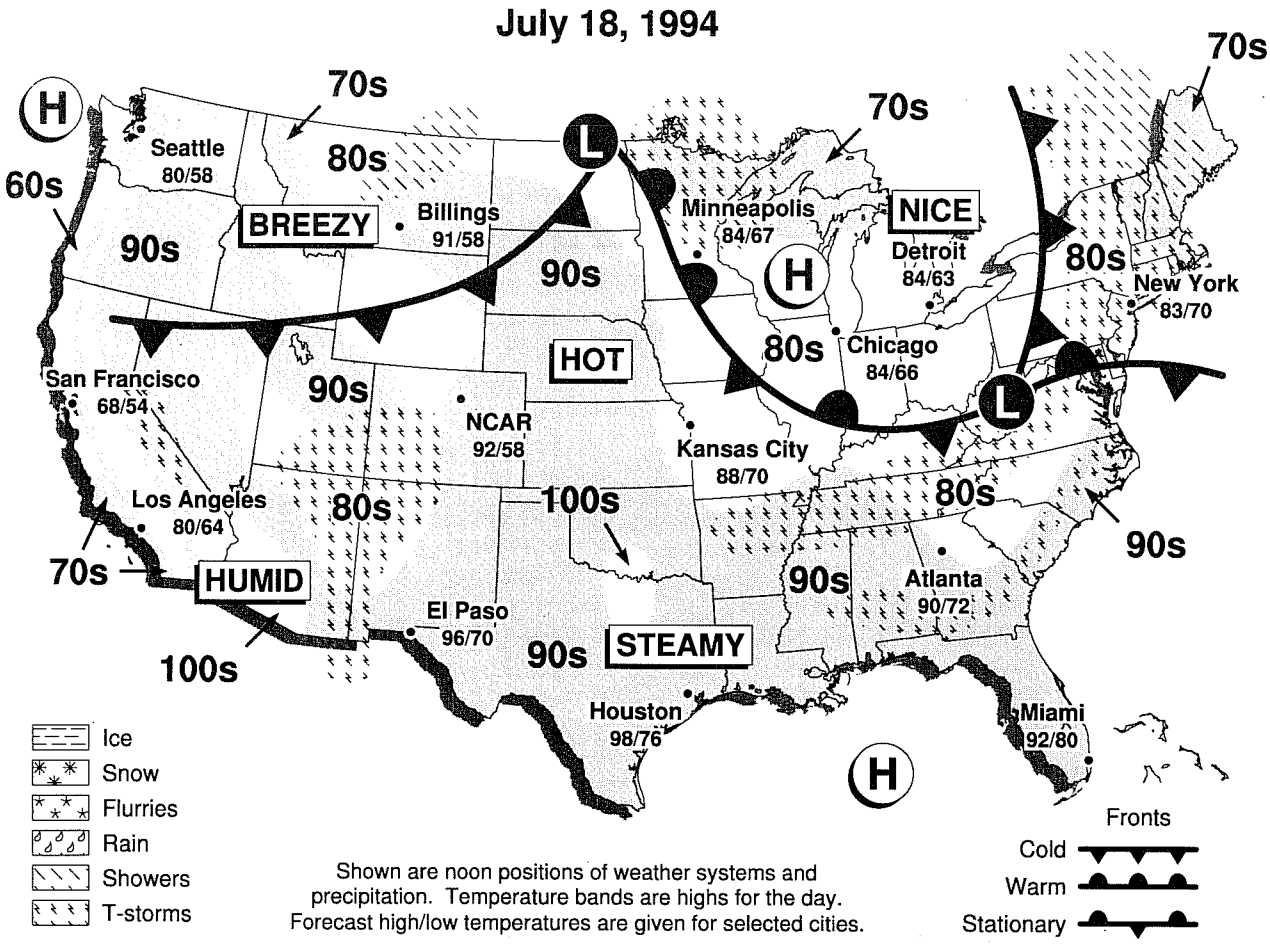
INTRODUCTION TO NCAR GRAPHICS VERSION 4.0

R. L. Lackman
National Center for Atmospheric Research
Boulder, Colorado USA

Summary: NCAR Graphics continues to evolve into an integrated environment for performing data processing and scientific visualization. Version 4.0 now contains multiple interfaces to accommodate diverse user skill levels and diverse application needs. Moreover, many related tools are bundled in with the distribution to perform functions such as metafile translation, viewing, editing, animation, format conversion, zooming, and output. Software additions to Version 4.0 include an integrated set of object oriented utilities, a powerful data language, and a prototype graphical user interface.

Fifth Workshop on Meteorological Operational Systems
ECMWF, Shinfield Park, Reading/Berks., United Kingdom
13-17 November 1995



1. NCAR Graphics HISTORY

Since NCAR's inception as a national research center in the early 1960s, graphics has played an important role in supercomputing for the atmospheric sciences. Early on, the Computing Facility of NCAR, now the Scientific Computing Division (SCD), generated a low-level collection of routines to draw lines, curves, and dashed-line patterns called the NCAR System Plot Package. By the 1970s, NCAR scientists and SCD staff had developed several higher level graphics utilities which were integrated into a graphics package that could be distributed throughout the scientific community.

In the early 1980s, graphics and programming language standards had gained wide acceptance, so NCAR Graphics utilities were converted to FORTRAN 77, the metafile was based on a private encoding of the Computer Graphics Metafile (CGM) standard, and utilities were modified to depend on the Graphical Kernel System (GKS) standard.

1.1 New functionality emphasized in the 1990s

Functionality enhancements of the 90s have included:

* Interactivity
* Animation
* Expanded interoperability
* High quality fonts
* A data language
* Multiple interfaces
* An expanded toolset
* Hypertext documentation

Major additions to Version 4.0 include an object-oriented (OO) interface with C and Fortran bindings, an interactive data processing and data visualization language called the NCAR Command Language (NCL), and a first draft of a Graphical User Interface (GUI).

2. GOALS FOR VERSION 4

2.1 Project Overview

The goal of the NCAR Interactive project has been to create, from existing utilities, a fully integrated scientific visualization environment. To be fully integrated, this environment had to provide a means of reading and writing data, a means of manipulating that data, and a means of visually analyzing the data interactively. It had to be easy to use, work efficiently, and be user-extensible. Furthermore, this environment had to support users with a wide range of skill levels: from users with extensive programming and technical skills to users who have not developed these skills.

3. FOUR SUPPORTED INTERFACES

There are now four major interfaces to the NCAR Graphics package:

- **Low Level Utilities (LLUs)** - This library contains the traditional Fortran and C subroutine library that has hitherto been known as NCAR Graphics.
- **High Level Utilities (HLUs)** - The HLUs are an object-oriented programming interface library with Fortran and C bindings.

- **The NCAR Command Language (NCL)** - NCL is a text-based data manipulation and graph specification language interpreter. It uses an internal netCDF data model.
- **A Graphical User Interface (GUI)** - At this point the GUI is only a first draft of what we plan to provide in the way of a professional, full-functionality, "point and click" interface.

Figure 1 shows the functional relationship of the various components of the new NCAR Graphics system. The new High Level Utilities are built on top of the subroutine libraries. The NCAR Command Language provides the data I/O interface, translates all GUI button clicks into actions, and calls the High Level Utilities to create visualizations.

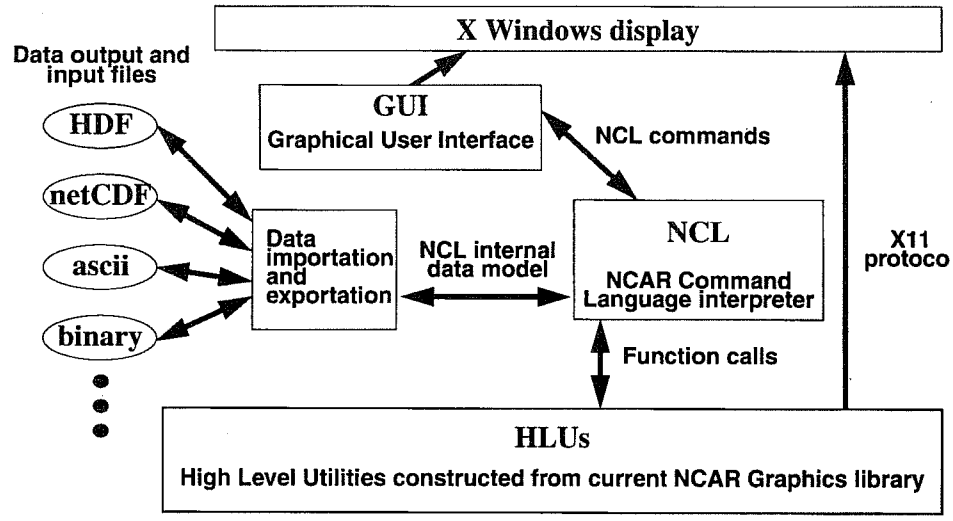


Figure 1. NCAR Graphics Version 4.0 Overview.

3.1 Low Level Utilities (LLUs)

Version 3.2 of NCAR Graphics, with enhancements, is a set of about 500 low level graphics routines with Fortran and C bindings. In Version 4.0, these routines have been relabeled the Low Level Utilities (LLUs). The Version 3.2 Fundamentals manual, which tells how to use this library, has been renamed "Low Level Utility Fundamentals." This subroutine library interface will continue to be supported to ensure that the many current NCAR Graphics codes spread throughout the research community will continue to run. However, when users generate new codes they may want to use one of the new interfaces: the High Level Utilities (HLUs), NCAR Command Language (NCL), or Graphical User Interface (GUI).

3.2 High Level Utilities (HLUs)

The new object oriented utilities are also a programmatic library interface with both C and Fortran bindings. Although the HLU interface makes use of many of the existing NCAR Graphics low level subroutines, it has a number of advantages over the LLU interface. First, it hides a great deal of the underlying low-level implementation details from the application programmer. Second, the order dependence of low level sub-routine calls is eliminated. One simply creates an object, such as a ContourPlot, and sets its resources to configure the output.

However, the main objective in creating the HLU interface was to extend NCAR Graphics to support the demands of interactive graphics, primarily giving users the ability to change features of the output frame efficiently and quickly. Other features, like being able to retrieve information about the current state of an HLU and retrieving data coordinate information, are built into the HLUs.

3.2.1 *HLU Objects*

Important HLU objects include data objects for controlling data input, workstation objects for controlling data output, view (graphical) objects for display, transformation objects to control data transformations, a plot-manager object to control overlays, and an annotation object to control annotations. Figure 2 shows the kinds of objects which are typically created to build an HLU application.

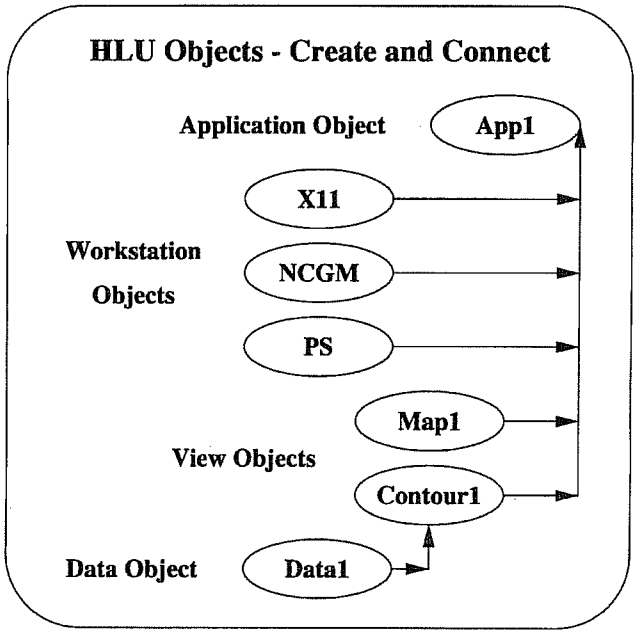


Figure 2. Typical HLU Objects in a Program.

View (graphical) objects included in the Version 4.0 library are:

- **TextItem** - A generic text placement object that supports filled fonts.
- **LabelBar** - A generic filled label bar which is used by the Contour object.
- **Legend** - A generic legend that displays both symbols and lines with text. It can be used by the XyPlot or Contour objects.
- **Title** - It positions titles for plots.
- **TickMark** - It draws tick marks around plots in 5 transformation styles (LOG, LINEAR, IRREGULAR, GEOGRAPHIC, and TIME).
- **XyPlot** - It draws curves and symbols. It includes Title, TickMark, and Legend objects as composites.
- **ContourPlot** - Draws filled contours with and without maps. It incorporates Legend, LabelBar, TickMark, and Title objects as composites.
- **MapPlot** - It generates maps in any of the 10 standard NCAR Graphics transformations.

Figure 3 shows a monthly mean sea surface temperature (Shea et al., 1990) as labeled contour lines, masked by the land masses, and overlaid with ocean surface ice. This plot was created using three plot objects: a MapPlot object, and two ContourPlot objects. One ContourPlot object does a raster contour of ice versus no ice. The other ContourPlot object does line contours of the sea surface temperature.

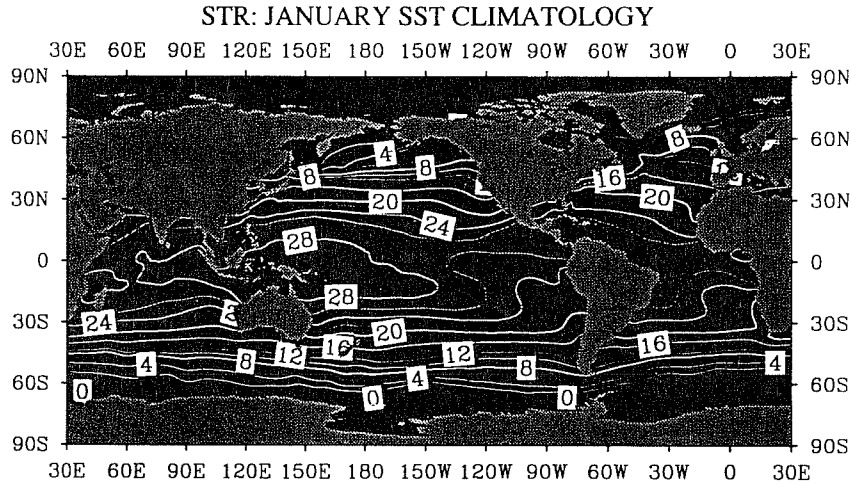


Figure 3. An Example HLU Graphic.

3.3 **The NCAR Command Language (NCL)**

The heart of the new interactive system is the data manipulation and graph specification language called the NCAR Command Language (NCL). NCL provides easy and intuitive access to datasets and allows users to explore and process their data prior to visualization. Since datasets often come in a variety of data formats, grid sizes, grid resolutions, and units, very different datasets often need to be combined, compared, and used at the same time. Currently, specialized applications must be developed to read individual datasets and transform them into a form that is compatible with other datasets being used, as well as with the graphics package being used.

NCL allows different datasets, in different storage formats, to be imported into one uniform and consistent data manipulation environment. The primary data format used internally by NCL is the netCDF data format. netCDF is a network-transparent version of the NASA Common Data Form, which was developed and is being supported by the Unidata Project of the University Corporation for Atmospheric Research. NCL does not place any restrictions or conventions on the organization of input netCDF files. NCL is a complete programming language that provides flexibility and configurability. In NCL, the primary data type is the data file record. A data file record stores one or more variables, dimension information, coordinate variable information, and attribute information as one NCL object. A binary file can be input, then dimension names, variable names, attributes, and coordinate variables can be assigned to it using NCL language constructs. The resulting file record can then be written to any of the currently supported formats, including netCDF, without writing a single line of source code. NCL also provides the ability to create and manipulate the HLU graphical objects utilizing the same resources available through the HLU toolkit interface.

NCL's function set contains built-in data processing and mathematical functions. Users can extend NCL functions to implement custom data processing techniques and custom data ingestion.

NCL commands can be interactively interpreted and executed in line mode. In addition, an API for NCL is under development that provides GUI writers a uniform way to do data access, data manipulation, and visualizations from within their application using NCL scripts. This simplifies the implementation of application-specific GUIs by reducing the amount of design time spent on the visualization code and the data access code.

3.3.1 *Data Import/Export*

Data input/output is performed by NCL. Currently implemented input formats are netCDF, ASCII, binary, and HDF. The next format to be implemented will be GRIB.

NCL supports Fortran 90 style indexing. Thus, one can selectively access elements or ranges of elements in a multi-dimensional array.

3.3.2 *Coordinate Indexing*

NCL supports coordinate indexing as well as numeric indexing. Consider a four-dimensional temperature variable T. The following NCL script shows how one would extract all of those temperatures for the first time step (index 1), at the first level (index 2), between latitudes of 20N to 60N (index 3), and longitudes of 30W to 150E (index 4):

```
begin
file1 = addfile("temp.cdf","r")
Temp = file1->T(0,0,{20:60},{-30.:150})
end
```

The variable Temp will be two-dimensional with the number of latitude and longitude indices that fulfill the selection range. The curly braces denote coordinate subscripting.

3.3.3 *Visualization Blocks*

NCL has a convenient shorthand for creating and altering visualizations called the visualization block. Figure 4 shows a typical NCL visualization block which defines the XyPlot shown.

```
xy_plot = create "xy_plot" xyPlotClass wks
  "vpXF": .15
  "vpYF": .85
  "vpWidthF" : .7
  "vpHeightF" : .7
  "xyCoordData": field1
  "trYReverse" : "True"
  "xyYStyle"  : "irregular"
  "xyYIrregularPoints"  : (/1000.,850.,7
  "trYMaxF" : 1000.
  "trYMinF" : 100.
  "trXMaxF" : 40.
  "trXMinF" : -80.
  "tiMainString" : "Upper Air Forecasts"
  "tiMainFontHeightF" : 0.035
  "tiXAxisString" : "Temperature (Deg F)"
  "tiYAxisString" : "Pressure (mb)"
  "tiXAxisFontHeightF" : 0.02
  "tiYAxisFontHeightF" : 0.02
  "tiMainFont" : "helvetica-bold"
  "tiXAxisFont" : "helvetica-bold"
end create
draw(xy_plot)
frame(wks)
```

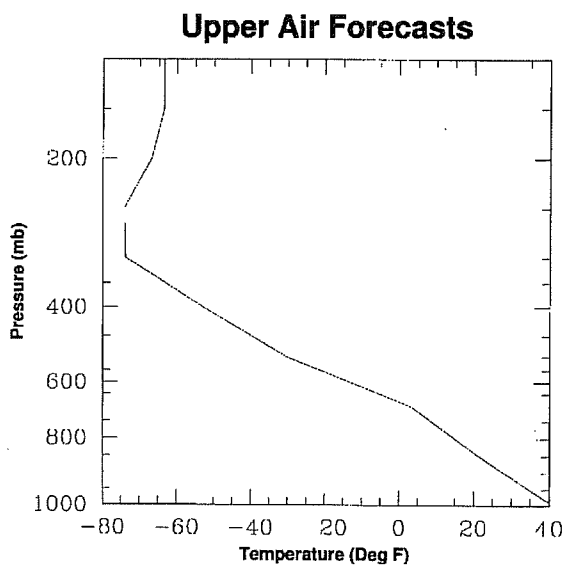
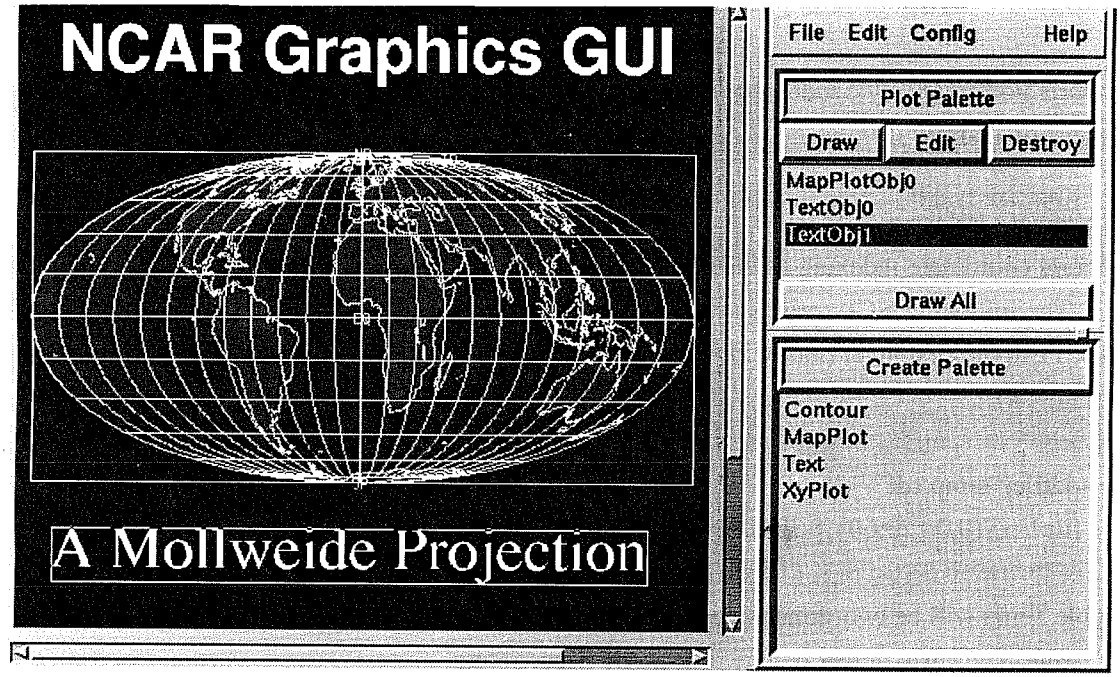


Figure 4. An NCL Visualization Block.

3.4 **The Graphical User Interface (GUI)**

The GUI for NCAR Interactive is built on top of NCL, which is built on top of the HLUs. Thus the GUI will provide access to the data input and manipulation tools as well as the visualization specification. The GUI simplifies many of the user's tasks by providing a "point-and-click" style interface for selecting data, adding data to a plot, and positioning plots within the output frame. Furthermore, the GUI allows users to "probe" data visualizations by attaching NCL scripts that perform some analytical function to various mouse and key press events, making true data interaction possible. Figure 5 shows a number of graphical objects drawn on the GUI canvas. Each instance of a graphical object can be selected, moved, resized, and edited.



[Figure 5. An Example GUI Graphic.

Much work remains to be done on the GUI in the areas of extended functionality and improved user interface.

4. ADDITIONAL FEATURES OF VERSION 4

There are many other features and tools contained within the entirety of the NCAR Graphics package including:

4.1 **Hypertext Documentation**

All of the Version 4.0 interfaces are documented online in hypertext form. There is a very complete documentation set including:

- A User Guide
- A Quick Start Guide
- Reference Manuals
- Low Level Utility Fundamentals
- NCAR Graphics Contouring and Mapping Tutorial
- NCAR GKS-0A User Guide

The NCAR Graphics web homepage is accessible via:

URL = http://ngwww.ucar.edu/

4.2 **Command tools**

Command tools have been added for running the many example programs included within the distribution and for linking user codes with the NCAR Graphics libraries. There are additional tools including a set of file filters, an NCAR CGM (NCGM) file editor, NCGM and raster file viewers, an interactive NCGM display tool that provides animation and zooming capabilities (idt), and more.

4.3 **Graphical Output**

Graphical output can be to an X11 window, a CGM, or a PostScript file. All variations of PostScript are supported including mono and color for EPS, EPSI, and standard PostScript. Output can be in landscape or portrait mode. Page sizes can be controlled to use all, or any part, of a physical page for a PostScript page.

4.4 **Portability among UNIX platforms**

NCAR Graphics is supported on most UNIX platforms that are popular in universities and other research communities. The current list is maintained within the online documentation set discussed in Section 4.1; the URL is http://ngwww.ucar.edu/info/systems.html.

Shea, Trenberth, and Reynolds, A Global Monthly Sea Surface Temperature Climatology, NCAR/TN-345+STR, November 1990.