Appendix D Using DTM

Overview

Users can develop their own applications that send data to NCSA Collage using DTM (Data Transfer Mechanism). This description assumes the reader has a programming level understanding of the DTM library. Further information about DTM and supporting source code may be found on the NCSA anonymous ftp server `ftp.ncsa.uiuc.edu`.

The source distribution of collage comes with the DTM extensions. Some of the DTM routines used to communicate with NCSA Collage will be found in the source. Additionally, if you are interested in communications, you can examine the module **`net.c`**. This module is highly portable and has been added to several NCSA data analysis tools easily so as to allow them to communicate with NCSA Collage.

DTM and Clients

NCSA Collage uses the DTM as its protocol for communicating over the network. In a collaborative session, NCSA Collage communicates with other NCSA Collage clients via the NCSA Collage server. Other clients such as data analysis tools, simulation software, or graphics applications may send data to the NCSA Collage session to be viewed by all participants.

A collage client has one DTM input port and one DTM output port. These ports may be specified at the start of NCSA Collage with the command line arguments:

`collage -DTMIN` *inport* **`-DTMOUT`** *host*`:`*outport*

When NCSA Collage connects to a remote process via DTM it will handshake with an SRV message class. The SRV message class is used to establish a connection within a collaborative session. The message relayed is, “I’m connecting to you, here is my userID and my inport.” When the NCSA Collage server receives this connection, it will be on its global inport. The server sets up a separate inport and outport for each client. The return message from the server will be, “OK, change your outport to this address.” The new address passed back will be a special inport on the server used exclusively for reading incoming messages from this client. The client will then close its outport to the server and reconnect to the new outport that the NCSA Collage server has given it. The same initial connection sent by the client is sent to the NCSA Collage server. At this point, connection is established with the NCSA Collage server. This arbitration is only necessary with a client that will be participating as a collaborator in the NCSA Collage session. For a client that will just be sending data, this arbitration is not necessary.

DTM Classes

A client communicating with the NCSA Collage server may receive any DTM class, but any class that is not recognized is ignored. DTM classes that NCSA Collage currently supports are listed below.

General DTM classes that NCSA Collage supports are:

- RIS8 for raster images
- SDS for scientific data sets (n dimensional arrays of any type)
- PAL for 8 bit color palettes
- MSG for sending messages used in the Chat Box

DTM classes that are specific to NCSA Collage are:

- COL for selections and doodle annotations (currently only receives in NCSA Collage 1.2)
- DOL for selections and doodle annotations (new class for NCSA Collage 1.2)
- SRV for establishing and closing connections
- ANIM for animation commands
- TXT for sending text editing messages

Each of these classes has the additional field of a userID of the sender set in the form of

XXXsetID("usernamehere")

Associations

When sending RIS8 and SDS, usually a palette is associated with the image. After the RIS8 or SDS has been sent, the palette is sent with an associated tag in the header. The call

COLsetAssoc(header, "name of data associated")

will let the receiver know that this palette should be used when displaying the image. The associated tag is also used when a data set and an image are related. This will cause selection in NCSA Collage to appear on both the image and dataset.

Animations

Animations are sent as 2D SDS. The DTM header is marked "to be treated as an animation". This is done with the call

ANIMmarkAnimation(header)

This will cause NCSA Collage to put the SDS in an animation window rather than a single frame window. Additional frames with the same name and animation mark will be appended to the animation sequence in the order that they were received.