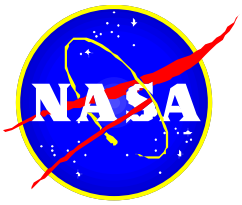

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas 77058

November 1997

Human Research Facility (HRF)
Human-Computer Interface (HCI) Design Guide

LS-71130

PROJECT DOCUMENT APPROVAL SHEET

DOCUMENT NUMBER	DATE	NO. OF PAGES
LS-71130	12/15/97	35

TITLE:

Human Research Facility (HRF)
Human-Computer Interface (HCI) Design Guide

APPROVED:	SP3/B. Woolford Human Factors Manager	Original Signature on File
APPROVED:	SM4/D. Grounds HRF Project Manager	Original Signature on File

DATE	PREPARED BY	CHANGE APPROVALS	CHANGE NUMBER

Report Number LS-71130 **Date** **11/24/97**

Human Research Facility (HRF)
Human-Computer Interface (HCI) Design Guide

Prepared by: _____ _____
Tina Holden Date
Human Factors Specialist

Prepared by: _____ _____
Jurine Adolf Date
Human Factors Specialist

Approved: _____ _____
Margaret A. Klee Date
HRF Software Lead

Approved: _____ _____
J. C. Kukla Date
HRF Hardware Development Section Manager

Approved: _____ _____
E. H. Patnovic Date
HRF Project Manager

Prepared by:

Lockheed Martin Space Missions Systems and Services
Houston, Texas
for the
National Aeronautics and Space Administration
Johnson Space Center
Contract No. NAS9-19100
TO No. HELSAB1A

Lockheed Martin Engineering & Science Services (November 1995)

DOCUMENT NUMBER LS-71130	**DOCUMENT CHANGE/ REVISION LOG**	PAGE 1 OF 1

CHANGE/ REVISION	DATE	DESCRIPTION OF CHANGE	PAGES AFFECTED
BASIC	12/15/97	Baseline Issue	

Altered pages must be typed and distributed for insertion.

CONTENTS

CONTENTS (Cont'd)

ACRONYMS AND ABBREVIATIONS

C&W	Caution and Warning
CDR	Critical Design Review
COTS	Commercial-Off-the-Shelf
GMT	Greenwich Mean Time
HCI	Human-Computer Interface
HRF	Human Research Facility
ISS	International Space Station
PCS	Portable Computer System

1.0 PURPOSE

The Human-Computer Interface (HCI) Design Guide for the Human Research Facility HRF is intended as a brief, easy to use reference guide for the designers of all HCIs within the HRF, including the ground development facility. The approach of this document effort was to compile and summarize selected HCI guidelines collected from numerous published texts. This document should: (1) educate designers in the HCI design process and basic design principles, (2) point designers to additional resources where necessary and (3) be used to establish a common ground for collaboration with human factors professionals in the design of HRF HCIs. This document does not control the HRF software development and configuration management processes defined in the Software Development Plan for the Human Research Facility (LS-71020), however, the HCI design process described in this document should be integrated into the software development process by the user interface design team.

This design guide is relevant to all software interfaces for HRF components, whether they be a portable or workstation computer, or any device that includes an electronic visual display. This guide should be referenced for all procurement activity where the vendor will write HRF specific software, but it is not applicable to Commercial-Off-the-Shelf (COTS) software and other software not specifically written for HRF. Given the potential for multiple software platforms within HRF, this document should be used in conjunction with platform specific HCI guidelines, such as those referenced in Section 7.0. Consultation and/or collaboration with human factors professionals is strongly recommended in order to ensure proper implementation of these guidelines.

2.0 SCOPE

The HCI Design Guide for the HRF consists of high-level guidelines and design principles. The design of individual screen objects, or widgets, is out of scope for this document. Platform specific widget design information can be found in the documents referenced in Section 7.0. The rationale for the limited scope and brevity of this document is two-fold: (1) a poorly designed HCI is rarely due to lack of compliance with very specific guidelines, but is almost always due to lack of attention to the most basic design principles, and (2) a brief, high-level document is more likely to be used, thus increasing the chances for a common design approach across all HRF interfaces. A more exhaustive set of guidelines, can be found in the publications cited in the Bibliography.

This document has been designed for the HRF Project, versus other International Space Station (ISS) systems or payloads which may have HCI designs. It is a companion document to the HRF Flight Support CSCI document. The ISS Display Style Guide and ISS requirements documents, listed in the following section, should be consulted for guidelines outside the scope of this document.

3.0 APPLICABLE DOCUMENTS

Document Number	Title
LS-71000	Program Requirements Document for the Human Research Facility (February, 1997)
LS-71020	Software Development Plan for the Human Research Facility (July, 1997)
SSP 50005	International Space Station Flight Crew Integration Standard (August, 1995)
TBD	ISS Display Style Guide (June 20, 1997) (web site http://kria.jsc.nasa.gov/CDDT/iss_disp_public.html)

4.0 APPROACH

The approach taken in the development of this design guide was to research and compile the latest HCI guidelines publications available. The guidelines were prioritised and the most important were included in this document. In addition, platform specific guidelines documents were sought out for the platforms most likely to be used in the HRF. These are listed in Section 7.0. Where applicable, guidelines resulting from NASA HRF investigations have also been included.

HRF will support multiple computer platforms. There is a potential for confusion given the terminology differences across operating systems. For example, similar widgets have different names in different operating systems. A table of cross-platform terms is presented in Appendix C for further clarification.

5.0 HCI DESIGN PROCESS

The HCI Design Process is distinguished from the Software Development Process in several important ways. At a high level the Software Development Process consists of three basic phases: (1) Develop requirements and specifications, (2) Design and build code to specifications, (3) Review and test code for proper functionality (i.e., "Does it work?"). Software development typically has little to no involvement with the end users. The emphasis is on meeting specifications and making sure that the code is error free.

Although some parallels exist, the HCI design process is different. The most important difference is that the HCI design process is intentionally iterative. An optimal user interface is the product of multiple design, review and evaluation iterations. Here, the evaluation is more targeted toward usability issues (i.e., "Is it easy to use?") rather than functionality. In addition, involvement of users early in HCI development is an absolute requirement. This approach is referred to as "User-Centered Design." The philosophy is to fit the system to the user, rather than make the user adapt to the system, as has been typical in the past. The HCI Design Process consists of Assembling a Design Team, Functional Requirements Definition, Task Analysis, Prototyping, User Evaluations, Small-Scale Design Studies (optional) and Formal Usability Evaluation. Figure 1 shows the flow of the HCI Design Process.

The following sections provide detailed descriptions of each stage. If HCI developers work with human factors professionals, this process can be simplified or abbreviated when necessary to meet project cost and schedule constraints and to optimize the cost-benefit ratio of the HCI development effort.

5.1 DESIGN TEAM

Before design work can begin, it is important to create a balanced user interface design team. Designing optimal interfaces requires multiple areas of expertise. The core design team should be made up of individuals who have (1) content expertise, (2) technical implementation expertise and (3) design expertise. This combination is rarely found in a single individual. Content expertise is provided by subject matter specialists (usually within the sponsoring organization). Technical implementation expertise is provided by developers/prototypers, and design expertise is provided by human factors professionals. At least one user should be included as part of the team; the user's perspective is critical. It is important that the team work together to ensure that the best design decisions are made. The team should participate in the following design stages, along with other personnel brought in as necessary.

5.2 FUNCTIONAL REQUIREMENTS DEFINITION

One of the most important first step in design is defining the functional requirements. During this stage, the user interface design team should collaborate with the software developers in order to understand the functional requirements which have been defined for the system. The user interface design team should then define the user requirements with respect to the functional requirements of the system. All functions and subfunctions of the system should be thoroughly discussed and clearly described during this stage. It is important that the concerns and perspectives of all personnel involved be represented during this stage. The focus should be on user needs and

determining the variety of ways that the new system may be used. The development and use of scenarios can be helpful in defining requirements.

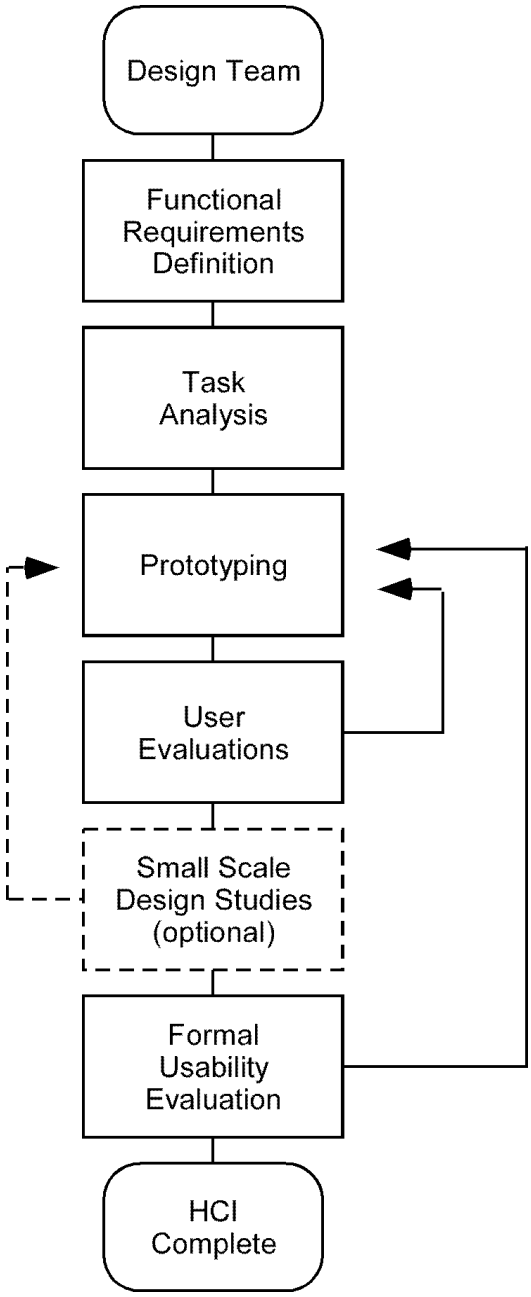

Figure 1. HCI Design Process

(NOTE: The dashed box indicates a stage in the process that is optional, or performed on an "as needed" basis. The arrows indicate opportunities for multiple iterations of the design.)

5.3 TASK ANALYSIS

A task analysis is the collection and analysis of information about the intended users of the system and the tasks that they will perform. There are a variety of techniques for collecting and analyzing this information, but the primary activities include gathering of information from the parties knowledgeable about the system being developed. This also often involves the development of task scenarios. Users should be included in this process. The main question to be answered during this stage is "What is the Task?". Table 1 shows a list of typical questions that could be asked during a Task Analysis. Much of information collected during this stage will fulfill requirements due at the Critical Design Review (CDR).

TABLE 1. EXAMPLE TASK ANALYSIS QUESTIONS

- What is the sequence of user actins, decision points, common functions?
- What is the relative importance of each subtask?
- Where are the dependencies?
- Where is the task performed and what are the environmental conditions?
- What features/practices from past tasks should be retained or discarded?
- What other tools does the user have?
- How do users communicate with each other?
- What knowledge level of the system and task does the user possess?
- How often is the task performed? Will refresher training be required?
- What are the contingency plans?

5.4 PROTOTYPING

Prototyping is the creation of numerous design concepts for an HCI. This can be accomplished with paper and pencil, or an interactive prototyping tool. The goal during the first concept prototyping stage is to get multiple ideas down in a visual form, so that they can be reviewed and discussed. The development process is iterative, therefore, a minimal amount of time should be spent developing the initial concept prototypes. The prototypes should evolve and mature in detail and fidelity as each design/review iteration is completed. The design principles and guidelines in this document should be exclusively on the design of the interface and thus may not include the underlying functionality of the software system.

5.5 USER EVALUATIONS

The User Evaluations stage goes hand in hand with the prototyping effort. Once several prototypes have been developed, the team should review and discuss them.

Additional users should be brought in to discuss the advantages and disadvantages of different display concepts with the team. Based on the results of the discussion, the prototypes are iterated and reviewed again until the team feels that the designs are mature enough to begin formal evaluation.

5.6 SMALL SCALE DESIGN STUDIES

Often during concept prototype reviews there is not a consensus regarding the best design solution for a particular display object, or there is no logical case for the selection of one design over another. In these situations, the best course of action is to perform a small scale design study in order to compare the design alternatives. A small scale design study refers to a laboratory experiment designed to specifically compare multiple design solutions, (e.g., which icon design is recognized the quickest and most accurately). This stage is optional, but is the best way to be confident that the best design has been selected. This stage may not be necessary for low impact decisions, but is particularly important when critical design decisions are at stake. Results of the small scale design study should be reflected in the next iteration of the display prototype and reviewed by the team.

5.7 FORMAL USABILITY EVALUATION

Once there is general agreement that the proposed designs are acceptable, it is time to collect formal usability data. This serves two primary purposes: (1) It provides an opportunity for the interface to be used and evaluated within the context of a real-world task, and thus opportunities to discover any problems with the interface, and (2) it provides for the collection of objective data, as opposed to subjective opinions only. This objective data can be used for determining whether or not the usability goals have been met or for comparing the new system against existing systems. This data can also be used for preliminary timelining of the task, as well as for marketing the new system.

There are many different techniques for conducting usability evaluation. These evaluations are usually scenario-based, whereby the participant completes a list of procedures designed to “Exercise” all of the human interface components and functions. These sessions are usually timed and all responses recorded. Following the task, users are asked to complete a questionnaire or rating scale about various aspects of their experience with the interface. Participants are often videotaped to capture attention, frustration, and verbal comments during participation. A technique called “Verbal Protocol Analysis” is very useful for collecting preliminary (i.e., “first pass”) usability data. In this technique, participants are asked to verbalize (i.e., speak their thoughts), while they are performing the task. In this way, the experimenter is able to easily identify points of frustration and confusion in the interface. Once a usability evaluation has been completed, and the results analyzed, the problems identified should be addressed through design iteration, as shown in Figure 1.

6.0 DESIGN GUIDELINES

Three types of design guidelines are presented below. These include 1) General Design Principles, which can be thought of as high-level design goals, 2) Display Object Design Guidelines, which provide specific guidance for the design of objects on the display and 3) Interaction guidelines, which cover design issues related to the interaction between the human and the software interface.

6.1 GENERAL DESIGN PRINCIPLES

6.1.1 Directness

Design Principle: The interface should be direct in style.

The interface should be designed such that users feel they are in control and can directly and naturally interact with objects on the screen.

a. Users should be able to see which options are available to them at any point in time.

b. Users should be able to immediately see the consequences of their actions.

c. Users should NOT have to rely on memory in order to interact with the system.

d. Interfaces should be based on the design of the physical system or use a metaphor where possible to allow users to take advantage of their knowledge and experience in interacting with the system.

6.1.2 Consistency

Design Principle: The interface should be consistent (1) within a display, (2) across displays within a system, and (3) across systems that are to be used together. For example, HRF displays should be as consistent as possible with ISS Portable Computer System (PCS) displays.

Consistency is important for helping users to carry their knowledge and experience of how the interface works from display to display. Building in consistency helps to minimize or eliminate user frustration and errors caused from switching between different displays and different systems.

a. Interfaces should have a consistent look. Widgets that serve a similar purpose should be similar in appearance.

b. Interfaces should have a consistent style of interaction. Widgets that serve a similar purpose should function in the same way, have the same types of inputs, outputs and visual attributes.

6.1.3 Forgiveness

Design Principle: The interface should be forgiving of mistakes and provide support for error recovery.

An interface is forgiving if it prevents user errors from (1) causing substantial rework on the part of the user, (2) negatively impacting the mission or (3) damaging the system.

a. An interface should ensure that no single user action will cause irreversible errors or compromise the system or mission.

b. User tasks which include important elements (e.g., those that could result in the loss of data) should be safeguarded via confirmation dialog boxes or the equivalent (i.e., arm/fire).

c. Error messages should fully describe the error in human readable terms.

d. Error messages should provide guidance on recommended corrective action.

e. Error messages should point the user to error specific help which more fully describes the function being attempted.

f. Help should be available toe the user at all times.

g. Context specific help should be provided where possible.

h. Non-context specific help should include the capability for the user to enter a term for which help is desired. The system would locate the help text or allow the user to browse a list of help topics and select one to read.

6.1.4 Feedback

Design Principle: Visual and/or auditory feedback should accompany every user action.

Feedback plays an important role in human-computer interaction and should not be thought of as being limited to error messages. For example, the simple user action of pressing a keyboard key typically results in key travel (kinesthetic feedback) as well as the appearance of the letter on the display (visual feedback). This important information lets the user know that the key has been successfully pressed. Visual feedback (e.g., button highlights, wait cursor) should be provided so that the user is aware that the system has received the input.

a. Visual feedback should accompany every user action.

b. Feedback should be provided as close in time as possible to the completion of the user entry/action. This may consist of a momentary highlight when a button is clicked or the appearance of a wait cursor when a command has been sent.

c. The complexity of the feedback should be tied to the corresponding user action. For example, a visual highlight is sufficient for a simple button press, a status bar is appropriate for an operation which takes thirty seconds (e.g., a file copy), and a message box with recovery information is warranted where a serious error has been made.

d. The capability for auditory feedback should be provided for use with headphones only. Auditory feedback should be limited and significantly different than the alarms and tones reserved for Caution and Warning on PCS displays (PCS reference). Auditory feedback should be redundant with visual feedback.

e. The user should have the capability to disable auditory feedback on HRF interfaces.

6.1.5 Aesthetics

Design Principle: Aesthetics should be considered in display design, as long as ease of use and functionality are not compromised.

Aesthetics are important for user interest, motivation and acceptance. However, there is a fine balance between designing for aesthetics and designing strictly for usability. Excessive decoration may degrade performance and detract from the usability of the interface.

a. In designing the overall "look" of the display, attention should be given to order, balance, sequencing, structure, color selection and other variables which can enhance visual appeal.

b. The use of color or graphics for decorative purposes alone should be minimized. Every visual element that appears on the screen competes for the user's attention.

6.1.6 Content and Navigation

Design Principle: A display should contain only the information that is relevant to the current task, in the proper format, at that point in time.

It is important to be selective when making decisions about the content to be displayed. There is seriously flawed common belief that every piece of data should be available to the user on the primary display at the same time. A display that shows too much content by default at all times will result in poor usability and poor performance, since the user must sift through all of the information to find what is relevant for their task at that time. However, a display that hides too much information in deeper layers, creates an additional interaction burden on the user, since they are now required to select additional buttons and menus to access the information that they need to perform their task. The key is to identify the minimum core information for default display without creating excessive overhead for the user. Usability evaluation helps to insure that the proper balance has been achieved.

a. The primary information required for performing the task should be on the main task display. Supplemental or secondary information should be provided upon user request only.

b. Information layering should be used to limit data display to that which is needed for the task at hand.

c. For most applications, information should be layered no deeper than 4 layers.

d. Users should be able to see where they are in the display hierarchy and have access to any level at any time.

e. Users should be able to return to the top level of application with a single action.

6.1.7 Organization

Design Principle: Information within a display should be organized according to logic and standard accepted conventions.

A display that is well-organized communicates important information to the user. Good organization helps user know where to look for information when quickly scanning, identifies to the user the relevant information within a set of data and also indicates the required sequence of actions within a procedure.

a. HRF HCIs should be visually distinct from PCS displays (e.g., via HRF logo) to reduce confusion about the capabilities of the system.

b. Information should be grouped according to purpose or function.

c. Headers, titles and labels should be used throughout the display.

d. Information/display objects to be used together should be placed in close proximity on the screen.

e. Information used in a sequence should be organized in either a left to right or top to bottom orientation.

f. Displays should be mapped to actual system configurations when possible.

6.2 DISPLAY OBJECT GUIDELINES

6.2.1 Windows

a. All windows should have a title bar containing a centered title

b. Display windows should include widgets to allow resizing, iconification and movement of the window.

c. If all the information within a display does not fit within the window, scroll bars should be provided.

d. A sequence of displays with which the user will have to interact in close temporal proximity should be contained in separate windows which can be displayed simultaneously.

e. The user should have the capability to select between “tiling” and “overlapping” window environments.

6.2.2 Logon

The logon capability provided by the operating system (OS) should be used. If no capability is provided by the OS, a logon should be provided.

6.2.3 Text

a. Helvetica should be used as the primary font on all displays because it is a sans serif font that is highly legible at variable distances and will be used on PCS displays.

b. For displays with 640 X 480 resolution, the minimum point size should be 10 point.

c. For displays with 1024 X 768 resolution or higher, or when the resolution may vary, the minimum point size should be 14.

d. For displays used primarily under normal illumination, all text should be black, except when indicating unavailable options, when it should be gray.

e. In environments requiring dark adaptation, light characters on a dark background should be used.

f. All text should be shown in mixed case, except for major titles, headings, labels and acronyms.

g. Text should generally be left justified (ragged right edge), including the first word of a paragraph.

h. Line lengths of extended text should be between 52 and 80 characters in length.

i. A high brightness contrast ratio between text foreground and background should be used to ensure readability of the text.

j. Whenever text is selected, the visual indication of the selection should be a reverse video of the text.

6.2.3.1 Acronyms and Abbreviations

a. Abbreviations and acronyms should be used only if a display does not have sufficient space for the unabbreviated word or if the abbreviation or acronym is more frequently used that the full word or phrase (e.g., NASA).

b. All acronyms shall be selected from the HRF approved acronyms list (reference document – TBD).

c. Definitions for all acronyms and abbreviations used within an application shall be available in a help file.

6.2.3.2 Titles, Headers and Labels

6.2.3.2.1 General Guidelines

a. Each display or window should have a unique, meaningful identifier (e.g., in the title bar).

b. Major titles, headings and labels should be displayed in uppercase to facilitate scanning. The number of uppercase labels on a display should be small.

c. Lower level headings and labels associated with graphics should be displayed in mixed case. The majority of labels on a display should be displayed in mixed case.

d. The background color of the title, header or label should be the same as the background color on which it appears.

e. The foreground color of the title, header, or label should be black.

f. Display titles should always be consistent with the menu item or button label that was used to access that display.

g. Labels should be located in close proximity to the objects they are labeling.

h. Labels should always be displayed in a normal orientation in relation to the display (i.e., left to right). When necessary to display a label on the vertical axis as opposed to the horizontal axis, the letters should retain their normal orientation to one another (i.e., read sideways), as opposed to being oriented to be read from top to bottom.

i. If all the data described by a label has the same unit of measure, the label should include the unit of measure.

j. If data described by a label do not have the same unit of measure, each data item shall include the units of measure.

k. Field labels should appear on top or to the left of the data, followed by a colon.

l. When feasible, field labels should provide cues beside the label in parentheses. For example, "Cost ($):____".

m. Where the entry is fixed length or fixed format, the field should indicate the length or format using symbols (e.g., "Date (mm/dd/yy): __/__/__).

n. When a set of data is grouped by a border, the group label should be incorporated into the top center of the border (in line with the border), and separated from the border by 2 character spaces on each side.

o. Whenever display objects or menu items are not available, the text label and object outline (if appropriate) should be dimmed/grayed out to indicate that it is not available for selection.

6.2.3.2.2 Naming Conventions

a. "OK" should be used to commit changes to a window and to acknowledge messages. When "OK" is selected, the window automatically closes.

b. "Apply" should be used to commit changes made to the content of the window, without closing the window. "Apply" should NOT create a new saved state.

c. "Reset" should restore the contents of the window to the last saved state and should always accompany an "Apply" button.

d. "Stop" should be used to terminate an ongoing process.

e. "Continue" should be used to continue a process that has been interrupted by the operating environment. This button should be accompanied by the "Cancel" button.

f. "Retry" should be used to allow the user to correct and retry a process that has been interrupted by the operating environment. This button should be accompanied by the "Cancel" button.

g. "Pause" should be used to temporarily suspend an ongoing process without terminating the process. This button should be accompanied by the "Resume" button and the window should remain open.

h. "Resume" should be used to continue a process that was paused. "Resume" should be accompanied by a "Cancel".

i. "Close" should be used to close a window without affecting a process.

j. "Cancel" should stop the process and close the dialog box, if there is an ongoing process.

k. If there is no ongoing process, "Cancel" should close the window without applying any changes to the window contents.

l. "Yes" should be used to indicate a positive response and close the window.

m. "No" should be used to indicate a negative response and close the window.

n. "Help" should be used to access specific help on the function in progress.

6.2.4 Numbers

a. The number of digits shown beyond the decimal point should be only the number required for decision making during the task (usually no more than 2). If whole numbers are used for decision making, no decimal places should be shown.

b. Leading zeros in numeric entries for whole numbers should NOT be shown (i.e., display 28 rather than 0028), unless specifically required for the task.

c. If the number being presented is a decimal with no preceding integer, a leading zero should be shown, (i.e., display 0.43 rather than .43).

d. Units of measure displayed should be the units of measure needed by the task. The user should NOT have to convert units of measure in order to use the data. If multiple units of measure are required, these should be available to the user on request.

6.2.5 Tables

a. In a tabular display, a blank line or thin line separator should be shown between sets of rows in the table, at consistent intervals (e.g., every 5 lines).

b. If horizontal scrolling is provided, the column labels should scroll with the data.

c. If vertical scrolling is provided, the column labels should NOT scroll with the data.

d. Alphabetic and alphanumeric columns of data should be left-justified.

e. Numeric columns of data should be right-justified by either the fixed decimal point or implied decimal point (i.e., whole numbers).

6.2.6 Diagrams

a. Diagrams depicting high to low quantities should show components oriented from top to bottom.

b. Diagrams depicting signal flow from source to destination should show components oriented from left to right.

c. Diagram symbology and use of color shall be consistent with the symbology and color use on PCS displays (see ISS Display Style Guide).

d. A diagram should only contain the level of detail required for the task. Additional detail can be provided on request.

e. A flowchart should be organized according to its use. A flowchart should show the first event on the leftmost side if the flow or sequence is horizontal and on the top if the flow is vertical.

f. Maps and diagrams used for similar purposes should be displayed with a consistent orientation and reference points. The ISS Display Style Guide should be used for depicting standard orientations and reference points for ISS modules.

g. In the absence of ISS standard orientations, graphical display elements should be designed such that conflicting information is NOT communicated (i.e., make sure "LEFT" is NOT shown on the right, "UP" is NOT on the bottom, "LOW" is NOT on the top, "STOP" is NOT written on a green background, etc.).

6.2.7 Graphs

a. A graph should be used when users need to monitor changing data, or quickly scan and/or compare sets of data.

b. A graph should be used when showing categorical or trend data.

c. A graph should be used when showing continuous data that can be categorized without a loss in information content.

d. In general, a graphical display should use the fewest lines or objects possible to accurately represent the data.

e. In graphs, the user should be able to identify off-nominal values (e.g., color change) in tasks where there is a need to discriminate between such values.

f. A scatterplot should be used to show how two variables are correlated or distributed.

g. A bar graph should be used to show a comparative measure for discrete variables, for discrete levels within a variable, or for a variable at different times.

h. If there is some sequence implied in the variables show in a bar graph, that sequence should be reflected in the order of the bars on the X axis. For example "LOW, MEDIUM, HIGH" should appear in that order, left to right, "1, 5, 10" in that order, left to right, etc.

i. A line graph should be used to portray changes through time for one or more sets of data, such as trends over a period of hours, days, weeks, months or years.

j. Whenever it is not feasible to label each object that is coded, a legend that can be hidden on user request should be provided.

6.2.8 Multimedia

6.2.8.1 General Guidelines

a. Use the same type of control for all media types. For example, if icons are used to access video, icons should be used to access audio.

b. Present information in the media form which is most compatible with the way the information will be used. For example, if the task requires the identification of tones, the information should be presented aurally. The tones should NOT be described with text only. Multiple modes of presentation should be used when possible (e.g., text + auditory + graphics).

c. For critical information that must be accessible to the user at any time, use text or graphics as the primary mode of presentation, with should or video as secondary, due to their transient nature.

6.2.8.2 Sound (headphones are the only sound output device for HRF)

a. Sound should be used judiciously because it requires a great deal of storage space and can distract attention away from the primary task if not absolutely required for the task.

b. Sound should be used when

-- sound is the subject, e.g., heart murmurs, warning and error tones

-- immediate action is required

-- the visual system is overloaded

-- monitoring tasks in a high stress environment

-- exploring large data sets; tells eyes where to look

-- there is a need to add to the dimensionality of the visual displays

c. When sound is not the topic, it should be used only as a secondary cue – redundant with another coding method (e.g., text message).

d. When sound is the subject, allow the user to replay the audio signal as many times as desired.

e. A sound track can be used in combination with a clear static picture as an alternative to full motion video. This requires less storage space and performs better on many platforms.

f. When a set of audio tones must be differentiated, locate the audio tone controls in close proximity (either initially, or in a summary area) so that the user can repeatedly play the tones and try to differentiate among them.

6.2.8.3 Music (headphones only for HRF)

a. Carefully consider the need for music. Music usually cannot convey any specific information; it typically only provides an emotional context. It may be perceived as distracting, annoying, and frivolous.

b. If music is used in the background, make the volume lower that that of the subject matter, but louder than ambient background noise.

6.2.8.4 Voice (headphones only for HRF)

a. Use voice when speech is the message, e.g., learning a language.

b. Use a narrator with good diction and a clear well-modulated voice. Base the choice on how well the voice sounds on the target computer platform.

c. If recognizing the words is sufficient (i.e., subtleties and inflections are not important), digital compression of the recording can save storage space and download time.

d. Use voice to explain complex subjects. Voice can narrate an animation without distracting from it.

6.2.8.5 Text (see section 6.2.3 for more Text guidelines)

Text is typically seen as the most credible source of information, and should be used to transmit critical information whenever possible.

6.2.8.6 Graphics

a. Use graphics when the object is familiar, but the name may not be.

b. Use graphics when spatial relationships are important. Graphics can show the arrangement of parts, important surroundings, etc.

c. Use graphics to reduce display density.

d. Use graphics to display dynamic data.

e. Do NOT use graphics when exact numeral readings are required. In this situation, digital displays provide more accuracy than analog graphical displays.

f. Graphics should be no more complex than is necessary to convey the desired point. Keep graphics as simple as possible in terms of colors, image shapes, and temporal variations.

g. Scale size and orientation of graphics should be consistent with the other objects to which they are related.

h. Successive zoom images should be used when there is a need to maintain context or hierarchy, e.g., when teaching a procedure involving disassembly.

6.2.8.7 Video and Animation

a. Video and animation should be reserved for situations where the dynamic aspects of a subject are being presented. Do NOT use gratuitous animation or video.

b. Video and animation should be used when the majority of the information can be communicated in the video or animation itself. If extended explanation is required, static pictures with text or spoken narration are a better choice.

c. Every scene or moving segment must make an important point.

d. Do NOT show characters talking about a subject (e.g., talking heads). Use voice-over narration to explain the concept.

e. Use visual transitions (e.g., dissolves, wipes, fades) to bridge gaps in time and location.

f. Keep the scenes clear and simple/uncluttered. Such sequences are easier to comprehend and can be compressed more. Avoid complex backgrounds.

g. The user should be able to pause, stop, fast forward, reverse, and replay video and animation as many times as needed. Variable speed control is sometimes useful for both forward and reverse video, if accompanying audio is disabled. Random access to points in a sequence is also useful for long duration video or animations.

h. Use animation to show movement and action by generic objects or people (e.g., to address subject too dangerous, too sensitive or too difficult to photograph).

i. Use animation to show abstract concepts, non-existent objects or processes that cannot easily be seen/videotaped.

J. Use animation to avoid the distracting details of photographic images.

k. Use animation when it is necessary to highlight or isolate a portion of an assembly for viewing.

6.2.9 Information Coding

Information can be coded for either identification or discrimination. Coding for identification means that the user should be able to identify the meaning of a coded

Object (e.g., a red object means there is an emergency). Cooling for discrimination means that the user should be able to discriminate between different groups or classes of objects on the screen (e.g., various colors on a line graph, use of highlighting for important information). In this case, the blue in a line graph does not carry any meaning, the color merely serves to set one group of data apart from another.

a. Whenever display objects or text are selected, the visual indication of the selection should be a reverse video effect of the object or text, or the item should be boxed.

b. Whenever display objects or menu items are not available, the text label and object outline (if appropriate) should be dimmed/grayed out to indicate "not available".

e. Coding for discrimination (highlighting) should always serve to enhance the usability of a display by drawing attention to, or highlighting important information.

f. Coding used for emphasis or highlighting should be minimal. When coding techniques (e.g., bolding or color) are overused, the emphasis is lost. Examples of good uses of highlighting include data that has exceeded limits, abnormal conditions, important parameters that have changed, a display item that requires the attention of the user before a process can continue, and errors that would have a significant negative effect.

6.2.9.1 Color

a. In order to avoid confusion with color assignments on PCS computers, the use of red or yellow should be avoided in the display of HRF objects and icons. The use of red or yellow may be used in graphs where there is little chance for confusion with the meanings reserved on the PCS displays.

b. Although Caution and Warning (C&W) messages will not be enunciated on HRF displays, developers should be aware of the following PCS color assignments:

Nominal	=	black foreground with white background
Caution	=	black foreground with yellow background
Warning	=	white foreground with red background
Emergency	=	white foreground with red background

c. The PCS Color Library (see Appendix A) should be used when representing the same types of items that will appear on PCS displays.

d. The default background color of all primary windows should be gray.

e. The default foreground color of all windows should be black.

f. A display should NOT rely on the use of a color to distinguish among display elements. Displays should be initially designed without color. Add color only where necessary.

g. Use color as a redundant code only; in other words, color should always be used in conjunction with another coding mechanism. Monitor differences, lighting and color deficiencies (i.e., color blindness of the user) can result n misinterpretations of onscreen colors.

h. No more than 11 colors should be used in a display for discrimination.

i. When using colors for discrimination (as in a graph), use red, blue, green and yellow first (see "a." above for restrictions on the use of red and yellow).

j. For dark backgrounds, unsaturated colors such as light yellow, cyan and green should used for foreground. Dark, saturated colors, such as red, should be avoided.

k. For light backgrounds, saturated colors such as red and blue should be used for foreground. Highly unsaturated colors such as light yellow should be avoided.

l. When selecting a set of color codes for categories where no order or pattern is intended, spectrally extreme hues should be selected (e.g., green, yellow, orange, red, violet, blue).

m. When selecting a set of color codes for categories where order is intended, hues from adjacent colors should be selected (e.g., blue, blue-green, cyan, blue-violet, violet).

n. No more that 5 colors should be used in a display for identification.

o. Colors should only be used to carry one meaning within a display.

p. Where there are no color standards for a particular object or meaning, follow the popular stereotype. For example, water is typically represented as blue, thus it would NOT be a good design choice to code water orange.

q. When using color for identification, avoid highly saturated colors, especially for large fill areas.

6.2.9.2 Reverse Video

Reverse video should be used to indicate a selected state.

6.2.9.3 Bordering

Bordering should be considered for highlighting text and can be used in combination with color. Bordering text with a blue rectangle, as opposed to printing the message in blue, communicates the same highlight message without interfering with the legibility of the text.

6.2.9.4 Blinking

Blinking is NOT a preferred method of information coding and should not be used. Blinking is distracting, makes text difficult to read and frustrates the user.

6.2.9.5 Shape

Shapes (e.g., triangles, squares, circles) can be used to convey information about status (e.g., a triangle could indicate caution), or that elements of a data group are similar (e.g., circles could represent stars, hexagons could represent meteors).

a. Keep the shapes simple. Provide only the minimum amount detail required to recognize the shape.

b. Use a minimum of color within the shape.

c. Adequate foreground/background contrast should be provided in order for the shape to be optimally detected.

d. No more than 15 different shapes should be used for information coding, since this is the maximum that can be accurately discriminated.

6.2.9.6 Position

a. Position (location) coding should be used whenever possible since it does not require additional visual complexity. For example, place similar objects in the same location across displays.

b. Position coding should be used to group information. Related information should be physically grouped together on the display. Displays should provide cohesive groupings of display elements so that users perceive large screens as consisting of smaller identifiable pieces or chunks.

6.2.10 Software Controls

a. Appendix B, containing a table of controls and guidelines for their selection, should be used when selecting a control type.

b. Since recognition is better than recall, available commands should be presented to the user for selection (e.g., as in a menu), rather than the user having to remember commands and command syntax.

c. All software controls that are selectable should appear to have raised surfaces (e.g., 3-D look).

d. Whenever buttons or icons are selected, the visual indication of the selection should be a 3-D depression (i.e., appears pushed in).

e. The minimum selectable object size should be 10 mm X 10 mm.

f. Labels for selectable objects that lead to a dialog box, rather than result in the immediate execution of a command should be followed by an ellipses (i.e., "…").

g. Buttons and menu items which are used for navigating to another display should show the label followed by ">>".

h. Buttons that provide for navigation among displays should be located in a grouping on the right side of the display (for consistency with PCS).

i. Menus should consist of between 2 and 15 selections.

j. Menu items should be listed in a logical order based on sequence of operations, functionally or frequency of use. If no basis for menu item order exists, items shuld be listed in alphabetic order.

k. If a menu bar is used, category labels should be centered over each menu and separated by at least two standard characters from other category labels.

l. The categories listed across the menu bar should be ordered according to sequence of operations, functionality or frequency of use.

m. Menu items and pushbuttons which have two states (i.e., ON, OFF) for only one function (e.g., “Grid”), should show both possible states as selections, with an indicator (e.g., checkmark, circle fill) for the selected menu item.

n. Menu items and pushbuttons which have two states (i.e., ON, OFF) for multiple functions (e.g., bold vs. plain text vs. italics), should show one item label with an indicator (e.g., checkmark) when the function is turned on and should remove the indicator when the function is turned off.

o. Buttons or menu items which are used for commanding and result in immediate execution when selected should appear as standard rectangular pushbuttons or menu items with standard labels.

p. Buttons or menu items that are used for option selection shall appear as radio buttons when the choices are mutually exclusive and as checkboxes when the choices are non-mutually exclusive.

q. The menu bar menu labels should always begin with “File” or something equivalent and end with “Help”.

r. Even if all of the items within a cascaded menu are unavailable, the menu choice leading to the cascaded should NOT be grayed out.

s. No more than 2 levels of cascading should be used in menus.

t. A cascade menu should be identified as such by a cascade marker alongside the menu name (e.g., a right-pointing arrowhead).

u. Icons should be intuitive representations of concrete objects or actions.

v. Icons should NOT be easily confused with other icons used in the set of displays.

w. For maximum discriminability, icons should NOT be displayed inside of geometric border (e.g., squares) since this reduces icon uniqueness and adds visual complexity.

x. Icons should be simple, closed figures when possible to promote quick recognition.

y. When an icon does not clearly resemble the object or action it is supposed to represent, it should be accompanied by a text label. User evaluations should be performed to determine icon recognition rates and the need for labels or icon redesigns.

z. Icons should be selected form the HRF icon library to ensure that icon designs for the same object or function are the same. The icons in the HRF icon library should resemble, as closely as possible, the icons in the PCS icon library (reference).

aa. A data entry or input field should appear editable. It should be bounded by a black border that causes the field to appear etched in. The background of an input field should be white.

bb. A display or output field that is not cursor addressable should be shown without a border, on a background color that is the same color as the window on which it is displayed.

6.2.11 Dialog Windows (Dialog boxes)

a. Dialog windows should appear near the control that called the dialog or else centered on the display.

b. Dialog windows should NOT be resizable.

c. Dialog windows should contain a title.

d. Dialog windows should contain at least 1 button to acknowledge or cancel the operation.

e. When ordering dialog pushbuttons, place positive response buttons first, negative response buttons second and canceling response buttons last. If "Help" is included, it should be the last button on the right.

6.2.12 System Messages

a. Error and system messages should appear in dialog boxes in the center of the screen.

b. There should be a visual indication of applications that are currently active.

6.2.13 Date/Time Information

a. Date and time information should be expressed in Greenwich Mean Time (GMT).

b. Date and time information should be shown at the top right of the display, on the title bar.

6.2.14 File Naming Conventions

The Software Development Plan for the Human Research Facility (SL-71020) should be referenced for application file naming conventions.

6.3 INTERACTION GUIDELINES

6.3.1 Keyboard

HRF interfaces will be primarily controlled with cursor control devices. However, it is important to include keyboard equivalents where appropriate to allow for (1) expert users who prefer a faster means of input and (2) situations where the use of a cursor control device is not convenient or is malfunctioning.

a. Keyboard equivalents to menu items (shortcut keys) should be available.

b. Keyboard equivalents should be displayed next to the menu item label.

c. Mnemonics (i.e., usually represented as an underlined character of a menu item label) should be made available for quickly moving the cursor to a desired menu item.

d. Keyboard commands should be considered across applications.

e. If function keys are used, the user should be able to show and hide the function key label/mappings on the display.

f. Function keys and accelerator (control) keys should only be implemented for a few of the most common commands.

g. The following accelerator key mappings are accepted as standard:

Control+Z	Undo
Control+X	Cut
Control+C	Copy
Control+V	Paste

6.3.2 Cursor Control

a. The user should have the capability to use a cursor control device for all user entries (except alphanumeric).

b. The leftmost device button should be used for object selection.

c. The rightmost device button should be used for shortcut popup menus.

d. The middle button (if applicable) should be used as a double-click.

6.3.3 Object Manipulation

a. When designing indirect interactions for onscreen objects, the object-action paradigm should be followed. For example, for deletion of an item, the user should be required to first select the object and then select the delete action.

b. A drag and drop direct manipulation approach should be available for all icons or other onscreen objects that can be manipulated.

c. To the greatest extent possible, a keyboard and direct manipulation (cursor control device) method should exist for interacting with all onscreen objects.

d. A single click over an onscreen object should select that object (if it is selectable). If the pointer is moved off of the object before the selection button of the cursor control device is released, the object should NOT be selected.

e. Items which are not available cannot be selected until they return to an "available" state.

f. The location cursor (which indicates the current keyboard input focus) must be visually distinctive against all backgrounds and must be easy to locate.

g. There should only be one location cursor in a window at a time.

h. If the entry field has a default value, it should be provided in the field.

i. Standard text editing features (e.g., cut, copy, paste, undo) should be available in text entry fields.

j. The user should NOT be required to enter case specific entries unless required by the operating system or COTS application.

k. When an entry field is used for input of a restricted set of values, user input should be validated immediately and feedback provided.

l. If the entry field is contains multiple lines, text wrap-around should be provided.

m. When filling in a form-like display, users should be able to use the "Tab" key to advance to the next field and "Shift Tab" to return to the previous field. Users should also be able to directly place their cursor into the entry field.

n. Wherever possible, the need for typing should be minimized. When the user needs to select from a finite set of options, a list of alternatives (e.g., drop down list) should be provided.

6.3.4 Modes

A mode is a state where only certain user actions are acceptable. For example, insert mode in text editing, or draw mode in a drawing package.

a. The use of modes in the interface should be used sparingly, since they restrict user control of the system and can cause user frustration and errors.

b. If modes are used, a visual indication of the current mode should be shown (e.g., cursor shape change).

c. All primary windows should be modeless.

6.3.5 System Statusing

a. Whenever the system is processing information and the user has to wait, the user should be made aware that there will be a delay. Any delay requires user notification.

b. A short processing delay of less than five seconds should be indicated with a wait cursor. When the delay begins, the cursor should change shape to become a clock, watch or hourglass.

c. If the processing delay will be longer than five seconds, a different type of wait indicator should be considered. A text message that indicates that processing is in progress, a "Time remaining: __" display or a horizontal bar graph that fills to the right indicating percent complete and rate of completion should be used.

d. A processing delay in excess of 2 minutes should be accompanied by a pop up text message explaining that there will be a long delay. If possible, an estimate of the wait time should be provided.

e. Status messages should be timestamped.

6.3.6 Multi-device interaction

Information on multiple screens which must be used together, should be located/displayed in close proximity to minimize scanning.

6.3.7 Procedures

Procedures formats should follow MOD standards for writing procedures (reference?).

7.0 PLATFORM SPECIFIC WIDGET DESIGN

HRF experiments will be hosted on a number of different computer platforms. The following resources are recommended for guiding detailed human interface design on each of the platforms.

7.1 WINDOWS

The Windows Interface Guidelines for Software Design, Microsoft, Microsoft Press, 1995 (covers Windows 95 and Windows NT™).

7.2 SOLARIS

Open Look: Graphical User Interface Application Style Guidelines, Sun Microsystems, Addison-Wesley Publishing Company, 1993.

OSF/Motif Style Guide, Open Software Foundation, Prentice Hall P T R, 1993).

7.3 OS2

Object-Oriented Interface Design: IBM Common User Access Guidelines, IBM, QUE Corporation, 1992.

7.4 MAC OS

Macintosh Human Interface Guidelines, Apple Computer, Inc., Addison-Wesley Publishing Company, 1992.

8.0 BIBLIOGRAPHY

The following sources were used in the preparation of this document:

Bellcore (1995). Design Guide for Multiplatform Graphical User Interfaces. Piscataway, NJ: Bell Communications Research.

Brown, C.M. (1989). Human-Computer Interface Design. Northwood, New Jersey: Alex Publishing Corporation.

Dickenson, L. & Kane, K. (1996). The Multimedia Development Process. Http://www.dmc.tc.umn.edu/lab/mmdev.html.

Jones, & Okey, (1992). Interface Design for Computer-based Learning Environments. Http://ws114.adm.csupomona.edu/lab/articles/paper15.html.

Marcus, A., Smilonich, N., Thompson, L. (1995). The Cross-GUI Handbook For Multiplatform User Interface Design , Reading Massachusetts: Addison-Wesley Publishing Company.

Microsoft Corporation (1995). The Windows Interface Guidelines for Software Design. Redmond, Washington: Author.

National Aeronautics and Space Administration (1993). Human Computer Interface Guide, Rev A, Space Station Freedom Program Office (SSP 30540).

Rubinstein, R. and Hersh, H. The Human Factor: Designing Computer Systems for People , Burlington, Massachusetts: Digital Press.

Schneiderman, B (1987). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, Massachusetts: Addison-Wesley Publishing Corporation.

APPENDIX A

ISS COLOR TABLE

R	G	B	Color	Hexa-decimal Value	Meaning
0	0	0	Black	000000	Text, labels, borders, symbols, titles, and generally unrestricted.
255	255	255	White	FFFFFF	Background color fill for nominal state data output fields, and text on red and black fields.
255	0	0	Red	FF0000	Emergency or warning. Off scale high/low or out of critical limits status indicators.
0	255	0	Green	00FF00	Button indicator indicating on/active.
106	90	205	Slate Blue	6A5ACD	Button indicator indicating on/active.
0	255	255	Cyan	00FFFF	Static data status condition, Nonessential.
255	255	0	Yellow	FFFF00	Caution. Out of operational limits status indicator.
0	139	0	Green	4008B00	Oxygen
50	205	50	Limegreen	32CD32	Nominal/good status.
221	160	221	Plum	DDA0DD	Carbon dioxide
244	164	96	SandyBrown	F4A460	Nitrogen
245	222	179	Wheat	F5DEB3	Certain subsections of displays
190	190	190	Gray	BEBEBE	Window background
150	150	150	Gray#59	969696	Subsection background

ISS COLOR TABLE (Cont'd)

R	G	B	Color	Hexa-decimal Value	Meaning
205	183	158	Bisque#3	CDB79E	Ammonia
255	105	180	Hot Pink	FF69B4	Hydrazine
160	32	240	Purple	A020F0	Missing, dead, invalid, or Commfault status conditions
255	0	255	Magenta	FF00FF	Radiation
65	105	225	Royal Blue	4169E1	Fluid pipe: water
153	50	204	DarkOrchid	9932CC	Malfunction active
211	211	211	LightGray	D3D3D3	Unavailable or inactive within the simulation, background
255	165	0	Orange	FFA500	Fluid pipe: Hydrogen; Warning
173	216	230	Light Blue	ADD8E6	Freon
144	238	144	Light Green (also called palegraeen2)	90EE90	Helium
255	182	193	Light Pink	FFB6C1	FC40 coolant
102	205	170	MedAquamarine	66CDAA	Air

APPENDIX B

GUIDELINES FOR CONTROL SELECTION

Control	When to Use
Check Box	On or off state, single or multiple-choice, less than six fixed options
Combo Box	List of choices with user entry option, greater than six choices
Command Button	For frequently used fixed action or routing choices, less than six choices
Container	Used to group and to view any number of objects
Drop-Down Combo Box	Drop-down list of choices with user entry option, greater than six choices, conserves space
Drop-Down List Box	Drop-down list with no user option, greater than six fixed choices, conserves space
List Box	Selectable list of choices – text or graphics, greater than six choices
Notebook	Used to display large number of objects or setting choices (except another notebook) that can be arranged in a logical group (tabbed divider-pages)
Radio Buttons	Single choice, mutually exclusive, less than six fixed choices
Scroll Bar	Large list, not fully visible within a window
Slider	Analog representation, fixed setting in a range, less than sixty visible increments
Spin Box	Ordered input values, less than ten fixed choices
Text Box	Used for entering text
Value Set	Graphical choices that are mutually exclusive like color palettes

APPENDIX C

CROSS-PLATFORM EQUIVALENT TERMINOLOGY

Desktop Terminology

	Windows	Solaris (Motif)	OS2	Macintosh
Desktop	Desktop	Workspace (Root window)	Workplace	Desktop
Desktop Manager	Program Manager	Window Manager	Presentation Manager	Finder
Close Control	Control Menu	Window Menu	System Menu	Close Box
Content Area	Content Area	Client Area	Client Area	Content Area
Message Area	Message Bar	Message Area	Information Area	Status Bar
Menu Bar	Menu Bar	Menu Bar	Menu Bar	Menu Bar
Status Bar	Status Bar	Status Area	Status Area	Status Bar
Title Bar	Title Bar	Title Bar	Title Bar	Title Bar
Window Frame	Window Frame	Window Border	Window Border	Window Frame

Window Terminology

Types of Windows	Windows	Solaris (Motif)	OS2	Macintosh
Application	Application	Primary or main application	Primary	(Virtual Window)
Document	Document	Secondary Dialog Box	Secondary	Document
Others	Dialog Box	Menu Window	Dialog Box	Dialog Box Alert Box

Control Terminology

Control	Windows	Solaris (Motif)	OS2	Macintosh
Close Control	Control Menu	Window Menu	System Menu	Close Box
Maximize Button	Maximize Button	Maximize Button	Maximize Button	Zoom Box
Minimize Button	Minimize Button	Minimize Button	Minimize Button	N/A
Restore Button	Restore Button	Maximize Button	N/A	Zoom Box
Scroll Bar Control	Scroll Bar	ScrollBar	Scroll Bar	Scroll Bar
Scroll Arrows	Scroll Arrows	Arrow Buttons	Scroll Buttons	Scroll Arrows
Scroll Bar Shaft	Scroll Bar Shaft	N/A	Scroll Bar Shaft	Gray Area
Scroll Box	Scroll Box	Slider	Scroll Box	Scroll Box
Size Control	Window Frame or Resize Borders	Window Frame or Resize Borders	Window Borders or Resize Borders	Size Box
Split Box	Split Box	N/A	Split Box	Split Bar
Split Bar	Split Bar	Separator and Sash	Split Bar	Split Line
Split Windows	Window Panes	Paned Windows	Window Panes	Window Panes

Menu Terminology

Types of Menus	Windows	Solaris (Motif)	OS2	Macintosh
Pull-Down	Drop-Down	Pulldown	Pull-Down or Action bar Pull-Down	Pull-Down
Cascading	Cascading or Submenu	Pulldown	Cascaded or Cascading Pull-Down	Hierarchical or Submenu
Pop-Up	Pop-Up	Popup	Pop-Up	Pop-Up
Tear-Off	N/A	TearOff	N/A	Tear-Off
Other	N/A	Option	N/A	Scrolling

Menu Bar Contents

Windows	Solaris (Motif)	OS2	Macintosh
File	File	File (Application windows) <name of object> (Object windows)	File
Edit	Edit	Edit	Edit
View	View	View	N/A
Application-specific	Application-specific	Application-specific (Options, Windows)	Application-specific
Help	Help	Help	Help Icon
N/A	N/A	N/A	Application Icon

Control Terminology

	Windows	Solaris (Motif)	OS2	Macintosh
Check Box	Check Box	CheckButton	Check Box	Check Box
Combo Box	Combo Box	N/A	Combo Box	N/A
Command Button	Command Button	PushButton	Pushbutton	Button
Container	File Folder	N/A	Container	N/A
Drop-Down Combo Box	Drop-Down Combo Box	N/A	Drop-Down Combo Box	N/A
Drop-Down List Box	Drop-Down List Box	OptionMenu	Drop-Down List Box	Pop-up Menu
List Box	List Box	List	List Box	Scrolling List
Notebook	N/A	N/A	Notebook	N/A
Radio Button	Command Button	RadioButton or ToggleButton	Pushbutton	Button
Scroll Bar	Scroll Bar	ScrollBar	Scroll Bar	Scroll Bar
Slider	Slider	Scale	Slider	Slider
Spin Box	Spin Box	Arrow	Spin Button	Arrow
Text Box	Text Box	Text	Entry Field	Text Entry Field
Value Set	Value Set	N/A	Value Set	Value Set

DISTRIBUTION LIST
FOR
LS-71130

DF14/C.H. Armstrong

EA5/E. Bauer
- D. Baumann
- F. Booker
- C. Chambellan
- H. Dabis
- H. Nguyen
- A. Rodriguez
- B. Young

ND323/M. Iwasa

SD/B. Kyle

SD4/D. Brown/KI

SM3/M. Buntin

SM4/R. Garcia
- D. Grounds
- C. Haven
- J. Villarreal

SP3/B. Woolford

SP4/G. Branch

GHG/S. Nealon

USA/M. Shultz/D0612

KRUG
- K. Collier/SD5/KI

LOCKHEED MARTIN

Building 36
- STI Center (2)
- T. Adams/S361
- H. Rahman/S362
- A. Money/S363

Building 57
- A. Al-Hajas/SJ57
- J. Cheng/SJ57
- G. Lei/SJ57
- L. Li/SJ57
- T. Mills/SJ57
- M. Yalcinkaya/SJ57

LM2/K. Kaufman/B31

LM4/HRF Library (2)
- J. Adolf
- D. Babic/S18
- S. Brandt/S06
- G. Brown/S06
- T. Budzik
- W. Cohen/S06
- J. Dardano/S06
- S. Ezell/S06
- J. Fox
- J. Hoge/S06
- K. Holden
- M. Klee/S06
- J. Kukla/S06 (10)
- P. Li/S06
- E. Patnovic/S06
- C. Richard/S06
- M. Romell/S06
- L. Selvan/S06
- P. Shah/S06
- S. Tarver
- R. Tiniakoff/S06
- Y. Tomaro/S06
- E. Witt/S06
- J. McDonald/S07
- B. Rhodes/S12
- L. Wallace