file describes the rights you have for the "Mesa" libraries libMesaGL.so and libMesaGLU.so. This library and the rights described herein, do NOT apply to the whole of, or any other part of the FreeSurfer.

Mesa, a graphics library with an OpenGL-like API, is used by FreeSurfer to render graphics. It is provided under the terms of the Free Software Foundation's GNU General Public License described below. This means that you are free to make and distribute copies of the libMesaGL.so and libMesaGLU.so libraries provided along with FreeSurfer. It also means that CorTechs will provide you with source code for Mesa if you so request. Mesa is also available directly from the internet at the following URL: http://www.ssec.wisc.edu/~brianp/Mesa.html Option 6(b) is offered to users who would like to modify the library and relink.

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy. For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!