

# **Joint Command Decision Support 21st Century Technology Demonstration**

## *Human Factors Style Guide*

CAE Professional Services

Prepared By:  
CAE Professional Services  
1135 Innovation Drive, Suite 300  
Ottawa, ON  
K2K 3G7

CSA: Dr. Wenbi Wang and Kevin Trinh, HSI Section, DRDC Toronto

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

## **Defence R&D Canada – Toronto**

Contract Report  
DRDC Toronto CR 2009-047  
March 2009

Principal Author

*Original signed by CAE Professional Services*

---

CAE Professional Services  
Human Factors Engineering

Approved by

*Original signed by Dr. Wenbi Wang*

---

Dr. Wenbi Wang  
Research Engineer - HSI Section

Approved for release by

*Original signed by Kathy Sutton*

---

Kathy M. Sutton  
Head, Document Review and Library Committee

- © Her Majesty the Queen in Right of Canada, as represented by the Minister of National Defence, 2009
- © Sa Majesté la Reine (en droit du Canada), telle que représentée par le ministre de la Défense nationale, 2009

## Abstract

---

The aim of the Joint Command Decision Support for the 21st Century Technology Demonstration Program project (JCDS 21 TDP) is to demonstrate a Joint Net-enabled, Collaborative Environment to achieve Decision Superiority at the strategic and operational levels of command. JCDS 21 applications are intended for short-duration, domestic operations that may require military and civilian responders to work together to resolve the impact of natural or man-made incidents. This document is intended to leverage recent background research which identified relevant documentation that could be used as a basis for developing a high-level usability framework for the JCDS 21 TDP Style Guide; this Style Guide contains general guidance for the development of Graphical User Interfaces (GUIs) for use in a Windows-based environment and within a joint Command and Control (C2) environment.

This work was completed under sub-contract to Fujistu Consulting and with cooperation from Defence Research and Development Canada - Toronto) and Prolicity.

## Résumé

---

Le but du Projet de démonstration de technologies – Aide à la décision des commandements interarmées pour le XXI<sup>e</sup> siècle (PDT ADCI 21) est de démontrer qu'un cadre collaboratif réseaucentrique peut permettre à un commandement interarmées d'obtenir la supériorité décisionnelle aux niveaux stratégique et opérationnel. Les applications du PDT ADCI 21 sont conçues pour les opérations nationales de courte durée qui peuvent nécessiter une certaine collaboration entre les intervenants civils et militaires pour faire face à l'impact d'une catastrophe naturelle ou causée par l'homme. Le présent document vise à tirer profit des récents travaux de recherche qui ont identifié la documentation qui pourrait servir de base à l'élaboration d'un cadre d'utilisabilité de haut niveau pour le guide de style du PDT ADCI 2. Ce guide de style contient des directives générales sur le développement d'interfaces graphiques personnalisées (GUI) conçues pour être utilisées dans un environnement Windows et dans un environnement de commandement et de contrôle (C2) interarmées.

Ce travail a été effectué en sous-traitance par Fujistu Consulting, en collaboration avec RDDC Toronto et Prolicity.

This page intentionally left blank.

## Executive summary

---

### Joint Command Decision Support 21st Century Technology Demonstration: Human Factors Style Guide

CAE Professional Services; DRDC Toronto CR 2009-047; Defence R&D Canada  
– Toronto; March 2009.

**Introduction or background:** The aim of the Joint Command Decision Support for the 21st Century Technology Demonstration Program project (JCDS 21 TDP) is to demonstrate a Joint Net-enabled, Collaborative Environment to achieve Decision Superiority at the strategic and operational levels of command. JCDS 21 applications are intended for short-duration, domestic operations that may require military and civilian responders to work together to resolve the impact of natural or man-made incidents.

This document is intended to leverage recent background research which identified relevant documentation that could be used as a basis for developing a high-level usability framework for the JCDS 21 TDP Style Guide.

This work was completed under sub-contract to Fujitsu Consulting and with cooperation from Defence Research & Development Canada (Toronto) (DRDC-Toronto) and Proliity.

**Results:** This Style Guide contains general guidance for the development of Graphical User Interfaces (GUIs) for use in a Windows-based environment and has applications for use in a joint Command and Control (C2) environment.

**Significance:** The current findings provide preliminary style guidelines for the development of JCDS 21 applications for use within a Windows-based environment and within a joint C2 specific environment. Utilization of the style guide should lead to a common look and feel. Conformance to the HF guidelines presented herein should reduce training time as well as workload, resulting in more efficient performance.

The intent of the current version of the Style Guide is to provide guidance that can be considered by designers and developers involved in the iterative development of the JCDS 21 applications. As the development of the JCDS 21 applications continues to progress, subsequent versions of this style guide are expected to provide formal recommendations associated with the GUI design.

**Future plans:** An iterative design process is being used to develop the JCDS 21 applications. The content of the JCDS 21 TDP Style Guide will continue to be refined on this basis of this iterative development.

# Sommaire

---

## Joint Command Decision Support 21st Century Technology Demonstration: Human Factors Style Guide

CAE Professional Services; DRDC Toronto CR 2009-047; R & D pour la défense  
Canada – Toronto; Mars 2009.

**Introduction ou contexte :** Le but du Projet de démonstration de technologies – Aide à la décision des commandements interarmées pour le XXI<sup>e</sup> siècle (PDT ADCI 21) est de démontrer qu'un cadre collaboratif réseaucentrique peut permettre à un commandement interarmées d'obtenir la supériorité décisionnelle aux niveaux stratégique et opérationnel. Les applications du PDT ADCI 21 sont conçues pour les opérations nationales de courte durée qui peuvent nécessiter une certaine collaboration entre les intervenants civils et militaires pour faire face à l'impact d'une catastrophe naturelle ou causée par l'homme.

Le présent document vise à tirer profit des récents travaux de recherche qui ont identifié la documentation qui pourrait servir de base à l'élaboration d'un cadre d'utilisabilité de haut niveau pour le guide de style du PDT ADCI 2.

Ce travail a été effectué en sous-traitance par Fujitsu Consulting, en collaboration avec RDDC Toronto et Prolity.

**Résultats :** Ce guide de style contient des directives générales sur le développement d'interfaces graphiques personnalisées (GUI) conçues pour être utilisées dans un environnement Windows et dans un environnement de commandement et de contrôle (C2) interarmées.

**Portée :** Le présent document contient des directives préliminaires sur les applications du PDT ADCI 21, qui sont conçues pour être utilisées dans un environnement Windows et dans un environnement C2 interarmées. L'utilisation du guide de style permet une présentation visuelle commune. En se conformant aux directives du présent document sur les facteurs humains, on peut réduire la durée de la formation ainsi que la charge de travail, et améliorer ainsi la performance.

Le but de la version actuelle du guide de style est de fournir des directives aux concepteurs et aux développeurs qui participent au développement itératif des applications du PDT ADCI 21. À mesure que le développement des applications du PDT ADCI 21 progressera, de nouvelles versions de ce guide de style fourniront des recommandations officielles sur les GUI.

**Recherches futures :** Un processus de développement itératif est utilisé pour les applications du PDT ADCI 21. Le contenu du guide de style du PDT ADCI 21 continuera d'être mis à jour en fonction des résultats de ce développement itératif.

# Table of contents

---

Abstract .....	i
Résumé .....	i
Executive summary .....	iii
Sommaire .....	iv
Table of contents .....	vi
List of tables .....	ix
1 Introduction.....	1
1.1 Background .....	1
1.2 Objective .....	1
1.3 This Document .....	1
2 Approach.....	3
2.1 General .....	3
2.2 Guideline Format.....	3
2.3 Conventions.....	4
2.3.1 Use of Shall Versus Should.....	4
2.3.2 Operators versus users.....	4
2.3.3 Representation of keys .....	4
2.4 Source Documents.....	4
3 Design Decision Filters.....	7
3.1 General .....	7
3.2 User population .....	7
3.3 End User Skills and Experience .....	7
3.4 Environmental Conditions of Use .....	7
4 Design Guidance.....	8
4.1 General .....	8
4.2 High Level Design Goals .....	8
4.2.1 Consistency .....	8
4.2.2 Interoperability.....	9
4.2.3 Usability .....	9
5 Guidelines .....	12
5.1 General .....	12
5.2 Input Devices.....	12
5.3 Handheld Devices and Small Screen Displays.....	17
5.3.1 Navigation and Links .....	17
5.3.2 Page Content and Design .....	21
5.3.3 User Input.....	25

5.3.4	UI Text Guidelines.....	28
5.4	Web Portals and Portlets .....	29
5.4.1	Web Portals .....	29
5.4.2	Web Portlets.....	30
5.4.3	Usability standpoint – Web Applications vs. Portlets.....	31
5.5	Data Entry and Data Display .....	34
5.5.1	General Guidelines.....	34
5.5.2	Data Entry and Manipulation .....	35
5.5.3	Data Display.....	42
5.5.4	Data Validation and Error Checking.....	60
5.6	Map and Tactical Display Characteristics .....	61
5.6.1	Tactical Displays.....	62
5.6.2	Drill down capability.....	69
5.7	Charts and Graphs .....	71
5.8	Presentation of Information (Scheduling graph) .....	76
5.9	Alarms and Alerts.....	78
5.9.1	Alarms as information systems .....	79
5.9.2	System Feedback.....	82
5.10	Information Coding .....	85
5.10.1	General Guidelines.....	85
5.10.2	Alphanumeric Coding .....	86
5.10.3	Graphical Coding .....	87
5.10.4	Colour Coding.....	88
5.10.5	Coding with other visual techniques .....	90
5.11	Symbology.....	91
5.11.1	General.....	91
5.11.2	Symbol Composition.....	92
5.11.3	Colour Schemes and Intent .....	93
5.11.4	Shape Schemes and Intent.....	94
5.11.5	Symbol Frames .....	95
5.12	Colour.....	96
5.12.1	Colour Sets.....	96
5.12.2	Specific Use of Colours .....	98
5.12.3	Colour Guidelines .....	103
5.13	Security.....	105
6	References.....	106
Annex A	Windows .....	109
A.1	Windows.....	109
A.1.1	Primary Windows .....	109
A.1.2	Secondary Windows .....	112

A.2	Window Design Guidance.....	114
A.3	Navigation and Selection.....	119
A.3.1	Navigation.....	119
A.3.2	Navigation within Windows .....	121
A.3.3	Object Selection .....	123
A.3.4	Object Transfer .....	126
A.3.5	Interactive Controls.....	131
Annex B	Controls.....	134
B.1	General .....	134
B.2	Menus and Navigation.....	137
Annex C	Special Functions and Formats .....	138
C.1	Date/Time and Latitude/Longitude .....	138
C.2	List-to-List Transfers.....	140
C.3	Message Handling .....	141
C.4	Spell Checking.....	144
C.5	Imagery Manipulation .....	145
Annex D	Security and Simulation.....	149
D.1	Security and Simulation Characteristics.....	149
D.2	System Login.....	150
D.3	System Logout.....	152
	List of symbols/abbreviations/acronyms/initialisms .....	154
	Distribution list.....	155

## List of tables

---

Table 1: Symbology Comparison: Symbol Composition.....	92
Table 2: Symbology Comparison: Colour Schemes.....	93
Table 3: Symbology Comparison: Shapes.....	94
Table 4: Symbology Comparison: Frames .....	95

# 1 Introduction

---

## 1.1 Background

The aim of the Joint Command Decision Support for the 21<sup>st</sup> Century (JCDS 21) Technology Demonstrator Project (TDP) is to demonstrate a Joint Net-enabled, Collaborative Environment to achieve Decision Superiority at the strategic and operational levels of command.

The JCDS 21 TDP will demonstrate a collection of tools and applications across a variety of platforms that provide commanders in Joint, Interagency, Multinational, Public (JIMP) operations with increased situational awareness and decision support capability. These tools will rely heavily on the intelligent filtering, sorting and representation of large data sets from multiple sources. As they may be utilized in time-sensitive and critical situations all tools should display information efficiently and have Graphical User Interfaces (GUIs) that are both intuitive and consistent with commanders' information requirements. In addition, GUI designs associated with the collection of tools or applications should be consistent and, where possible, should adhere to standard symbology and 'best practices' that are relevant to the command and control (C2) community.

The intent of the JCDS21 Human Factors Style Guide is to provide guidance for the design of future Canadian joint applications. The preliminary guidance currently documented in this style guide provides information that can be considered by designers and developers involved in the iterative development of the JCDS 21 applications. As the development of the JCDS 21 applications continues to progress, subsequent versions of this style guide are expected to be developed in order to provide formal recommendations associated with the GUI design. To that end, the style guide is a 'living' document that will continue to evolve in accordance with new design requirements. Adherence to the guidelines housed in the style guide will improve human performance and reduce training requirements by ensuring a consistent and usable design of the JCDS 21 TDP tools and applications.

## 1.2 Objective

CAE Professional Services (Canada) was contracted by Fujitsu to provide a structured set of human factors (HF) guidelines that define the overall 'look and feel' philosophy of the GUI concepts across a multitude of designs and platforms that are being developed as part of the JCDS21 TDP as well as to describe the specific design standards that may apply. The objective of the guidelines is to provide consistency both from a 'look and feel' perspective as well as a user interaction perspective.

This work was completed under sub-contract to Fujitsu Consulting and with cooperation from Defence Research & Development Canada (Toronto) (DRDC-Toronto) and Prolity.

## 1.3 This Document

This document describes preliminary HF guidelines for the development of JCDS 21 applications for use within a Windows-based environment and within a joint C2 specific environment. These

guidelines can be used to provide guidance on the implementation of design concepts associated with the JCDS 21 collection of applications that run across a variety of platforms (including handheld devices) including but not limited to Windows XP and Windows Mobile. As the development of the JCDS 21 applications continues to progress subsequent versions of this style guide are expected to provide formal recommendations associated with the GUI design.

This report consists of the following sections and annexes:

1. **Section 1: Introduction.** This section describes the project background and the document overview.
2. **Section 2: Approach.** This section articulates information associated with the high level design goals, design decision filters, conventions, guidance structure and source documents that were used to develop the SG document.
3. **Section 3: Design Decision Filters.** This section presents the design decision filters which represent the system characteristics that determine the nature with which general OMI guidelines are defined for a specific system.
4. **Section 4: Design Guidance.** This section provides an overview of the guidance associated with design GUIs for use in a general Windows-based environment and in a joint C2 environment.
5. **Section 5: Guidelines.** This section presents general guidance pertaining to a GUI design developed for a Windows-based environment.
6. **Annex A: Windows.** This annex contains details associated with the design of windows.
7. **Annex B: Controls.** This annex presents guidance related to the design and implementation of standard controls for the Microsoft Windows environment.
8. **Annex C: Special Functions and Formats.** This annex articulates guidelines associated with specialized functionality such as list-to-list transfers and spell checking. Furthermore, guidelines pertaining to special formats such as date/time and latitude/longitude are presented in this annex.
9. **Annex D: Security and Simulation.** This annex contains detailed information associated with enabling security within a Windows-based environment.

Guidelines presented in the annexes are primarily operating system specific guidelines (as oppose to application specific). Since the focus of the style guide is on mainly JCDS applications, these topic areas were removed from the body of the report and relegated to a series of annexes for completeness.

## 2 Approach

---

### 2.1 General

The intent of the current version of the JCDS21 Style Guide is to provide guidance for designers and developers involved in the iterative development of the JCDS 21 applications. As the development of the JCDS 21 applications continues to progress subsequent versions of this style guide are expected to provide formal recommendations associated with the GUI design.

### 2.2 Guideline Format

The following section describes the format of each guideline that is presented within the style guide. The intention was to impose a consistent and logical format on each guideline to assist the reader in finding the relevant guideline(s) quickly. The format is illustrated in below.

2.2	Colour coding	<b>Source:</b> HFES 200
<b>Guideline(s):</b>		
<ol style="list-style-type: none"><li>1. If colour coding is used, it is preferable to use no more than six colours in addition to black and white.</li><li>2. If six or more colours are used, a persistent legend should be part of the display.</li></ol>		
<b>Discussion:</b>		
<ul style="list-style-type: none"><li>• This maximum does not refer to colours within images or graphical representation also on a display</li></ul>		

The legend for the different sections of the guideline format is as follows:

1. **Guideline Number.** A unique reference number given to each guideline, or set of guidelines, to enable rapid searching for a particular guideline.
2. **Guideline Title.** A short title that summarizes the topic of the guideline(s).
3. **Source.** A reference for the source document(s) from which the guideline(s) was (were) taken from.
4. **Guideline(s).** A list of guidelines relevant to the topic. These are worded as 'shall' (i.e., mandatory) statements.

5. **Discussion.** Where relevant, supporting evidence for, and/or further discussion of, each guideline is presented in this section.

## 2.3 Conventions

### 2.3.1 Use of Shall Versus Should

Many guideline documents draw a distinction between mandatory (or shall) provisions, and recommended (or should) provisions. All of the provisions in this document are written as mandatory and use shall. The intent of the language is not to imply that all guidelines presented in this document are required. The aim is for the system developer to adhere to the guideline presented herein if the design employs a feature described in this document. The language of *shall* rather than *should* was chosen for historical contractual reasons. Historically, and often by training, developers focus on the required provisions and avoid guidance that is not required.

### 2.3.2 Operators versus users

Throughout this document the terms *Users* and *Operators* are used interchangeably to refer to the individuals that operate the JCDS21 applications. The terms users and operators do not refer to members of the design or engineering team.

### 2.3.3 Representation of keys

Representations in this document such as “< >” are used to refer to input device selections (or sequence of selections) assigned to the function described by "< >". Thus, <Shift> means press the Shift key.

## 2.4 Source Documents

This work is guided, in part, by a number of existing standards (e.g., MIL-STD-1472, DEF-STD-00-25, etc.) and style guides that provide general guidance and recommendations on the ‘look and feel’ of an interface (e.g., layout, symbology, interaction methods). The following references are the main source documents that were used to formulate the guidance within this document. As part of the tailoring process, statements from the different source documents have been incorporated as originally written or have been modified as appropriate for the Canadian operating environment or to reflect more recent research. It is important to note that all the following documents cite many other references; however this document does not cite these secondary or tertiary references. Instead, the GUI developer is encouraged to read the source documents if further clarification or information is required.

1. **Microsoft Windows User Experience (MSWUE).** The official Microsoft Windows User Experience guidelines provide design direction that helps to establish a high quality and consistency baseline for all Windows-based applications. All JCDS21 applications will be required to comply with the Microsoft standard user interaction practices in order to both leverage the operators’ experience with these conventions as well as establish consistency

with existing CF applications. The intent of this style guide is not to replicate the MSWUE guidelines in their entirety since they are commercially available. Designers and developers are encouraged to reference this resource in order to ensure adherence to the necessary guidelines.

Reference Acronym: *MSWUE*

2. **Command Decision Aiding Technology (COMDAT) Operator Machine Interface (OMI) Style Guide.** The COMDAT OMI Style Guide Version 2.0, published in 2004, defines the overall ‘look and feel’ philosophy of the OMI screens and functions to support development of Command Decision Aids for Halifax-Class Canadian Patrol Frigates (CPFs)<sup>1</sup>.

Reference Acronym: *COMDAT*

3. **Human Factors Engineering of Software User Interfaces.** The objective of the HFES standard for Human Factors Engineering of Software User Interfaces (HFES 200) is to consolidate available design guidance to provide design requirements and recommendations that will lead to usability benefits such as increased ease of learning and ease of use of software, and accessibility benefits such as increased compatibility of assistive technology with available Operating System software.

Reference Acronym: *HFES 200*

4. **Human Factors Design Standard.** The Human Factors Design Standard (HFDS) provides reference information to assist in the selection, analysis, design, development, and evaluation of new and modified Federal Aviation Administration (FAA) systems and equipment. This document is predominantly based on the 1996 Human Factors Design Guide (HFDG) produced by the FAA. This standard covers a broad range of human factors topics that pertain to automation, maintenance, displays and printers, controls and visual indicators, alarms, alerts and voice output, input devices, workplace design, system security, safety, the environment, and anthropometry documentation. This document also includes extensive human-computer interface information.

Reference Acronym: *HFDS 2003*

5. **Mobile Web Best Practices – Basic Guidelines.** The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding. The Mobile Web Best Practices – Basic Guidelines specifies guidelines for delivering Web content to mobile devices with the principal objective of improving the user experience of the Web when accessed from mobile devices.

Reference Acronym: *W3C*

6. **Windows Mobile-Based Platforms (WMBP) – Design Guidelines.** The Windows Mobile is a platform for mobile devices based on Windows Embedded CE, and used in a wide variety

---

<sup>1</sup> The OMI guidance from the COMDAT OMI Style Guide was revised to reflect the requirements of the INCOMMANDS OMI.

of third-party hardware such as personal digital assistants (PDAs) and smartphones. The associated design guidelines provide information about developing Windows Mobile-based applications that provide a good user experience.

*Reference Acronym: WMBP*

7. **Portlet Usability Model (PUM).** Portals tend to be constructed by means of portlets (i.e., a multi-step, user-facing application to be delivered through a Web application). In turn, the portlet usability model provides guidance with respect to ensuring understandability, learnability, customizability, and compliance—all dimensions for ensuring usability of portlets.

*Reference Acronym: PUM*

8. **Common Warfighting Symbology (MIL-STD-2525B).** This standard provides common operational symbology along with details on their display and plotting to ensure the compatibility, and to the greatest extent possible, the interoperability of Land Component Command, Control, Communications, Computer, and Intelligence (C4I) systems, development, operations, and training. MIL-STD-2525B addresses the efficient transmission of symbology information through the use of a standard methodology for symbol hierarchy, information taxonomy, and symbol identifiers. These symbols are designed to enhance joint interoperability by providing a standard set of common symbols. It constitutes a single system of joint military symbology for land based formations and units, which can be displayed for either automated map display systems or for manual map marking. It covers all of the joint services and can be used by them.

*Reference Acronym: MIL-STD-2525*

## **3 Design Decision Filters**

---

### **3.1 General**

Design decision filters define a perspective on the design of the software system. The design decision filters represent the system characteristics that determine how general OMI guidelines are defined for a specific system. Design decision filters include the characteristics of the user population, the physical environment, and the devices used to access the software.

### **3.2 User population**

This document is intended for developers who develop software, technical demonstrators, or prototypes to be used by anticipated JCDS21 operational personnel. Operational personnel are not expected to have advanced software skills. They will have training on the system but the training should not be used to replace a usable design.

### **3.3 End User Skills and Experience**

The purpose of this guide is to provide a common framework for OMI and decision support design and implementation. The goal is to promote higher productivity, fewer errors, and reduced training needed to acquire the skills necessary to use the system.

It is anticipated that the end user population will include military personnel and civilian personnel who will respond to short-duration, domestic natural or man-made incidents. Applications, designed in accordance with the style guide, should accommodate users having a range of experience levels such that assistance is provided to novice users and expert users are able to access work-around solutions. Familiarity with a system, how frequently the system is used, and training all affect the design. The design team should ensure that all levels of operator expertise are supported and that the support for one level of expertise does not interfere with the support for operators with different levels of expertise.

### **3.4 Environmental Conditions of Use**

Users will operate JCDS 21 applications within environmental contexts that range from standard office environments to in-field exercises and operations. As such these environments will include low ambient operations room lighting, red lighting used to maintain dark adaptation, standard office lighting, and direct sunlight. The JCDS 21 applications must also accommodate use on hand-held devices and small screen displays within the full range of environmental contexts.

## 4 Design Guidance

---

### 4.1 General

There exist numerous principals fundamental to the design and implementation of effective interfaces for GUI environments. Principles can take the form of design goals (or heuristics) as well as guidelines. These principles can be differentiated as follows:

1. Design Goals (or Heuristics) are the motherhood rules and principles that describe common properties of usable interfaces. They are generally more abstract than traditional guidelines. Consequently, possessing OMI design knowledge and experience helps to accurately understand and interpret them.
2. Guidelines can be construed as good practices within a general design domain (such as Windows in the case of the Microsoft Solutions product portfolio). Holistically, they are based broadly on the usability heuristics. Specifically, they provide useful low-level guidance on the design of usable interfaces in areas such as control design, branding elements, and window behaviour. Since they are generally more specific than heuristics, less design knowledge and experience is required to understand, interpret, and apply them.

While the objective of the Style Guide is to present guidelines to support the design of the JCDS21 applications, the following design goals are presented as guidance that should be adhered to as part of the design process.

### 4.2 High Level Design Goals

The GUI designer has a variety of options in implementing the functionality of a system through the GUI. The challenge is to incorporate these options into a design that results in an optimal experience for the users when using the system to achieve their goals. At a high level, the design of the GUI should adhere to principles surrounding consistency, interoperability, and usability.

#### 4.2.1 Consistency

Consistency is the primary goal of a style guide. JCDS 21 TDP tools and applications should be designed to be consistent; appearing, behaving, and responding the same throughout. Each instance of inconsistency produces unnecessary cognitive processing and affects the cognitive processing available to the user for the achieving their goal. From a high level, compliance with Microsoft UI guidelines will ensure consistency between JCDS 21 applications as well as other Canadian government applications.

Examples of factors that should be considered for consistency include the following:

1. **Language.** Small changes in the language lead to errors and confusion. Users assume that different terms reflect differences in the software. For example, the term 'Close' is expected to result in a different action than 'Exit' so these should not be used to label the same action.

Conversely, if more than one term is used to convey the same concept, then the user must determine if two different terms reflect the same software activity or a different software activity. For example, an application may incorrectly use three notations: Stop, Cease, and End each to mean that the processing will not be continued.

2. **Symbols and Icons.** Using more than one icon design to represent instances of a single type of control will lead to errors and confusion. For example, using a door icon and an X symbol both to indicate Close in an application will lead the user to assume that the X and Close operate differently. Similarly, differences in choices of track or symbology colour (reversing the colour code for Friend and Hostile, for example) will inevitably lead to critical errors.
3. **Controls.** Using different interaction techniques or controls to do similar functions can lead to confusion on the part of users. For example, if one application uses a date picker control to allow a user to enter a date while another application requires the user to enter the date using a text field, it can lead to confusion on the part of the user as to why there is not a consistent method to do the same action across or within applications.
4. **Feedback.** The interface should have a reliable and consistent method of system response across applications. Transactions made by the operator should produce a consistent perceptual response whether it is in visual, tactile, or auditory form.
5. **Operator's mental model.** Display-control relationships must be compatible with the operator's expectations, and require minimum processing to extrapolate the information from the system.

## 4.2.2 Interoperability

JCDS 21 collection of applications and tools must be interoperable. Users are expected to work with many C2 systems over the course of their careers. Differences in the GUI layouts of the systems, the controls, navigation, or presentation can create delays, errors, and confusion. To that end, the overall design should be as consistent as possible with other systems that the operator will be using. This recommendation is not intended to inhibit innovative design concepts. However, the introduction of new concepts should be based on proof that the advantage in terms of reduced training, workload and more efficient interaction with the system more than offsets the requirement to learn a new way of doing business.

## 4.2.3 Usability

Usability is a quality attribute that assesses how easy user interfaces are to use. Furthermore, usability can be viewed as encompassing the following five quality components:

1. **Learnability.** The ease with which users can accomplish basic tasks they first time they encounter a design.
2. **Efficiency of use.** The speed at which users can perform tasks once they have learned the design.

3. **Memorability.** The ease of which users can re-establish proficiency when they return to the design after a period of not using it.
4. **Errors.** The number of errors a user makes, the severity of the errors made, and the ease of which users can recover from the errors.
5. **Subjective satisfaction.** User satisfaction with the usability and functionality of the design.

Usability can also be defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [HFES 200]. Regardless of the definition, maximizing usability of a system helps to improve, amongst other things, operator performance.

Numerous usability heuristics currently exist to guide the design of GUIs [e.g., Nielsen, 1994; Tognazzini, n.d.]. The following usability heuristics [Nielsen, 1994] have been developed to achieve the high level usability goals described above, are widely accepted within the human factors industry and are applicable to JCDS 21 TDP application development. To that end, they are considered to be basic requirements for GUI development and, as such, can be used as overarching guidance throughout the development of JCDS 21 TDP tools and applications.

1. **Visibility of system status.** The system should always keep operators informed about what is going on, through appropriate feedback;
2. **Match between system and the real world.** The system should speak the operators' language, with words, phrases and concepts familiar to the operator, rather than system-oriented terms. Real-world conventions should be followed to make information appear in a natural and logical order;
3. **Operator control and freedom.** Operators often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Undo and redo functionality should be supported where practical;
4. **Consistency and standards.** Operators should not have to wonder whether different words, situations, or actions mean the same thing. Platform conventions should be followed where practical;
5. **Error prevention.** Careful design should prevent problems from occurring in the first place; if errors do occur, appropriate alarms and alerts should be presented to the operator;
6. **Recognition rather than recall.** Objects, actions, and options should be visible to the operator at all times. The operator should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate;
7. **Flexibility and efficiency of use.** Accelerators (e.g., hot-keys), which are unseen by the novice operator, should be used to speed up the interaction for the expert operator. In this

way, the system can cater to both inexperienced and experienced operators. Operators should also be allowed to tailor how they perform frequent tasks (e.g. re-configure windows);

8. **Help operators recognize, diagnose, and recover from errors.** Error messages should be expressed in plain language, precisely indicates the problem, and constructively suggests a solution. Users should be able to navigate away from an error message in the event that the error cannot be resolved; and,
9. **Help and documentation.** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the operator's task, list concrete steps to be carried out, and not be too large.

## 5 Guidelines

---

### 5.1 General

General guidance pertaining to the GUI design developed for a Windows-based environment has been considered during the current analysis. Details associated with existing industry-accepted guidelines, source documents, standards, symbology and colour within this environment are provided in the following sections.

### 5.2 Input Devices

Operators interact with objects in the interface by using different types of input devices. The most common input devices are the mouse and the keyboard. The main input devices for the JCDS 21 applications and tools are anticipated to be the keyboard, touchscreen, and mouse.

1. **Keyboard.** The keyboard is used primarily for entering and editing textual information. However, the Windows interface also supports the use of the keyboard to navigate, toggle modes, modify input, and, as a shortcut, invoke certain operations (see MSWUE Chapter 5 on Input basics).
2. **Mouse.** The mouse will be redundant to the touchscreen for navigating and interacting with the OMI displays. For guidelines on mouse applications see MSWUE guidelines Chapter 5 on Input basics.
3. **Touch screen.** Touch screens are employed to interact with objects in the portable small screen displays and can be used as an input device for the navigating the GUI (i.e., stylus).

5.2.1	Conform to standard criteria for input device selection	<b>Source:</b> COMDAT (4.1) HFDS 2003 (9)
<b>Guideline(s):</b> <ol style="list-style-type: none"><li>1. Two main input devices shall be provided, the keyboard and a pointing device.</li><li>2. The keyboard shall be an extended commercial off the shelf (COTS) design.</li><li>3. The appropriate pointing device shall be determined for the application.</li><li>4. Keyboards and input devices shall be in accordance with good ergonomic design principles.</li></ol>		
<b>Discussion:</b>		

- A pointing device is a non-keyboard device that allows a user to navigate rapidly around the screen and to specify and select objects for manipulation and action. Examples include a mouse, trackball, stylus and grid, and light pen. A pointer is a symbol displayed on the screen that is controlled by a pointing device. Its shape may change depending on the function that is invoked at a particular moment or its location on the screen. [HFDS 2003].
- Standard means for facilitating user interactions with software applications is via a keyboard and input device. The specific design and selection of these devices will vary in accordance with factors such as the platform (e.g., portable vs. non-portable) and operator requirements.
- The advantages and disadvantages of non-keyboard devices are summarized in HFDS 2003, Chapter 9.

5.2.2	Ensure interchangeability among input devices	<b>Source:</b> COMDAT (4.2) HFDS 2003 (9.5)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The pointing device shall be the primary means of user-computer interaction.</li> <li>2. The keyboard shall be available for performing operations, primarily as a backup or for shortcuts.</li> <li>3. If more than one input device is present, a user shall be able to control computer interaction with all of them.</li> <li>4. Navigation within and between the OMI windows, object selection, and other keyboard manipulations shall be consistent with the Microsoft windows style guide.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Operators shall be able to use the keyboard and the pointing device interchangeably in order to interact with the application. Typically, the keyboard is used primarily as a backup so that users can continue to operate a system if the pointing device fails. Accordingly, new hardware designs should be developed that permit the keyboard to be stowed in an easily accessible compartment, thus leaving the work surface free when the keyboard is not in use.</li> <li>• The interchangeability among input devices by the user can be useful during specific operations. Users may want to perform some actions using a keyboard and others actions using a pointing device. The ability to choose which input device must be optional to the user and not a requirement by the system.</li> <li>• Full interchangeability is not required. It is assumed that a user will select the input device that is most appropriate for the task being performed. For example, a user may rely on direct manipulation, using a pointing device such as a mouse or trackball, as the primary</li> </ul>		

means of interaction for object selection and manipulation. Similarly, a user may use a keyboard primarily for text entry and for object selection being performed in conjunction with or interspersed with text entry. However, a keyboard should be capable of executing navigation and selection operations when used in conjunction with a mouse or other input devices.

5.2.3	Provide standard pointer interactions	<b>Source:</b> COMDAT (4.3) HFDS 2003 (8.13.10)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The following basic physical operations shall be supported with the pointing device: Press, Drag, Release, Click, and Double Click</li> <li>2. Double-click capabilities shall be accessible by either single-click or by press and-release operations.</li> <li>3. A separate, explicit, action distinct from cursor position, shall be required for the actual entry (e.g., enabling, actuation) of a designated position.</li> <li>4. The home position for the cursor shall be consistent across similar types of displays.</li> <li>5. When fine positioning accuracy is required, as in some forms of graphic and image processing applications, the displayed cursor shall include an appropriate point designation feature (such as crosshairs).</li> <li>6. The pointer shall have a distinctive visual attribute that does not obscure other displayed entities.</li> <li>7. The pointing device shall be associated with a single pointer on the screen.</li> <li>8. The hotspot of the pointer shall indicate the precise location where operations occur.</li> <li>9. The pointer shall move anywhere on the screen.</li> <li>10. When users move the pointing device, the pointer shall move in the corresponding direction.</li> <li>11. The pointing device-to-pointer movement ratio shall be close to 1:1 for most user interactions.</li> <li>12. The pointer shall remain in place until moved by users; it shall not be moved by an application.</li> <li>13. The pointer shall deviate less than .05 inch in any direction; .01 inch for high stability.</li> </ol>		

<p>14. When a system uses multiple physical displays, the pointer shall move between multiple displays when users move the pointing device.</p> <p>15. When a system uses only one physical display, the pointer shall not move beyond the physical display boundary or disappear from sight.</p> <p>16. The location of the hotspot shall not move as the pointer changes shape. A list of functions for which shapes are defined in the MSWUE guidelines. When the system identifies individual operators (via a login process, for example), then the cursor controls shall be set by the operator. If individual users are not identified then the cursor sensitivity shall be fixed and be compatible with the required task and user skills.</p> <p>17. If the cursor is moved by pressing a key, releasing the key shall cause the cursor to stop moving.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Standard implementation of pointer functionality ensures that operators can leverage their mental model with this input device based on previous experience and avoid confusion.</li> </ul>

5.2.4	Implement standard pointer shapes	<b>Source:</b> COMDAT (4.3) HFDS 2003 (8.13.10)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The MSWUE shall be the primary reference for pointer shapes.</li> <li>2. New pointer shapes shall follow the details of designs and technical guidelines available in MSWUE (MSWUE Chapter 14).</li> <li>3. The applications shall redefine pointer shapes only when the pointer is in an application window.</li> <li>4. An upper-left-pointing arrow shall be used for object selection in most windows.</li> <li>5. An X pointer shape shall not be used by an application.</li> <li>6. New pointer shapes shall not be created for functions that already have a shape.</li> <li>7. Pointer shapes shall not be associated with functions they were not designed to represent.</li> <li>8. New pointer shapes shall be easy to see, with a hotspot that is obvious and easy to locate.</li> <li>9. New pointer shapes shall suggest their purpose and shall not be confusable with other</li> </ol>		

objects.

**Discussion:**

- Position or pointing cursors are used to point to controls on a display. They may at times obscure other screen objects. The pointer cursor is often the left pointing arrow. The pointer is used to make selections and to click in menus and control buttons; to resize windows; to click, hold, and drag objects; and to click on a location to move the location cursor in text and field editing.
- Implementing standard pointer shapes, where possible, ensures conformance with the operator's mental model based on previous experiences for this functionality.

5.2.5	Implement standard button functionality for pointing device	<b>Source:</b> COMDAT (4.3) HFDS 2003 (8.13.10)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. The Select (S) function shall be bound to the left button on a two- or three-button pointing device. This button will be referred to as the S button throughout the remainder of this document.</li><li>2. The Transfer (T) function shall be bound to the middle button on a three-button pointing device. This button will be referred to as the T button throughout the remainder of this document.</li><li>3. The middle button on a three-button pointing device shall be reserved for advance features and shall duplicate features accessible by the primary (S) button.</li><li>4. The Menu function shall be bound to the right button on a three-button pointing device. This button will be referred to as the M button throughout the remainder of this document.</li><li>5. The Transfer function shall be bound to the right button on a two-button pointing device.</li><li>6. Chording (pressing multiple pointing device buttons simultaneously) shall not be used.</li><li>7. Left-handed users shall be permitted to exchange the functions between the left and right buttons.</li><li>8. The Select (S) button shall be considered the primary button. The secondary mouse button (by default the right button on a two-button pointing device) shall duplicate functions already accessible by the primary button.</li></ol>		
<p><b>Discussion:</b></p>		

- Only advanced users employ the middle button on a three-button device. In addition, many users do not use the secondary button on a two-button input device and experience difficulties with the double-click function. Since C2 operations require focused attention, unnecessarily complex control actions (such as the above) should be avoided so that the crewmembers can concentrate on their C2 operations rather than on the OMI itself.

### 5.3 Handheld Devices and Small Screen Displays

Many of the JCDS 21 applications and tools will be displayed on wireless handheld devices to provide up-to-date and relevant information. Some benefits of these devices are to provide maximum flexibility for commanders who have to handle multiple tasks at the same time on many occasions, access to real-time (or quasi real-time) information while away from the office or while travelling. While the benefits of these devices are obvious, the design of applications and tools for small screen displays provide some interesting design problems.

Small screen space can display relatively little data at a given time, resulting in difficulties in using the device for complex tasks. There is a need for techniques that enable a better, more maximized use of the limited screen size while still providing for intuitive use.

In the future small screen displays could be embedded within larger screen displays and therefore, guidance on the small screen designs will be adapted to meet these emerging needs. The following guidelines assume a web browser will be implemented on the handheld devices.

#### 5.3.1 Navigation and Links

5.3.1.1	Minimize Uniform Resource Locator Identifier	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Universal Resource Locators (URLs) of site entry points shall be kept short.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Typing URLs on mobile devices can be difficult, and it is expected that users will prefer to use alternative methods of obtaining URLs when available (e.g., following a hyperlink).</li> <li>• If typing a URL is the only option available, minimizing the length of site entry point helps to reduce the chance of user error when entering the URL.</li> <li>• Examples: <ul style="list-style-type: none"> <li>♦ <a href="http://example.org">http://example.org</a> instead of <a href="http://www.example.org/index.html">http://www.example.org/index.html</a></li> <li>♦ <a href="example.org/example">example.org/example</a> instead of <a href="www.example.org/example.html">www.example.org/example.html</a></li> </ul> </li> </ul>		

5.3.1.2	Minimize Navigation Controls	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Navigation at the top of the page shall be minimized.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Any other secondary navigational element may be placed at the bottom of the page if really needed. It is important the users should be able to see page content once the page has loaded without scrolling.</li> <li>• For example, the navigation can be limited to the homepage of a website. On other pages only include links back to the homepage and back to the last important point along the path users have taken. Show these links at the top and bottom of the page so they're never too far away.</li> </ul>		

5.3.1.4	Implement access keys	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Access keys shall be assigned to links in navigational menus and frequently accessed functionality.</li> <li>2. The same access key shall be provided for links that are repeated across pages such as links to the home page.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Where there is no pointing device, assigning an access key (keyboard short cut) to a link can provide a convenient way for users to access the link and avoid navigating to the link by repeatedly pressing of the navigation key.</li> </ul>		

5.3.1.5	Provide identification of link target	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The target of each link shall be clearly identified.</li> <li>2. The target file's format shall be displayed unless it is known that the device supports it.</li> </ol>		

**Discussion:**

- Users of mobile devices may suffer undue delay and cost as a result of following links. It is important to identify where a link leads so users can make an assessment of whether following it will be of interest to them. If possible give an idea of the size of the resource (in bytes or in an abstract way, e.g., large file).
- Links to content that is in a different format than the originating page (i.e., content that can only be interpreted by other applications or downloads) should be identified so that users are not lead to download content that their device may not support. However, some devices support the rendering of those formats by other applications once downloaded (e.g., music files). Additionally, users may wish to download content for later transfer to other devices altogether. So even if it is known that the user agent does not support a particular content type, that content should still be made available.
- Use clear, concise, descriptive link text to help users decide whether to follow a link. Identify the implications of following a link if the target is notably large and the user might not anticipate this from the context.

5.3.1.6	Ensure proper usage of image maps	<b>Source:</b> W3C
<b>Guideline(s):</b>  1. Image maps shall not be used unless the target client supports them effectively. If only small images can be displayed, break larger images up into smaller sections and deal with them separately.		
<b>Discussion:</b> <ul style="list-style-type: none"><li>• An image map divides an image into regions with associated actions (e.g., clicking an active region of an image map can cause an action to occur). Image maps allow fast navigation provided the requesting device can support the image involved and providing there is a means of navigating the map satisfactorily. Up, down, left, right, and enter are available on most mobile devices, even if there is no pointing device. This is usually sufficient to allow navigation of the active regions of client-side image maps.</li><li>• Many mobile devices lack a pointing device and server-side image maps cannot be used on such devices.</li></ul>		

5.3.1.7	Avoid refreshing, redirection, and spawned windows.	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Pop-ups or other windows shall not appear and shall not change the current window without informing the user.</li> <li>2. Creating periodically auto-refreshing pages shall be avoided unless the user is informed and provided a means of stopping it.</li> <li>3. Using mark-up to redirect pages automatically shall be avoided.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Each of the aforementioned activities is likely to cause user confusion, or add cost and delay to a user’s interaction with the device.</li> <li>• Many mobile devices cannot support multiple windows and consequently, attempting to open another one will have unpredictable results.</li> <li>• Auto-refreshing pages present accessibility problems. In a mobile environment, they may expose the user to undue cost if a page is left open or placed unnoticed in the background.</li> <li>• While redirection is a commonly employed mechanism, it must be remembered that redirection usually requires a round-trip to the browser. This adds a delay on slow links; so use a maximum of one redirect per page and limit the number of pages that are redirected.</li> </ul>		

5.3.1.8	Minimize externally linked resources.	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The number of externally linked resources shall be kept to a minimum.</li> <li>2. The number of images on a page shall be minimized.</li> <li>3. Style information shall be consolidated into a single sheet per page.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Each linked resource (images, style sheets and other objects) requires a separate request across the network. This may add significantly to the load time of the page in the mobile context.</li> </ul>		

### 5.3.2 Page Content and Design

5.3.2.1	Maximize use of page content.	<b>Source:</b> W3C
<b>Guideline(s):</b> <ol style="list-style-type: none"><li data-bbox="250 428 1276 457">1. The content of the page shall be designed to be suitable for use in a mobile context.</li><li data-bbox="250 491 808 520">2. Clear and simple language shall be utilized.</li><li data-bbox="250 554 945 583">3. Content shall be limited to what the user has requested.</li></ol>		
<b>Discussion:</b> <ul style="list-style-type: none"><li data-bbox="276 693 1373 823">• Users in a mobile context are often looking for specific pieces of information, rather than browsing. Content providers should consider the likely context of use of information and, while providing the option to access all information, should offer appropriate information first.</li><li data-bbox="276 844 1373 911">• Use of clear language is of particular importance for mobile delivery, where brevity and directness are generally more desirable than a discursive style.</li><li data-bbox="276 932 1373 1092">• Writing content in the traditional journalistic “front loaded” style can assist users determining whether information is of interest to them and allow them to skip it more easily if it is not. Placing distinguishing information at the beginning of headings, paragraphs, lists, etc. can also help the user contextualize when using devices with limited screen area.</li><li data-bbox="276 1113 1373 1243">• Mobile users often pay for bandwidth, so offering them content that is extraneous to their needs, especially advertising, costs them time and money and contributes to an unsatisfactory experience. In general, the user’s consent should be sought before initiating the download of content.</li></ul>		

5.3.2.2	Size pages appropriately	<b>Source:</b> W3C
<b>Guideline(s):</b> <ol style="list-style-type: none"><li data-bbox="250 1518 993 1547">1. Pages shall be divided into usable but limited size portions.</li><li data-bbox="250 1581 1373 1648">2. The overall size of page shall be designed in accordance to the memory limitations of the device.</li></ol>		
<b>Discussion:</b> <ul style="list-style-type: none"><li data-bbox="276 1751 1373 1780">• If pages are too big, a long time to load the page may occur. Furthermore, mobile devices</li></ul>		

typically have restrictions on the largest page that can be accommodated.

- If pages are too short, the user is required to make multiple requests to read the relevant information. This can lead to an unnecessary delay, since each request typically takes a measurable time to complete.

5.3.2.3	Limit scrolling direction	<b>Source:</b> W3C
<b>Guideline(s):</b>		
1. Scrolling shall be limited to one direction, unless secondary scrolling cannot be avoided.		
<b>Discussion:</b>		
<ul style="list-style-type: none"><li>• The layout of the page should facilitate repeated scrolling in the same direction (axis) to allow the user to experience all its content.</li><li>• If elements on the page (e.g., maps, images) require secondary scrolling, it must not cause the remainder of the page to require this. For example, if an object causes subsequent text to lay out with a significant margin to its left, then this text may not be visible once a user has scrolled past the object. Furthermore, if the presence of an object causes text to render beyond the right boundary of the page then the user will be required to scroll horizontally to read each line of text.</li><li>• If images require presentation larger than the screen size, then consider providing these images on a separate page with a link back to the main content.</li></ul>		

5.3.2.4	Optimize layout of material	<b>Source:</b> W3C
<b>Guideline(s):</b>		
1. Material that is central to the meaning of the page shall take precedence over material that is not.		
<b>Discussion:</b>		
<ul style="list-style-type: none"><li>• Web pages are typically designed with navigational and other elements at the top of or to the side of the page (e.g., menu bars, search functions). This provides a convenient and well-understood navigational metaphor on large displays. However, on small displays this can result in the navigation appearing instead of the actual content of the page when the page is first retrieved.</li><li>• Because it is important for the user to gain an idea of the content of the page on initial</li></ul>		

view, a minimal amount of clutter should be at the top of the page. The user should not have to scroll significantly to find the primary content of the page.

- Menu selections can be placed away from the top of the page with a simple link to the selection at the top of the page. Alternatively, a meta navigation on top of the page with simple text links to major sections of the website can be employed.

5.3.2.5	Avoid relying exclusively on colours	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Information conveyed with colour shall also be available without colour.</li> <li>2. Foreground and background colour combinations shall provide sufficient contrast to ensure readability of the content.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Mobile devices may not have good colour contrast and may be used in less-than-ideal lighting condition. As a result, information that is highlighted in colour may not be visible to users.</li> <li>• If colour is used to indicate a feature then that feature should also be indicated in a way that is not colour dependent.</li> <li>• Avoid using blue or purple text as this may be confused with hyperlinks, especially on devices that do not underline links.</li> <li>• Refer to Section 5.12 for additional guidance with respect to colour usage.</li> </ul>		

5.3.2.6	Ensure readability of foreground when using background images	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. When using background images, the content in the foreground shall remain readable on the device.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Images that are used indiscriminately can lead to content in the foreground that is hard to view, particularly with the limited contrast often found on mobile devices and in hostile viewing conditions (e.g., sunny days).</li> </ul>		

- If background images are used, ensure that the content is readable with and without the background image for devices that do not support them.

5.3.2.7	Employ short page titles	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A short but descriptive title shall be provided for each page.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Provide a descriptive title for the page to facilitate identification. Keep the title short to reduce page weight, and bear in mind that long titles may be truncated. For example, the short page title ‘MSWUE’ could be used as a substitute for the long page title ‘Windows User Experience Interaction Guidelines’.</li> <li>• Many mobile browsers do not display the title of a page. Where the title is displayed the available space may be limited.</li> <li>• The client may use the page title as the default label for bookmarks. Again, space may be limited, so use it to help identify the content and not for other purposes.</li> </ul>		

5.3.2.8	Avoid embedded objects and scripts	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A text equivalent shall be provided for every non-text element.</li> <li>2. Relying on embedded objects or script shall be avoided.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Downloading images to a mobile client adds to the time to display an image and the cost of displaying the page. Making the page readable in text-only mode can help the user assess its usefulness before images arrive.</li> <li>• Many mobile clients do not support embedded objects or script and in many cases it is not possible to load plug-ins to add support. Content must be designed with this in mind.</li> <li>• Even where a device does support scripting, do not use it unless there is no other way of accomplishing your objectives. Scripting increases power consumption and so decreases battery life.</li> </ul>		

- Design pages as though they were to be displayed on a text-only browser.

5.3.2.9	Avoid font related styling	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Reliance on support of font related styling shall be avoided.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Mobile devices often have few fonts and limited support for font sizes and effects. As a result of this, the use of font size, face or effect may not achieve the desired effect.</li> </ul>		

**5.3.3 User Input**

5.3.3.1	Facilitate user input	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The number of keystrokes shall be kept to a minimum.</li> <li>2. Free text entry shall be avoided where possible.</li> <li>3. Pre-selected default values shall be provided where possible.</li> <li>4. A default text entry mode, language and/or input format shall be specified, if the target device is known to support it.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Given the typical input limitations of a mobile device, the interface must as far as possible minimize user input. Where possible, use selection list, radio buttons and other controls that do not require typing.</li> <li>• Where possible use previous entries as defaults</li> <li>• Make it possible to select items using navigation keys and/or numeric input.</li> </ul>		

5.3.3.2	Ensure logical order	<b>Source:</b> W3C
<b>Guideline(s):</b>		
1. A logical order through links, form controls and objects shall be utilized.		
<b>Discussion:</b>		
<ul style="list-style-type: none"> <li>• As the user navigates through the page, present the various fields and objects in a logical order, especially as may of them will not be visible at the same time as the focus item.</li> <li>• Use document order to control layout and tab order.</li> </ul>		

5.3.3.3	Adapt for screen rotation	<b>Source:</b> WMBP
<b>Guideline(s):</b>		
1. Content shall appear and behave properly in landscape mode as well as portrait mode.		
<b>Discussion:</b>		
<ul style="list-style-type: none"> <li>• There are four design approaches that will help achieve compliance with this guideline: <ul style="list-style-type: none"> <li>– <b>Dynamically resize the content.</b> Dynamically resizing content to the dimensions of the client area produces the best user experience with the least number of trade-offs.</li> <li>– <b>Change the content.</b> If the content and controls fit can be displayed properly in one orientation but not the other, consider showing less content in the other orientation.</li> <li>– <b>Change the content layout.</b> Reorganizing controls with a window may be the only option when it is critical to have the same set of controls in each orientation.</li> <li>– <b>Design for a square client area.</b> Design content to fit the visible square common to both portrait and landscape orientations to ensure that it displays properly in either orientation without changes.</li> </ul> </li> </ul>		

5.3.3.4	Employ soft keys and menus	<b>Source:</b> WMBP
<b>Guideline(s):</b>		
1. The left soft key shall display a label even when the right soft key does not display a label.		
2. When a menu command applies to most of the screen elements currently displayed, disable		

it when it doesn't apply. If a menu command applies to only a few of the screen elements, remove it when it doesn't apply.

3. A menu command shall equal to 1 item and a divider shall equal to 1/2 item. The maximum number of commands that should appear in a menu are:
  - a. 11 entries on a Windows Mobile-based Pocket PC to support 240 pixel height.
  - b. 8 entries on a Windows Mobile-based Smartphone to support 176 pixel height.
4. Use of scrolling menus shall be minimized.
5. Submenus shall be used judiciously. Creating a submenu of a submenu shall be avoided if possible.
6. Mnemonics are shown on a menu when a keyboard is present for a Windows Mobile-based Pocket PC. Creating and hand-tuning mnemonics is encouraged. If you do not provide mnemonics, the device will auto-assign them. For more information, see Mnemonics Guidelines.
7. On the Windows Mobile-based Smartphone, numbered access is assigned to menu commands from top to bottom, 1 through 9 or if necessary, 1 through 0. Numbered access is also supported for commands on a submenu; numbering starts at 1. If possible, always assign the same number to a command. Consistent numbering enables users to perform common tasks more quickly.
8. Menu commands shall be organized in the order listed in the WMBP.

**Discussion:**

- The soft keys appear on a soft key bar located at the bottom of the screen. The soft key bar is located at the bottom of the screen and contains two soft key buttons. These buttons display an action and a menu to the user that are context sensitive and can be changed dynamically by an application. For example, in the Contacts list view, the soft keys are **New** and **Menu**. As the user begins creating a new contact in edit view, the soft keys change to **Done** and **Menu**.
- On the Windows Mobile-Based Smartphone, a user selects a soft key by pressing the corresponding hardware button located immediately below the key. On the Windows Mobile-Based Pocket PC, a user selects a soft key by tapping it on the screen.
- In general, the soft key bar should always be present to enable users to access the soft input panel (SIP) on the Windows Mobile-Based Pocket PC.
- Note The soft key bar replaces the menu bar included in previous versions of the Windows Mobile-based Pocket PC

### 5.3.4 UI Text Guidelines

5.3.4.1	Provide error recovery mechanisms	<b>Source:</b> W3C WMBP
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Informative error messages and a means of navigating away from an error message back to useful information shall be provided.</li><li>2. Identify the problem, indicate the cause if helpful, and provide a solution if possible.</li><li>3. Write phrases instead of complete sentences to conserve space. For example, write “Save using a different name” instead of “Save this document using a different file name.”</li><li>4. Title caps shall be used in the title bar of the message box and sentence caps in the message body text.</li><li>5. Bold command names instead of using quotation marks.</li><li>6. When there may be a consequence of a user’s action, preface the error message with the word “Warning.” For example, write “Warning: If you synchronize now, duplicate items may appear in your Inbox.”</li><li>7. Exclamation points shall not be used.</li><li>8. Content that implies that applications can think or feel shall not be used.</li><li>9. Using the word please shall be avoided.</li><li>10. The following alternative terms shall be used for abort, boot, and reboot:<ol style="list-style-type: none"><li>a. Abort – end, quit, or stop</li><li>b. Boot – start or switch on</li><li>c. Reboot – start</li></ol></li><li>11. Phrases shall be utilized for error and informational messages instead of complete sentences in order to conserve space. Example use “Save using a different name” instead of “Save this document using a different file name.”</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• When errors occur, the user should be provided with clear information regarding the fault they have experienced. This should help them to understand whether the fault was temporary or permanent, whether they should retry the attempt to access the content and</li></ul>		

how they may be able to escalate the problem.

- Providing easy navigation away from the error is particularly important in the mobile environment where browsers many not have an easy-to-find “back” button, where contextualization is frequently difficult and where re-entry of URLs as a means of error recovery is particularly difficult.
- Provide the user the ability to escape from the error condition. They should either be able to return to a prior page or to be able to move onwards to a convenient part of the service from where they can retry or alter the transaction they were attempting.
- Title caps rules specify that all words are capitalized with the exception of articles, coordinating conjunctions (and, but, for, not, or, so, and yet), and prepositions containing four or fewer letters.
- Sentence caps rules specify that only the first word and any proper nouns are capitalized.

5.3.4.2	Forms	Source: WMBP
<b>Guideline(s):</b>		
1. Design forms with an appropriate pixel width to accommodate the resolution of the form.		
2. Input controls shall be placed in separate lines instead of horizontally		
<b>Discussion:</b>		
• Form elements such as <INPUT TYPE=”TEXT”> and <INPUT TYPE=”BUTTON”> are not shrunk by the Default menu option of the IE Mobile and are never rendered wider than the width of the screen of the mobile device.		
• Placing input controls in separate lines will prevent the user from having to scroll horizontally to see controls.		

## 5.4 Web Portals and Portlets

Web based applications can extensively employ portals and portlets. The following guidelines provide direction to ensure the usability of this functionality.

### 5.4.1 Web Portals

A web portal, also known as an enterprise information portal, is a framework for integrating information, people and processes across organizational boundaries. It provides a secure unified access point, often in the form of a web-based user interface, and is designed to aggregate and personalize information through application-specific portlets.

Web portals are defined using a group of fundamental features which are described below.

1. **Single Point of Entry.** Portals can provide single sign-on capabilities between their users and various other systems. Command View for example, gives a one point access to most of the situational awareness information required by the CF operators to follow the progression of CF operations all over the world.
2. **Integration.** The connection of functions and data from multiple systems into new components/portlets. Various elements of the J-staff, planning, policy, force generators, allied agencies, OGDs, standard operating procedures, organizational documentation and command direction are integrated into the portal.
3. **Federation.** Individual portals can possess integrated content originating from other portals.
4. **Personalization.** Users can customize the look and feel of their environment. Also refers to the ability to prioritize most appropriate content based on attributes of the user and metadata of the available content.
5. **Permissions.** The ability for portal administrators to limit specific types of content and services users can access. For example, a company's proprietary information can be entitled for only company employee access.

#### 5.4.2 Web Portlets

A portlet is a component of a web portal that provides access to some specific information source or application, such as news updates, technical support, or an e-mail program among many other possibilities. Portals aggregate different content into a single interface; portlets connect the user to specific content within that interface. Most portals offer a selection of portlets that the user can select for a customized interface. A portlet can be thought as a Web component that comprises a full-fledged Web application to be delivered through the portal.

A portlet is a multi-step, user-facing application to be delivered through a Web application (e.g. a portal). Portals aggregate one or more portlets into web pages, which are usually personalized or customized for individual users or groups of users (Linwood and Minter 2004).

Characteristics of portlets include the following:

1. **Portlets generate fragments.** A fragment is a piece of markup (e.g., HTML, XHTML, WML) adhering to certain rules and it can be aggregated with other fragments to form a complete document (Java Community Process 2003). In order to form a portal page, the fragment generated by a portlet is aggregated to the fragments generated by other portlets. The necessary steps in the creation of a portal page include:
  - a. The portlet container receives the content generated by the portlets and hands the portlet content to a portal.
  - b. The portal server adds a title, control buttons and other decorations to the fragment generated by the portlet. As a result, the portal generates a portlet window.

- c. The portal server creates the portal page with the portlet windows.
  - d. The portal server sends the portal page to the client device where it is displayed to the user.
2. **Portlets have modes.** A mode is a way of behaving. Five modes are distinguished, namely (OASIS 2003):
- a. **View.** Portlet should render markup reflecting the current state of the portlet;
  - b. **Edit.** Portlet should provide content and logic that let a user customize the behaviour of the portlet;
  - c. **Help.** Portlet may provide help screens that explain the portlet and its expected usage;
  - d. **Preview.** Portlet should provide a visual sample of how this portlet will appear on the End-User's page with the current configuration; and
  - e. **Custom.** This mode allows the declaration of additional custom modes.
3. **Portlets cannot be deployed alone.** A portlet is to be included within a third application (e.g. a portal framework). This can imply the portlet to be tuned to account for the idiosyncrasies of the hosting portal. Furthermore, the likely rendering of various portlet, simultaneously, makes the functionality of the portlet to be more focused than a traditional, stand-alone Web application.

### 5.4.3 Usability standpoint – Web Applications vs. Portlets

According to Moraga et al. (2007), Web applications and portlets differ from a usability viewpoint in the following manner:

- 1. Web applications are self-sufficient whereas a portlet needs to be included into a hosting portal. As such, usability should not only take into consideration the end-user but also the portal administrator.
- 2. Web applications provide minimal support for modes, whereas this is an essential part of a portlet. Some of these modes are used to facilitate portlet understanding and operational mode which are key issues from a usability perspective.
- 3. Portlets require a greater support for personalization mechanisms as compared to typical Web applications. The portlet needs to be customised to the aesthetic guidelines of the hosting portal. Furthermore, portals, which are the most popular containers for portlets, have personalization as one of their trademarks. A portal should tune services and contents based on the current user profile and therefore impacts portlets.

There are general guidelines that can be applied during the development of web portals and portlets.

5.4.3.1	Ensure understandability	<b>Source:</b> PUM
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Portals shall support different languages</li> <li>2. Portlet functionality shall be described.</li> <li>3. Descriptions and documentation shall be easily accessible to the user</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Understandability refers to the capability of the portlet to enable the user to understand what the portlet is about (Moraga, et. al, 2007). The portal administrator needs to understand the portlet functionality in order to integrate it into the portal.</li> <li>• Attributes of understandability include: interface language, documentation, documentation language, and description.</li> <li>• Understandability measures should be capable of: <ul style="list-style-type: none"> <li>– Evaluating the behaviour of users who have no previous knowledge on portlet operation</li> <li>– Measuring the user’s difficulty in understanding portlet functions, operations, and concepts.</li> </ul> </li> </ul>		

5.4.3.2	Address learnability	<b>Source:</b> PUM
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Guidance on the use of the portlet shall be accessible to the user</li> <li>2. Portlet interface icons shall be easily related to the actions the portlet performs (predictability)</li> <li>3. The presentation of the portlet shall be structured and easy to understand (structured presentation)</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Learnability addresses the capability of the portlet to enable the user to learn how the portlet achieves its aim (Moraga, et. al, 2007). To that end, the portlet must be easy to learn.</li> <li>• Learnability measures should be capable of:</li> </ul>		

- Evaluating the ability of the portlet to provide help in order to assist a user with a starting point of no knowledge
- Measuring the ability of the user to learn how to use the portlet

5.4.3.3	Address customizability	<b>Source:</b> PUM
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The portlet captures and exploits information about the location from which it is accessed.</li> <li>2. The portlet is able to determine and exploit the hardware and software capabilities of the device accessing the application.</li> <li>3. The portlet can adapt itself to different networks; considers adaptation from the network viewpoint, and whether network context information (e.g. bandwidth or packet losses) affects the application</li> <li>4. The portlet considers the need for personalization.</li> <li>5. Window states support space left for portlet rendering. Specifically, the following four window states should be supported:             <ol style="list-style-type: none"> <li>a. Normal: portlet is likely sharing the aggregated page with other portlets</li> <li>b. Minimized: the portlet should not render visible mark-up but it is free to include non-visible data such as JavaScript or hidden forms</li> <li>c. Maximized: portlet is probably the only one being rendered in the aggregated page or that the portlet has more space compared to other portlets in the aggregated page</li> <li>d. Solo: portlet is the only one being rendered in the aggregated page</li> </ol> </li> <li>6. The portlet should provide the end-user with a mode for configuring the portlet. Within this mode, a portlet should provide content and logic that let a user customize the behaviour of the portlet (edit mode).</li> <li>7. The portlet should support categories of users, in that the content generated and displayed within the portlet should be dependent on the category of user who is interacting with the portlet.</li> <li>8. The content of the portlet can be tailored to specific users depending on the configuration (window state, category of users, user profile, user's preferences, etc.)</li> </ol>		

**Discussion:**

- Customizability refers to the attributes of portlet that enable the portlet to be customized by the user, to reduce the effort required to use it and also to increase satisfaction with the portlet (Moraga, et. al, 2007). As such, users must be able to customize the portlet.

## 5.5 Data Entry and Data Display

Interactions with the JCDS 21 applications will require data entry on the part of the user in order to complete the tasks and achieve the goals. In turn, the application will display the data to the user as feedback in response to the user action. The following section provides guidelines for enabling data entry, displaying data and information (i.e., organization of information) and techniques for manipulating and inserting data into the system.

### 5.5.1 General Guidelines

5.5.1.1	Adhere to standard data display and entry characteristics	<b>Source:</b> HFES 200 COMDAT (12.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The design of each window shall be consistent with the task being performed by users.</li> <li>2. If push buttons are used in a data entry window, they shall be separate from the main body of the window and shall be located at the bottom of the window.</li> <li>3. If the data extend across several pages, related data shall be placed on the same page.</li> <li>4. The format for the data display shall be compatible with the format used for data entry.</li> <li>5. When users work from a hard copy form, the window format has an identical format, unless a tailored soft version is demonstrably more effective.</li> <li>6. Default values shall be used to reduce user workload. Defined default values shall be displayed automatically in their appropriate data fields with the initiation of a data entry transaction and the user shall indicate acceptance of the default.</li> <li>7. The user shall be able to replace any default value during a given transaction without changing the default definition.</li> <li>8. Where a series of default values have been defined for a data entry sequence, the experienced user shall be allowed to default all entries or to default until the next required entry.</li> <li>9. Field labels shall be distinctively presented such that they can be distinguished from data</li> </ol>		

<p>entry. Labels for data entry fields shall incorporate additional cueing of data format where the entry is made up of multiple inputs.</p> <p>10. Complex formats and embellishments that do not convey useful information shall be avoided.</p> <p>11. The format shall be appropriate to the user’s level of training and experience.</p> <p>12. When appropriate, users should be able to select alternative styles of presentation (for example, graphical or text).</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Laying out information in a consistent and logical format helps to improve the efficiency with which operators can complete the task at hand.</li> <li>• Provision of default values help to reduce operator workload since the operator may not be forced to determine an appropriate value and enter the data. Furthermore, the probability of user error is diminished.</li> </ul>

## 5.5.2 Data Entry and Manipulation

Data entry functions shall be designed to establish consistency of data entry transactions, minimize input actions and memory load on the operator, ensure compatibility of data entry with display, and provide flexibility of operator control of data entry.

5.5.2.1	Characteristics	<b>Source:</b> HFES 200, COMDAT (12.2.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Data entry shall be consistent with the Microsoft Windows styles.</li> <li>2. A data entry window shall include a title that describes the contents of the window.</li> <li>3. Display formats shall be consistent within a system. When appropriate for users, the same format shall be used for input and output. Data entry formats shall match the source document formats.</li> <li>4. Data entry functions shall be designed to establish consistency of data entry transactions, minimize input actions and memory load on the user, ensure compatibility of data entry with data display, and provide flexibility of user control of data entry.</li> <li>5. Data entry shall be paced by the user, rather than by the system.</li> <li>6. The same set of actions to enter and manipulate data shall be used throughout the application.</li> </ol>		

7. The system shall provide a positive feedback to the user of the acceptance or rejection of entered data, if appropriate.
8. Feedback shall be provided which presents status information, confirmation, and verification throughout the interaction.
9. If the system rejects a user input, feedback shall be provided to indicate the reason for rejection and the required corrective action. Feedback shall be self-explanatory.
10. Editing commands such as Cut, Copy, and Paste shall be available if these commands shall facilitate the text entry process.
11. The default for text entry shall be insert mode.
12. When variable length information is entered in a text field, it shall be automatically justified. Text shall be left justified. Numeric data shall be justified on virtual or literal decimals.
13. Leading characters shall not be required to fill data entry space. For example, if a text field has a maximum length of four characters and the user wants to enter the number 900, the system shall not require the user to enter 0900. The leading character of 0 shall not be required (except as necessary for compass headings or similar conventions).
14. Users shall be able to enter numeric data via the keyboard and the numeric keypad.
15. Users shall not have to enter the unit of measurement associated with a numeric value.
16. Data shall be entered in units that are familiar to the user.
17. When multiple data items are entered as a single transaction, the user shall be allowed to re-enter, change, or cancel any item before taking a final enter action.
18. When users finish making entries in the appropriate data fields in a window, they shall enter the data into the system with an explicit action such as selecting a save option.
19. A data entry window shall contain the controls needed to support data entry and manipulation.
20. If users have to make multiple entries, the data entry window shall include controls to clear the window and restart the data entry.
21. Dialogue types (e.g., form filling, menus) such as forms and menus, shall be compatible with anticipated task requirements and user skills.
22. Groups of data shall be displayed vertically, not horizontally, although columns may be used.
23. If information being entered in a text field is a known fixed length, the text field shall be

the same length.

24. If the length of information being entered in a text field varies, the text field shall be as long as the longest possible entry.
25. A text field may include scroll bars if space in the window is limited.
26. The text field shall include a decimal point if numeric data are entered in decimal format.
27. Field formats shall be consistent with users' expectations and presented in meaningful chunks.
28. Data that are known or can be computed shall be automatically entered in a field.
29. Data shall not be entered by overwriting a set of characters in a field (such as a default).
30. When an item length is variable, the user shall not have to remove unused underscores.
31. Data entry fields that do require the operator to enter non-word character strings over four characters shall be divided into equal segments of no more than four characters each. If possible the groups shall be equal in size and have meaning (e.g., time HH:MM:SS).
32. Routine or default data shall be automatically entered in a field. If this data could produce a detrimental effect, it shall not be automatically entered or must be confirmed by the operator.
33. Where no source document or external information is involved, forms shall be designed so that data items are ordered in a logical sequence for input.
34. Form filling interactive control may be used where some flexibility in data to be entered is needed and where the users will have moderate training. A form-filling dialogue shall not be used when the system must handle multiple types of forms and the system response is slow.
35. Displayed forms shall be arranged to group related items together.
36. The capability shall be provided to label and store frequently used text segments, forms, and templates (e.g., signature blocks, organizational names, call signs, coordinates), and later to recall (copy into current text) stored segments identified by their assigned labels.
37. Actions that affect the items on the situation display (or tactical display) shall provide visual feedback to the operator that the action was completed as intended.
38. When operators access information in order to alter the data, the window shall initially show the current state of the information.
39. Data entry shall be supported by standard widgets that facilitate data entry and avoid keyboard entry by the operators. Examples include widgets such as spin boxes for numbers

<p>and calendars for dates.</p> <p>40. Widgets implemented to support data entry (e.g., spin boxes for numbers and calendars for dates) shall not replace the manual entry of data.</p> <p>41. To reduce workload and the likelihood of input errors, information should be automatically entered into data fields whenever possible.</p> <p>42. Except for extended text, the length of individual data items should be minimized.</p> <p>43. Data entry shall require an explicit completion action, such as pressing an ENTER key.</p> <p>44. Data should be entered in units that are familiar to the user.</p> <p>45. Data deletion or cancellation shall require an explicit action, such as depressing a DELETE key</p> <p>46. The amount of keying required should be minimized.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion required given the nature of the guidelines.</li> </ul>

5.5.2.2	Cursor Pointer Control	<b>Source:</b> HFES 200, COMDAT (12.2.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The text pointer shall be placed and moved with both the pointing device and the keyboard.</li> <li>2. When a window has input focus, the text pointer shall appear in the text area where the typing is most likely to occur.</li> <li>3. When the Select button is clicked in an empty text field, the text pointer shall appear at the first character space in the text field and the pointer shall remain displayed in insert mode.</li> <li>4. The pointer shall disappear from the screen when users begin typing and shall reappear when users stop typing or move the pointing device.</li> <li>5. The pointer shall be an I-beam shape only when users move the pointer into a text field where the text entry is possible.</li> <li>6. The text pointer shall only be allowed to be placed in areas where the text entry is possible.</li> <li>7. The text entry shall not be possible when the text pointer is not visible and shall only be</li> </ol>		

possible when the text pointer is visible.

8. Applications shall ensure that the text pointer is highly visible when it appears in a text entry area.
9. When the pointer is on non-editable text, its shape shall not change to an I-beam.
10. Clicking on non-editable text shall not change its appearance or display a text pointer.
11. Systems employing cursors shall provide cursor control capability. The user should be able to adjust the sensitivity of the cursor movement to be compatible with the required task and user skills.
12. The response of a cursor to control movements shall be consistent, predictable, and compatible with the user's expectations. For cursor control by key action a key artefact with a left- pointing arrow should move the cursor leftward; for cursor control by joystick, leftward movement of the control should result in leftward movement of the cursor.
13. Fixed function keys (e.g., ENTER) should be used for time-critical, error-critical, or frequently used control inputs.

**Discussion:**

- The pointer cursor is often the left pointing arrow. The pointer is used to make selections and to click in menus and control buttons; to resize windows; to click, hold, and drag objects; and to click on a location to move the location cursor in text and field editing.
- Implementing standard pointer shapes, where possible, ensures conformance with the operator's mental model based on previous experiences for this functionality.

5.5.2.3	Support keyboard navigation	<b>Source:</b> COMDAT (12.2.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Users shall move the text cursor between single line text fields by pressing &lt;TAB&gt;</li><li>2. Applications shall not automatically tab to the next field. (Developer note: Automatic tabbing shall be restricted to situations where data entry occurs in fixed-length entries. Developers shall not combine auto tabbed and manually tabbed fields in the same data entry window.)</li><li>3. When the text pointer is tabbed into an empty text field, the text pointer shall appear in the first character space at the left end of the text field. (Developer note: This occurs only if the text field is empty. If there is existing data in the field, then the pointer tabs to its previous</li></ol>		

location the last time the field was active.)

4. Users shall also be able to place the text pointer at any location within the text in a text field by positioning the pointer and pressing the Select button.
5. The size of the text pointer shall change to match the size of the text font in which it is placed.
6. In a multi-lined text field, the arrow keys on the keyboard shall move the text one increment (i.e., one line or one character) in the specified direction
7. In a multi-lined text field, <Ctrl> in combination with the arrow keys shall move one large increment (i.e., one word or one paragraph) in the specified direction.
8. In a multi-lined text field, <Home> and <End> shall move the text cursor to the beginning and end of a line.
9. In a multi-lined text field, <Ctrl> + <Home> and <Ctrl> + <End> shall move the text cursor to the beginning and end of the text file.
10. If multi-line text is in a window with a default action, <Enter> or <Ctrl> + <Return> shall execute the default action.
11. In a multi-lined text field, <Return> shall insert a carriage return.
12. In a single-lined text field, the arrow keys on the keyboard shall navigate to a different component.
13. In a single-lined text field, <Ctrl> in combination with the arrow keys shall move one large increment (i.e., one word or one paragraph) in the specified direction.
14. Users shall be provided the capability to back up to any text field and edit the entry prior to entering data into the system.
15. Selecting a text field with the pointer shall make that the focused or active field and place the cursor at the beginning of the text field. The entire field may be selected upon initial entry if desired by the designer.
16. A displayed pointer shall be positioned by the system at the first data entry field when the form is displayed. The cursor shall be advanced by a tab key to the next data entry field when the user has completed entry of the current field or shall automatically move to the next field when the end of the field is reached. The default value for the automatic movement of the cursor shall be off.

**Discussion:**

- As an alternative to the pointing device, navigation and data entry in the GUI shall be possible via the keyboard. This provides both a backup mechanism as well as another

user option (and possible preference) for interacting with the interface.

5.5.2.4	Text Field Manipulation	<b>Source:</b> COMDAT (12.2.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Users shall be provided the capability to easily specify the format of a document and the font type, size, and style.</li><li>2. Automatic line break and word-wrap at the right margin shall be available.</li><li>3. Automatic pagination (page numbers based on the number entered by users) shall be available.</li><li>4. The window shall be formatted as the printed output, or an option to see this format shall be provided.</li><li>5. A copy of the original document shall be retained until users confirm that it is to be changed.</li><li>6. The original document shall not be modified automatically as users make each editing change.</li><li>7. Applications shall provide both Find and Find and Replace capabilities for users. The Find and Find and Replace capability shall be in designed in accordance with the design guidance in MSWUE styles. In a Find capability, users shall type the text string (case-sensitivity shall be optional) and the first instance is highlighted. In the Find and Replace capability, users shall type the text string (case-sensitivity shall be optional) and the first instance is highlighted or if requested a global replace-all capability shall make all the changes requested.</li><li>8. Where text formats are a user option, a convenient means shall be provided to allow the user to specify and store for future use the formats that have been generated for particular applications.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Providing the flexibility with the presentation of the information ensures that the user remains in control of this capability in order to suit personal preferences and/or task requirements.</li></ul>		

### 5.5.3 Data Display

5.5.3.1	General guidelines	<b>Source:</b> COMDAT (12.3.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Multi-coloured text shall not be used as a code to sort data.</li><li>2. Multiple colour-codes shall not be used unless the colours are already associated with specific meanings.</li><li>3. Every third or fifth row of data shall be separated by a delimiter. The delimiter may include light background shading in groups of three to five rows (similar to old-fashioned computer paper), separator lines, or other means consistent with the application.</li><li>4. Information shall be automatically justified.</li><li>5. Justification rules shall be followed: Left-justify alphabetic data; right-justify integers; justify decimal data on the decimal point.</li><li>6. When five or more alphanumeric characters without natural organization are displayed, the characters shall be grouped in blocks of three to five characters within each group separated by a minimum of one blank space or other separating character such as a hyphen or slash.</li><li>7. Descriptive wording shall be employed when labelling data fields; use of arbitrary codes shall be avoided.</li><li>8. The ordering and layout of corresponding fields shall be consistent and have the same names.</li><li>9. Users shall not be required to translate feedback messages by use of reference system or code sheets. Abbreviations shall be avoided.</li><li>10. Only data essential to the user's needs shall be displayed.</li><li>11. Data presented to the user shall be in a readily usable and readable form such that the user does not have to transpose, compute, interpolate or mentally translate into other units, number bases, or languages.</li><li>12. A text window shall be wide enough to display an entire line of text without scrolling. Text windows shall be no wider than 40-60 characters.</li><li>13. Text fields that represent real data shall be distinguishable from text fields that represent test or simulated data.</li><li>14. If a specific format is required, an example of that format or an example entry shall be</li></ol>		

displayed.

15. Text fields that require certain text formatting shall provide this formatting automatically (e.g. convert small case to CAPS where required).
16. The operator shall be given visual cues for unacceptable entries, required entries, and the location of the present entry field.
17. Once an entry has been made, the operator shall not have to re-enter this information but rather it shall be selectable (e.g. once the operator enters a name the name can be selected from an option menu or list box).
18. A name entered into an option menu or list box the entry shall be able to be deleted at the request of the user.
19. Operators shall not have to memorize codes for objects with commonly known names unless the code is also commonly known by both novice and experienced users (e.g. do not have a code for selecting a certain missile, instead use that missile's name and have the system provide the required codes).
20. The user shall not have to rely on memory to interpret new data; each data display shall provide needed context, including recapitulating prior data from prior displays as necessary.
21. The user shall not be required to learn mnemonics, codes, special or long sequences, or special instructions.

**Discussion:**

- To “minimize user memory load”, systems should take over the burden of memory from the user. Systems are very good at remembering things very precisely whereas humans are not. This can be accomplished by promoting recognition over recall. In general, people have an easier time at recognizing something that is shown to them rather than having to recall the same information from memory without help.
- Whenever users are asked to provide input, the system should describe the required format and, if possible, provide an example of legal input. This also helps to minimize user memory load

5.5.3.2	Organize data to optimize performance with the system	<b>Source:</b> COMDAT (12.3.2)
<b>Guideline(s):</b> 1. When data fields have a naturally occurring order (e.g., chronological or sequential), such		

order shall be reflected in the format organization of the fields.

2. Displayed data items that are critical or require immediate user response should be grouped at the top of the display.
3. Sets of data that are associated with specific questions or related to particular functions may be grouped together to signify those functional relationships.
4. Data items used more frequently than others may be grouped at the top of the display.
5. Data fields shall be organized by sequence of use, frequency of use, or importance.
6. Displayed data items that are critical or require immediate user response shall be grouped at the top of the display.
7. Sets of data that are associated with specific questions or related to particular functions shall be grouped together to signify those functional relationships.
8. Data items used more frequently than others may be grouped at the top of the display.
9. Data fields shall be organized with related fields together and unrelated fields separated.
10. Data fields to be compared on a character-by-character basis shall be positioned one above the other with alignment of characters to be compared.
11. Users shall be able to obtain information about a data field and its contents.
12. Data field names shall be unique; the same names shall be applied throughout the application.
13. Required data entry text fields shall be distinguishable from optional data entry text fields.
14. Data fields with values that cannot be changed shall be displayed in a read-only information area.
15. Location of recurring data shall be similar among all data displayed and common throughout the system.
16. Long strings of numbers shall be delimited with spaces, commas, or slashes (not leading zeros).
17. Strings of alpha-numeric shall be grouped into sets of three to five characters or grouped at natural breaks. When a code consists of both letters and digits, common character types shall be grouped by common character type for ease of location.
18. If a data record extends beyond a line and the entire line of data must be displayed at all times, each line beyond the first line shall be identified as a continuation of the record.
19. If a data record extends beyond a line and the entire line of data does not need to be

<p>displayed at all times, a method shall be designed so that users can select an individual data record and view all the data fields available for that record.</p> <p>20. Information shall be presented (or requested) only to an operationally valid level of detail (tactical significance); unnecessary detail shall be removed from the display.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• By applying human perceptual and memory characteristics to interface design, the amount of work an operator must exert in order to understand the information being presented can be reduced and allow the operator to focus on important information.</li> </ul>

5.5.3.3	Adhere to perceptual principles	<b>Source:</b> HFES 200
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Information (elements) shall be organized by proximity, that is, near objects belong together. This can be done by organizing according to distance.</li> <li>2. Information (elements) shall be organized by density proximity. The percentage of the screen filled with text and graphics. For example, on content (i.e., text) pages, use some white space to separate and group content. Too much separation of items may require users to scan and scroll unnecessarily.</li> <li>3. Similar elements and information shall be organized together. Like objects belong together (size, shape, colour, brightness, orientation). Each group has similar elements and makes them distinct from one another (colour, proximity, font).</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Human eyes and minds see objects as belonging together if they are near each other in space.</li> </ul>		

5.5.3.4	Ensure aesthetics	<b>Source:</b> HFES 200
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Information shall be balanced on the display to provide stabilization or equilibrium. Create screen balance by providing an equal weight of screen elements, left and right, top and bottom</li> <li>2. Symmetry should be provided among elements and information on the screen. For</li> </ol>		

example, axial duplication; a unit on one side of the center line is exactly replicated on the other side can create balance, but unlike balance, where balance can be achieved without symmetry (i.e. Google site). Create symmetry by replicating elements left and right of the screen centre line

3. Uniformity of elements should be provided based on some principle or plan (regularity). Create regularity by establishing standard and consistently spaced horizontal and vertical alignment points.
4. Information on the display shall be economized to enhance simplicity and usability.
5. Similar element sizes, shapes, or colours shall be used for related information to create unity among elements.
6. Elements and information shall be sequentially placed on the screen with the most important information significantly placed. This will guide the eye through the screen in a logical, rhythmic order,

**Discussion:**

- Opposite of balance is instability, the screen elements are seemingly ready to topple over.
- Irregularity is when there is no apparent plan or principle.
- The opposite is intricacy, the use of many elements just because they exist. Economy means mobilizing just enough display elements and techniques to communicate the desired message, and no more.
- Economy: is the frugal and judicious use of display elements to get the message across as simply as possible. Economy can be achieved by using as few styles, display techniques, and colours as possible.
- Unity: is coherence, a totality of elements and information that is visually all one piece. With unity, elements seem to belong together, and are seen as one thing.
- The eye tends to move sequentially from:
  - ◆ Dark areas to light areas
  - ◆ Big objects to little objects
  - ◆ Unusual shapes to common shapes
  - ◆ Highly saturated colours to unsaturated colours

5.5.3.5	Guide/direct attention/eye sight	<b>Source:</b> HFES 200
<b>Guideline(s):</b>		

<ol style="list-style-type: none"> <li>1. An obvious starting point shall be provided in the screen's upper-left corner</li> <li>2. The most important and most frequent used elements or controls shall be located to the top left</li> <li>3. A top-to-bottom, left –to-right flow shall be maintained.</li> <li>4. The eye shall be guided by forming lines with white space and display elements.</li> <li>5. Elements and information shall be aligned to create simplicity. For instance, minimize the alignment points, especially horizontal or columnar.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• When groups of related info must be broken and displayed on separate screens, provide breaks at logical or natural points in the information flow.</li> <li>• Complex movements may require the aid of display contrasts.</li> <li>• Borders provide visual cues concerning arrangement of screen elements as the eye tends to stay within a border to complete a task. Sequence of use can be made more obvious through the use of borders around groupings of related info or screen controls.</li> <li>• Aligning elements will minimize screen scanning and navigation movements.</li> </ul>

5.5.3.6	Ensure legibility and readability of text	<b>Source:</b> COMDAT (12.3.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. It is essential for successful operations that text displayed to the operators shall be legible and readable under operational lighting conditions and at over-the-shoulder viewing distances.</li> <li>2. Characters shall be legible.</li> <li>3. Contextual text shall be readable.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Legibility refers to the identification of single characters and is a function of letter height, stroke width, height to width ratio, contrast ratio, and polarity. If standard fonts are used, character height is the most important factor when designing so that the characters are legible.</li> <li>• Readability is the ability to recognize groups of letters or words that have contextual</li> </ul>		

meaning. If the letter, word, or line spacing of text is too close, user will have trouble recognizing words. If the spacing is too large, reading performance will slow. The text line length also determines the speed and ease of reading.

5.5.3.7	Accommodate luminance/contrast/colour demands	<b>Source:</b> COMDAT (12.3.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Normal, dimmed, and highlighted text shall appear and behave as indicated in MSWUE.</li> <li>2. Black shall be used for text in windows.</li> <li>3. Fully saturated blue text shall not be used, as it is difficult to perceive.</li> <li>4. White text shall not be used on black backgrounds as irradiation causes characters to spread out and lose sharpness.</li> <li>5. Light text on dark backgrounds may require slightly less stroke width than dark text on light backgrounds. This is due to irradiation, which is a phenomenon that causes white features on a black background to appear to spread out. Any polarity advantages seem to lean toward dark text on light backgrounds.</li> <li>6. Either the characters or the background shall produce a minimum luminance level of 35 cd/m<sup>2</sup> (10 foot-lamberts).</li> <li>7. The text to background contrast ratio shall be a minimum of 1.5:1 for disabled text and 3:1 for normal text. Contrast ratios of 7:1 to 20:1 or more are preferred.</li> <li>8. Smaller character heights and stroke widths shall be designed with higher contrast ratios to maintain the same perceptibility as larger letters. For example, letters of 10 minutes of arc would need a contrast ratio of 19:1 to provide the same legibility as letters of 20 minutes of arc with a 3:1 (1.86:1) contrast ratio (equivalent contrast ratio formulas).</li> <li>9. If text is other than black, the text colour shall have sufficient contrast with background to be readable.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Note that these guidelines are specified to the Halifax-class CPF control room and must be applied accordingly to the use of the JCDS 21 applications.</li> </ul>		

5.5.3.8	Font/character stroke	<b>Source:</b> COMDAT (12.3.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Sans serif fonts shall be used for all text displayed.</li> <li>2. Use a font equivalent to the windows appearance of MS® San Serif or Tahoma for displaying system generated text in sentence format and labels.</li> <li>3. The title bar text shall use the bold setting equivalent of the basic system font chosen.</li> <li>4. Except for title bars, bold or italic fonts shall not be used for interface text.</li> <li>5. Bold text shall be used in menus to indicate that a command is a default action.</li> <li>6. Bold text shall be used in menus to indicate that a command is a default action.</li> <li>7. The text in each window shall be of sufficient size to be legible by users when they are at an over-the-shoulder viewing distance from the screen. (Note: The COMDAT OMI Style Guide requires that the text be legible at the minimum over-the-shoulder viewing distance (i.e., 27 inches). This more conservative specification benefits both over-the-shoulder viewing and the seated operator.)</li> <li>8. The minimum font size shall subtend 10 minutes of visual arc at the longest viewing distance. The minimum font size for text that requires rapid readability is 16 minutes of visual arc with a preferred size of 20-22 minutes of visual arc.</li> <li>9. For reading lines of text, the maximum font shall be 24 minutes of visual arc. Font heights greater than 24 minutes of visual arc shall only be used for titles or labels consisting of single words or short phases.</li> <li>10. Minimum, preferred, and maximum character heights at a minimal over-the-shoulder viewing distance are respectively 0.13, 0.16, and 0.19 inches. (The calculation for visual angle: Size (in.) = tan(Minutes Visual Arc/60) * Distance (in.)) The OMI shall be designed for the minimum over-the-shoulder viewing distance (27 inches).</li> <li>11. Text character height shall be 1/200<sup>th</sup> of the viewing distance.</li> <li>12. Text character width shall be 50-100 percent of character height.</li> <li>13. The text character stroke width minimum shall be 10-12.5 percent of character height</li> <li>14. Spacing requires one stroke width minimum between all characters and a maximum spacing of one average character width. Spacing between words requires an average character width. Spacing between lines shall normally be the height of an uppercase character but may be reduced to the height of a lowercase letter if space is at a premium.</li> </ol>		

<p>15. Text characters shall contain a minimum 7 X 9 dot matrix construction.</p> <p>16. Text used for briefing presentations shall use at minimum a 10 X 14 dot matrix format, double stroke width.</p> <p>17. Applications with word processing capabilities shall provide a choice of fonts as a user-selectable option.</p> <p>18. The character stroke width of a system font shall be at least 2 pixels in thickness.</p> <p>19. The selected font style shall be legible in various display resolutions from 72 pixels per inch to 120 pixels per inch.</p> <p>20. User font cells shall be the same height as the system fonts that will be used with them. The glyphs within the character cell for a user font shall be positioned at the same location and have the same height as those for the associated system fonts.</p> <p>21. Fixed width fonts shall be used for text field widgets receiving data entry and for system generated non-editable text fields.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• The provisions for text size and contrast as described above were developed for operations under normal office lighting. These provisions may require adjustment for operations under sub-optimal room lighting causing reduced visual acuity and should be considered to be minimum starting points for design. To ensure that the text is appropriate at the over-the-shoulder viewing distance, and appropriate under the ambient lighting conditions, all of the design solutions should be validated and tested under the expected operational conditions.</li> </ul>

5.5.3.9	Grammar/Context	<b>Source:</b> COMDAT (12.3.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Consistent grammatical structure shall be used for all non-editable text in windows.</li> <li>2. Wording shall be consistent and use familiar terms and task-oriented language of users.</li> <li>3. Blocks of text shall be broken into smaller, meaningful groups.</li> <li>4. Continuous text shall be phrased in simple sentences, in the affirmative, and in active voice.</li> </ol>		

<ol style="list-style-type: none"> <li>5. A sequence of events or steps shall be presented in the order the steps are performed.</li> <li>6. The referent for <i>it</i> or <i>they</i> in a sentence shall be easily identified.</li> <li>7. Normal punctuation rules shall be followed, and contractions and hyphenation shall be avoided.</li> <li>8. Editable text shall be displayed in upper/lowercase as appropriate to the task being performed.</li> <li>9. Stored text shall be shown in standard format; text editing is converted into the standard format.</li> <li>10. All labels, text, and titles, acronyms, words and headings shall be spelled correctly.</li> <li>11. All terms shall be used consistently throughout the OMI. Examples include definitions such as “best quality” from different sensors for local and fused data, terms such as “Platform”, and actions such as “clear alarms”.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion required given the nature of the guidelines.</li> </ul>

5.5.3.10	Labels	<b>Source:</b> COMDAT (12.3.4); W3C; WMBP
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Colons shall be included when static text is used to label another control.</li> <li>2. The baselines of text labels shall be aligned with the baselines of the text boxes.</li> <li>3. The presence and location of control input data entered by the user shall be clearly and appropriately indicated. Data displayed shall not mislead the user with regard to nomenclature, units of measure, sequence of task steps, or time phasing.</li> <li>4. The label shall be followed by a colon and two spaces before the text (Label: [space][space]text).</li> <li>5. If a label pertains to a group of text, this group of text shall be offset from the rest of the text by an outline or spacing. If an outline box, or separator line, is used, the label shall be integral with the outline box or separator line.</li> <li>6. Labels shall be simple, concise words or phrases in terms familiar to the operator.</li> </ol>		

7. Labels for push buttons, pull-down menu selection, or any other items that perform an action shall be verbs. Labels for radio buttons, checkboxes, lists or any items that present options shall be nouns. Labels for groups of components shall be nouns. See MSWUE for examples of noun/verb and label combinations
8. If a unit of measurement is always used, it shall be part of the label and does not have to be entered.
9. If text field labels appear to the left of their text fields, text field labels shall use a colon (:) to separate the label from the text field.
10. The label for a text field shall not display a shadow.
11. Text labels of more than one word shall not be run together, nor shall they be separated by an underscore, or any delimiters other than a single space.
12. Required data entry text fields shall be distinguishable from optional data entry text fields. A colour-coded one-pixel border may be placed around a text field. By limiting the border resource to one pixel, it will not interfere with the highlight border. The presence or absence of the border may be used to distinguish required or optional data.
13. A text field label shall appear to the left or above the text field and describes what is to be entered or what is to be displayed.
14. A text field label of a text field shall include cues regarding expected format of the entry.
15. Text field labels shall have the same background colour as the window on which they appear.
16. Labels shall be consistently located adjacent to (and preferably above or to the left of) the data group or message they describe.
17. Labels shall be unambiguously related to the group, field, or message they describe.
18. Labels shall be visually distinct from other text and the accentuating technique shall be different and easily distinguished from the method used to highlight or code emergency or critical messages.
19. Labels shall be unique and meaningful to distinguish them from data, error messages, or other alphanumeric.
20. Each text field, editable or non-editable, shall have a label located to the top or left of the field.
21. When data-entry fields are very long the label may be placed above the left edge of the field.
22. The label shall not end with a colon when the text box or drop-down list is embedded in a

sentence or when the drop down list appears in a main window. (WMBP)

**Discussion:**

- For handheld devices, use the label element in HTML or its equivalent in other languages. Furthermore, keep the label position consistent and close to the control so re-flowing or adapting the content intelligently will always recognize label controls and keep them together.

5.5.3.11	Justification	<b>Source:</b> COMDAT (12.3.5), HFES 200
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. All standard text and columns of text shall be left justified. All lines of text shall be wrapped to fit in the provided text window even if this window is resized. This eliminates the need for the operator to horizontal-scroll through text.</li><li>2. Tabular data and tables shall not be wrapped. Vertical scrolling shall be used to access text not displayed in the text window.</li><li>3. An automatic word wrap (carriage return) shall be provided when the text reaches the right margin for entry/editing of unformatted text. User override shall be provided.</li><li>4. Numeric data without decimals shall be right justified. Numeric data with decimals shall be justified on the decimal point.</li><li>5. Labels in columns shall be left justified with the associated alphanumeric text being left justified after the longest label. Labels shall be positioned to read left to right and/or top to bottom. Labels shall be of approximately equal length to facilitate associating the label with the text field.( note: the requirement that the labels shall be of approximately equal length was added.)</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Proper use of justification helps to improve visual scanning.</li><li>• Justification as defined in the guidance is in accordance with languages that read from left to right.</li></ul>		

5.5.3.12	Avoid leading zeros unless required	<b>Source:</b> COMDAT (12.3.6)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Leading zeros shall not be displayed unless required for clarity. However, leading zeros may be displayed during data input to ensure the user did not accidentally leave out a digit.</li> <li>2. Leading zeros shall be displayed for headings, bearings, and other compass references normally displayed in three digit groups.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Reducing the presentation of leading zeroes, unless part of a standardized practice, reduces visual clutter on the screen.</li> </ul>		

5.5.3.13	Numbers	<b>Source:</b> COMDAT (12.3.7)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. When displaying numbers the number zero shall have a slash through it so it is not confused with a capital “O”. The letter L and the digit 1 shall also be displayed so as not to be confused with each other.</li> <li>2. Error checking shall be provided so that letters are not permitted in fields specifically designated for numbers and vice versa.</li> <li>3. If numbers are rounded or estimated, the rounding shall be done by the system and visually displayed for the operator to see.</li> <li>4. Numbers shall only be rounded to the level of tactical significance.</li> <li>5. Any computations using rounded or estimated numbers shall display the answer in the appropriate number of significant digits.</li> <li>6. Numbers in columns shall be justified on the virtual or actual decimal.</li> <li>7. Graphic presentation aids interpretation. When graphic presentation of numbers positively impacts operational effectiveness, then numbers shall be presented graphically.</li> <li>8. For small numbers the graphic shall be a one-to-one representation of the numbers. For example, since the Halifax-Class CPF carries a maximum of eight Sea Sparrow missiles, the number of missiles remaining and fired can be presented as individual symbols (such as</li> </ol>		

<p>icons or geometric shapes) depicting each missile and missing missile. The number should also be presented in numerals so that the operators do not have to count.</p> <p>9. For large numbers, the graphic shall be a summary of the relevant information. A progress bar showing the time remaining is an excellent example. The progress bar gives at-a-glance information as to how far along the process is (in the example of ammunition, it would give a rough idea of how much ammunition is used and remaining). The graphics shall be supported with specific numbers so that the operator can report the precise values as required.</p> <p>10. Values of zero shall not be left blank but shall instead be presented as zeros.</p> <p>11. Missing or unknown values shall not be left blank but instead shall be displayed with place markers such as periods or dashes.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion provided given the nature of the guidelines.</li> </ul>

5.5.3.14	Units of measurement	<b>Source:</b> COMDAT (12.3.8); HFES 200
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. If the field is associated with a standard and consistent dimension (e.g. feet) then this dimension shall be provided by the system and listed to the right of the field.</li> <li>2. If a finite number of dimension are associated with the field then the possible dimensions as an option menu shall be provided and presented to the right of the field.</li> <li>3. If a multitude of dimensions are possible, use another text field (or a menu) for the input of the dimension.</li> <li>4. Any text that requires a unit of measure shall be immediately followed by either the proper unit of measure or a control to select a unit of measure.</li> <li>5. The system shall allow users to input data in a familiar unit of measure.</li> <li>6. The units of displayed data shall be consistently included in the displayed labels.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Displaying units as an additional label when it is required assists the user with interpreting the data in an entry field.</li> </ul>		

5.5.3.15	Apply standardized capitalization rules	<b>Source:</b> WMBP COMDAT (12.3.9)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Titles and major headings shall be presented in book title case.</li> <li>2. Use of all uppercase letters shall be reserved for acronyms and security classification banners.</li> <li>3. Arabic rather than Roman numerals shall be used when information has to be numbered.</li> <li>4. When numeric data is displayed or required for control input, such data shall be in the decimal, rather than binary, octal, hexadecimal or other number system.</li> <li>5. All sentences and lines of text shall be presented in a combination of upper and lower-case letters, following standard capitalization rules. Mixed case type is recommended as it results in faster word recognition, especially in sentences.</li> <li>6. Any displayed message or datum selected as an option or input to the system shall be highlighted to indicate acknowledgment by the system.</li> <li>7. Use title style capitalization for the following: button label, header, menu label, menu command label, option button label, pop-up menu command, split button label, soft key label, tab label, text header, toolbar buttons, screen tip, tree view label, window, dialog box, and error message titles.</li> <li>8. Use sentence style capitalization for the following: check box label, combo box label, drop-down list items, error message text, instructional text, list box label, option button label, progress bar label, slider label, text box label, drop-down list label, text sub header, and time picker label.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Judicious use of sentence style and title style capitalization facilitates readability of the text on the screen.</li> <li>• These guidelines apply to all interface types including hand-held devices.</li> </ul>		

5.5.3.16	Acronyms and abbreviations	<b>Source:</b> COMDAT (12.3.10)
----------	----------------------------	------------------------------------

**Guideline(s):**

1. An acronym in an application shall be used to denote only one meaning. Acronyms shall not be created when existing acronyms are already defined.
2. Acronyms and abbreviations shall be used only if shorter than the full name and only if understood by users.
3. Abbreviations shall be the shortest possible length that will ensure uniqueness.
4. Abbreviations shall be meaningful, recognizable, and used consistently.
5. Words not commonly abbreviated by the operators shall not be abbreviated.
6. Acronyms and abbreviations shall comply with MIL-STD-12, MIL-STD-411, and MIL-STD-783.
7. New acronyms shall be generated following rules contained in MIL-STD-12.
8. A dictionary shall be available (e.g., in Help) for decoding abbreviations and acronyms.

**Discussion:**

- Acronyms are a common within the standard military lexicon.

5.5.3.17	Tables	<b>Source:</b> COMDAT (12.3.11)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Tabular data windows shall be used for information display only.</li><li>2. Where items in a list are displayed in multiple columns the items shall be ordered vertically within each column.</li><li>3. Each column shall have a heading and be clearly separated from other columns by spaces (minimum of 4 spaces) and/or lines. (Note: MS1472F requires no less than three spaces.)</li><li>4. Tables shall provide sorting on all sortable columns through the column heading.</li><li>5. When selected, the sorting control on the heading shall indicate the column that was sorted and the direction in which is sorted.</li><li>6. If multiple sorts are required use check boxes in a table, a non-editable field shall be</li></ol>		

provided to display the order of the sort criteria. Column labels shall be placed in the non-editable field in the order in which the column label check boxes or push buttons were chosen. When a column label check box or push button is de-selected, the label shall be removed from the sort-order field.

7. Where items in a list are displayed in multiple columns then selecting an individual item in one column shall also select the data in the corresponding line of all the columns.
8. If there are multiple pages in table, the row and column labels shall remain visible along the edges of the page.
9. The widths of columns containing the same data elements shall be uniform and consistent within a table and from one table to another.
10. Tabular displays shall not extend over more than one page horizontally.
11. Tables shall be scrollable by means of one scroll bar that is located on the right side of the right-most column.
12. The width of each column shall at a minimum be the same width of the column label.
13. If the width of the longest data element in a list is unknown or is variable, horizontal scroll bars shall be used for individual columned lists. However, horizontal scroll bars shall only be displayed when needed.
14. Column labels shall be identical for all pages; the last line on a page is the first line on the next page.
15. Tables shall be arranged to show similarities, differences, trends, or relationships. Depending on the task, data may be arranged in sequential, spatial, alphabetical, functional and/or chronological order.
16. Data that are important, require immediate response, and/or are frequently used, shall appear at the top of the table.
17. Rows and columns shall be labelled distinctively.
18. Tabular data shall be displayed in rows and columns. If the data in the rows has order, the order shall increase from left to right. If the data in the columns has order, the order shall increase from top to bottom.
19. Tabular data displays shall be used to present row-column data to aid detailed comparison of ordered sets of data.
20. Tables or other displays that contain similar data shall be easily visually distinguished by the operators.

**Discussion:**

- Strict adherence to these guidelines for handheld devices may not be possible. This requires further investigation to determine the feasibility of applying these guidelines to handheld devices.

5.5.3.18	Multiple Pages	<b>Source:</b> COMDAT (12.3.12)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. If possible, related data fields shall appear on the same page.</li> <li>2. When a display contains too much data for presentation in a single frame, the data shall be partitioned into separately displayable pages.</li> <li>3. If there are multiple pages, each page shall have the same title, current page, and total pages indicator.</li> <li>4. When partitioning displays into multiple pages, functionally related data items shall be displayed together on one page.</li> <li>5. The numbering of items shall be continuous from one page to the next. For example, if the last numbered item on page 1 is 8 then the first numbered item on page 2 is 9.</li> <li>6. If a data window has more than one page, the window includes paging controls.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Partitioning of data across multiple pages should not adversely affect the ability of the user to perform the required tasks.</li> </ul>		

5.5.3.19	Support Data Query	<b>Source:</b> COMDAT (12.3.13)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The data query language provided to the users shall reflect the structure of the data as perceived by the users.</li> <li>2. The language shall allow the users to specify the data to retrieve, then display and manipulate it.</li> <li>3. Users shall be provided the capability to request data without having to tell the system how</li> </ol>		

<p>to find it.</p> <ol style="list-style-type: none"> <li>4. Queries shall use operationally meaningful terminology and do not reflect how the data are stored.</li> <li>5. Users may construct simple and complex queries, create predefined queries and save, retrieve, and execute these queries.</li> <li>6. The language shall permit alternate forms of the same query using natural language.</li> <li>7. Users shall be prompted to confirm a query if data retrieval time will be excessive</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• While queries may invoke a series of scripts on the part of the system, the user should only be exposed to a query capability that is intuitive and easy to use.</li> </ul>

#### 5.5.4 Data Validation and Error Checking

Syntactic error checking is done to be sure that data is entered in the correct format. This includes that the correct numeric data falls within minimum and maximum range requirements, and alphanumeric data is recognizable by the system. Alphanumeric data may include abbreviations and acronyms that must be in the proper format.

Semantic error checking is done to ensure that the meaning or context of information entered is correct. This means that one data entry field must be compared against another field. For example, an error should result if a crew has been assigned to a non-existent aircraft tail number. Detecting this type of error requires semantic error checking.

5.5.4.1	Provide error checking of user inputs	<b>Source:</b> COMDAT (12.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A validity check shall be done on data entered, with visual and/or error messages.</li> <li>2. Syntactic error checking shall occur immediately upon leaving a data entry field.</li> <li>3. Users shall be notified of syntactic errors in a standard message area or error message window.</li> <li>4. Semantic error checking shall occur as soon as is possible. If semantic error checking occurs upon leaving a field, users shall be informed of the error and be provided the capability to revert the collection of data entry fields (form, database record, etc.) to its previous state. If semantic error checking occurs when an entire form or record is</li> </ol>		

submitted as a collection, the user shall be provided with the capability to correct the individual fields in error before again submitting the collection of data entry fields. Due to the increased potential for user error with field-by-field data entry, it is recommended that data entry and semantic error checking be accomplished on a record-by-record basis.

5. Data entries shall be validated by the system for correct format, legal value, or range of values. Where repetitive entry of data sets is required, data validation for each set shall be completed before another transaction can begin.
6. Users shall not be prevented from leaving a field with an error with it. Rather, they shall be allowed to fix the error at their own pace.
7. When required data entries have not been entered, the omission shall be indicated to the user and either immediate or delayed input of the missing items shall be allowed. Delayed entry shall be avoided; however, if it is necessary, the user shall be required to designate the field to indicate that the missing item is delayed, not overlooked.

**Discussion:**

- Error checking on the part of the system helps to avoid the entry of erroneous information on the part of the user. Preference should be provided to prevent the user from the possibility of entering erroneous information by means such as restricting entry to an acceptable range of values.

## 5.6 Map and Tactical Display Characteristics

The use of maps for accessing and displaying information is currently being investigated for JCDS 21 applications and tools. Electronic maps allow the organization, presentation, analysis and communication of spatially referenced information on a wide variety of topics. This section provides guidelines related to the design of the UI characteristics for map and tactical displays, the use of alerts and drill-down strategies.

Note: Discussions are currently on-going with respect to using information that was gathered during the Advanced Linked Extended Reconnaissance and Targeting Technology Demonstration (ALERT TD) project (DRDC-V) related to the development of maps and tactical displays. There are two important aspects of the ALERT TD project work that are consistent with the development of the JCDS 21 TDP Style Guide. First, the ALERT TD project is currently in progress and includes the most current data related to the development of maps and tactical displays. Second, the ALERT project employs an iterative design process that is augmented by design milestones that are validated by SMEs. For this reason, the inclusion of the ALERT data in the current document is being pursued.

This section provides guidelines related to characteristics of general electronic map and tactical displays.

## 5.6.1 Tactical Displays

5.6.1	Characteristics	<b>Source:</b> HFDS 2003 (8.3.2) COMDAT (16.2.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. A tactical display window may contain both a tactical display and a set of controls for manipulating the display.</li><li>2. A tactical display area shall include identifying information about the tactical display (e.g., map name, area, scale) in the title bar.</li><li>3. Status information (e.g., coordinates, “drawing map”,” updating map”) shall be provided.</li><li>4. The tactical display window shall be supplemented by windows or information areas that provide amplifying information about selected objects and symbols.</li><li>5. The tactical display shall be initially presented in a default range that is appropriate to the operator position to provide the operator an overall tactical picture.</li><li>6. Users shall select and deselect symbols on the tactical display using standard selection methods.</li><li>7. Labels shall be able to be applied to any object on the tactical display. All labels are global so that once an object is given a label, other operators must use this label. As a default the label shall only be displayed in the amplification window, but each user shall have the ability to also display any label on the map. In either case, the label shall always be displayed in the amplification window. Only the originator and authorized operators may alter global labels.</li><li>8. The tactical display shall be a fixed window and take on all characteristics of fixed windows.</li><li>9. The tactical display shall include a means by which users can obtain help in identifying unknown symbols or other object information.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Tactical displays take priority on the display real estate given their importance with supporting the task at hand.</li></ul>		

5.6.2	Tactical Tools	<b>Source:</b> COMDAT (16.2.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Controls that affect the tactical display map, such as range selection and overlays, shall be located in an area just below the Tactical Display.</li> <li>2. The operator shall be able to enter geographic points (such as lat/long) to identify locations for graphic elements. Examples include specifying the location of a specific object or setting the parameters for a geographic area.</li> <li>3. Tactical displays shall provide a distance/azimuth function that calculates the distance (range) and azimuth (bearing) between any two selectable points or symbols. Distance shall be presented in user-selectable units (in feet, meters, miles, nautical miles, or kilometres). Azimuth shall be displayed in degrees from true North.</li> <li>4. The software shall provide a means to determine the range and bearing between any two points on the tactical display. The first point shall be selected by the operator and the second point will be the current location of the pointer. A line shall be drawn between the first point and the pointer location. As the pointer moves, the range and bearing information shall be updated. The bearing shall be given from the first point to the second point.</li> <li>5. Tactical displays shall provide an automated means for determining the bearing and range between points. If users need to judge precise distances, computer aids may be provided.</li> <li>6. Bearing and range lines shall display the bearing and range (in that order) near the bearing and range line approximately 1/3 of the distance between the two points and closest to the second point chosen.</li> <li>7. The display of bearing and range lines shall be operator-selectable as in an overlay.</li> <li>8. The tactical display shall provide a position determination function that calculates the position of an identified point. The point is provided by latitude and longitude, distance (in feet, meters, miles, nautical miles, or kilometres), and an azimuth. Coordinates shall be provided in a user-selectable coordinate system (e.g., Universal Transverse Mercator, latitude/longitude, or Military Grid Reference System).</li> <li>9. The software shall provide a means to determine the latitude and longitude location of the pointer at any location on a tactical display. The software shall also provide the bearing and range of the pointer location from Ownship.</li> <li>10. The software shall provide a means to compute a running distance along a series of lines. This would be used to determine non-straight line distance.</li> <li>11. The software shall provide a means to determine an area by defining a circle, rectangle, or</li> </ol>		

polygon region.

12. An operator shall be able to select a contact, input a time duration and bearing (default to current bearing) and the software will provide a predicted location for that contact assuming its bearing does not change.
13. Selecting two tracks (or contacts) or one track (or contact) and a fixed point, the operator shall be able to see dead reckoned lines and the location of the closest point of approach (CPA).
14. The tactical display shall provide a direction function that allows users to designate direction by pointer placement using the pointing device, entry of bearing and elevation angles, or specification of points of interest.
15. When tactical display location data is frequently used, a constantly visible display of coordinates associated with the pointer shall be displayed in appropriate coordinate units. The continuous display of location shall be augmented with the capability to fix (point on the tactical display) a location to facilitate moving overlay displays. If appropriate to the application, the capability to fix a location on the display and label it for storage, retrieval and centering of the display may be provided.

**Discussion:**

- Additional tactical tools may be required depending on the operational context. An analysis of the tasks required to be completed by the target population should inform the necessary tools to be designed and implemented.

5.6.3	Drawing tools	<b>Source:</b> COMDAT (16.2.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. The system shall provide a set of appropriate drawing tools. The operators shall only be provided with drawing tools that have operational relevance. Whenever operationally relevant drawing tools are provided they shall use the symbols and options described in MSWUE.</li><li>2. New drawing tools and symbols shall be provided if required. Such tools shall provide the operator with easy means to draw the required tactical objects.</li><li>3. Drawing tools shall include means for the objects to re-size or re-locate according to defined specification. For example, the drawing tool that supports drawing of an expanding area of interest must permit the operator to specify the rate of expansion.</li><li>4. Detailed specification of the actions of drawn objects shall be operable from the drawing</li></ol>		

<p>tool and shall not require additional menus to be opened.</p> <ol style="list-style-type: none"> <li>5. When line or figures must be drawn to represent numeric coordinates, computer aids shall include templates for entering the coordinates, and if necessary, selecting the appropriate units for those coordinates.</li> <li>6. Where graphic data must be plotted in predefined standard formats (e.g., target areas on maps), templates or skeletal displays shall be provided for those formats to aid data entry.</li> <li>7. Map points shall include a point designation feature (e.g., cross hairs or a V-shaped symbol).</li> <li>8. The OMI shall support direct manipulation of displayed objects on the situation display. Examples include selecting drawn objects and moving or dropping them.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Additional drawing tools may be required depending on the operational context. An analysis of the tasks required to be completed by the target population should inform the necessary tools to be designed and implemented.</li> </ul>

5.6.4	Selecting Contacts or Objects – Primary Hooking	<b>Source:</b> COMDAT (16.2.4.1-2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Selection aids shall be available to assist the user in the visual location and selection of a symbol or graphic object in a display with many other closely spaced or overlapping objects. These aids shall include an interactive track group list, sequential symbol hooking without pointer movement, a user-defined contact list, location by contact name and/or number, and sequential switching.</li> <li>2. Users shall be able to hook a track by placing the pointer over the symbol and performing a select. A contact that is hooked shall be placed in the visual “foreground” such that no other contacts obscure it or partially cover it.</li> <li>3. In contact-rich environments it is not always possible to determine which of the objects is hooked if the sole visual indicator on the tactical display is a graphic (such as the rounded shape currently used) surrounding the contact. The capability to step through overlapping tracks is necessary, but not sufficient. An auxiliary visual cue (e.g., making the hooked contact appear brighter for a period after it is hooked) shall be implemented.</li> <li>4. Contacts shall be automatically pre-selected when they are closest to the pointer as defined by the Advanced Hooking Algorithm. Pre-selection provides operator specified contact information in the pre-hook window and indicates to the operator which object will be</li> </ol>		

hooked if the operator performs a select. A white, dotted circle shall be placed around contacts when they are pre-selected unless the contact is already selected.

5. Users shall be able to select a single object on a map within a densely packed group of objects. When a graphical item is selected, it shall be highlighted. If appropriate to the application, users may reposition selected elements on the display.
6. Symbols other than contacts, such as special points or Identify Friend/Foe (IFF) symbols, are highlighted in the same manner as contacts.
7. Selected lines are highlighted by changing the colour and thickness of the line and any points associated with the line. The line width and any points shall be increased by 1.5-2 times their normal size.
8. Selected areas are highlighted by placing a coloured outline around the entire area thus changing the colour of its outlines and associated points. The outline width and any points shall be increased by 1.5-2 times their normal size.
9. Primary or pre-selection will remove whatever information is currently in the respective amplification window including contact information. The same guidelines apply as with contact selection. (note: Secondary selection is defined to permit multiple users to hook objects on a single display, access is via a middle input device button. Secondary selection is not addressed in the COMDAT style guide and is not referenced here.)

**Discussion:**

- Refer to Section 5.6.2 for related guidance on panning and zooming.

5.6.4	Selecting Contacts or Objects – Amplification Window	<b>Source:</b> COMDAT (16.2.4.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Amplifying information for any hooked object is provided via the same amplification area and pop-up windows used for contact information.</li> <li>2. The contact amplification window shall be a fixed window.</li> <li>3. The system shall contain a contact amplification window that displays the object information. The operator shall be able to select what amplification information is displayed.</li> <li>4. A contact amplification pop-up window shall be able to be toggle on and off for both hooked and pre-selected contacts through their respective amplification windows. When the pre-select pop-up window is on, it shall be displayed when the object is pre-selected.</li> </ol>		

<p>The hooked pop-up window may be displayed on selection as determined by the operator.</p>		
<p>5. A data block shall contain basic information about a hooked object. Data blocks are used to provide a view of the essential information about objects on the display without requiring that the operator take eyes off of the tactical picture display.</p>		
<p>6. If data blocks are implemented they shall have the following characteristics:</p>		
<p>a. The data block information shall be presented on a transparent background so as not to obscure the tactical picture.</p>		
<p>b. The display of data blocks shall be selectable by the operator.</p>		
<p>c. The data block shall be able to be hooked on the display and moved to another portion of the tactical display.</p>		
<p>d. The data block shall include a thin line connecting the data block to the object with which it is associated if the data block has been re-located by the operator.</p>		
<p>e. The data block shall move with the associated object and shall keep its position relative to the associated object.</p>		
<p>f. Information in the data block shall be presented in a fixed order that shall be common to all data blocks of specific types.</p>		
<p>g. Display of data blocks shall be operator-selectable.</p>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• A data block is a special case of amplification.</li> </ul>		

5.6.5	Sequential Contact Hooking	<b>Source:</b> COMDAT (16.2.5)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The user shall be able to place the pointer in the location of a group of symbols and successively hook each symbol in a specified range without moving the pointer. As the user presses the sequence function, the next symbol in the group shall be selected. Contacts shall be sequenced in the same order as their layering, with top contacts being sequenced first.</li> <li>2. A symbol that is currently highlighted in the symbol list shall appear in the visual foreground on the tactical map.</li> </ol>		

<ol style="list-style-type: none"> <li>3. The operator shall be able to select a contact by entering its contact number or selecting its name from a list of labelled contacts. The contact shall be highlighted with the selection ring and the selection ring's new position shall be identified to the operator by the location cursor that shall disappear after 2 seconds.</li> <li>4. Sequential switching shall be able to be toggled off and on by the user with the default being off. If any contacts are overlapping, the contacts are sequentially rotated from top to bottom. Contacts shall be cycled at a default rate of 2 Hz and this rate shall be adjustable by the user.</li> <li>5. Upon selection of a new object, the previously selected symbol shall be released. Primary selection shall release the primary selected object.</li> <li>6. The labels on dynamic graphic displays shall remain with the top of the label up regardless of the orientation of the object.</li> <li>7. The user shall have the capability to obtain exact map coordinates of selected symbol or map features.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Sequential contact hooking allows the operator to quickly navigate through a series of symbols quickly without the requirement to select each symbol with the pointing device. A visual indication of the selected symbol will allow the operator to quickly scan the interface to locate the desired object.</li> </ul>

5.6.6	Overlays and Filters	<b>Source:</b> COMDAT (16.2.7)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. An automated means shall be provided to register graphic data with background map information at all display scales.</li> <li>2. Maps shall allow situation display overlays on related map backgrounds. Overlays that obscure data shall not permanently erase any data, and covered data is automatically redrawn if the covering overlay is removed.</li> <li>3. Map labels and other overlay data shall be positioned consistently (e.g., beneath or within the feature) and all significant features shall be labelled.</li> <li>4. An option shall be provided to suppress some or all labels on a map display.</li> <li>5. Map overlays such as country or city names, contact names, road, rivers, or borders shall be able to be toggled on or off by the operator to avoid unnecessary clutter. The operator</li> </ol>		

shall be able to set preferences to toggle these values based on the current range scale.

6. If applicable, elevation features are represented using colour shade coding of a single colour range (for example shades of blue for depth of water), rather than hue coding (i.e., red, yellow, green, etc.). Terrain features identification and classification and operational data may use hue coding. Colours do not change in any window when the focus is shifted from one window to another.
7. Map labels shall remain legible at all display resolutions.
8. Topographic and bathometric data shall be displayed using shading of the land and water colours. Darker shades shall be used for lower elevations and lighter shades for higher elevations.
9. Maps or situational displays shall provide a user-selectable grid overlay that is keyed to the coordinate system of the map.
10. The intensity of the map and overlays shall be adjustable to fade out without losing all map features, while maintaining the brightness of selected overlays.
11. If appropriate to the application, map backgrounds with user-selected overlays (situation displays) may be named for ease of storage, retrieval, modification, display and hard copy.
12. Filters shall be provided and may include Category/ID filters; geographical filters; range and vector filters; attribute filters.
13. Essential contacts that shall not be filtered out of the display include the following: hostile tracks, unknowns, friendly interceptors or fighters, and any track subject to engagement status.

**Discussion:**

- Refer to Symbology section of these guidelines for recommendations on tactical display overlays.

### 5.6.2 Drill down capability

Tactical displays and maps are used as a navigation tool in many of the JCDS 21 applications and tools. This type of interfacing is considered a type of “zoomable user interface” (ZUI); a graphical environment where users can change the scale of the viewed area in order to see more detail or less. The two main characteristics of zoomable user interfaces are that:

1. Information objects are organized in space and scale, and
2. Users interact directly with the information space, mainly through panning and zooming.

In zoomable user interfaces, space and scale are the fundamental means of organizing information (Chung. 2001; Hornbaek, et al. 2002). The appearances of information objects are based on the

scale at which they are shown. Semantic zooming is commonly used with maps, where the same area on the map might be shown with different features and amounts of detail depending on the scale. An example of a ZUI interface is the “Google Maps” or “Google Earth”. The ZUI idea allows the user to see the overall information architecture, and to view relationships between types of information in a spatial reference. Such interfaces can improve subjective satisfaction and efficiency but is prone to a loss of user context, difficulty in finding information (information overload) and a complexity of zooming interaction.

It must be noted that ZUIs have yet to fully mature, and much of the research and development has been exploratory in nature, rather than experimental. As a result, most of the current literature regarding ZUIs only address their implementation and innovation without empirical evidence to validate some of their claims (an exception to this is the Hornbaek, 2002 paper).

The guidelines below recommend techniques to enhance the ZUI approach with an overview window as a more effective navigation tool. If other techniques are to be used, additional research is required to investigate of these types of strategies.

5.6.2.1	Overview Windows	<b>Source:</b> Hornbaek
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The overview and detail windows shall be tightly coupled, so that navigation or selection of information objects in one window is immediately reflected in the other windows.</li> <li>2. For any relation between overview and detail windows, the zoom factor shall be the ratio between the larger and smaller of the magnification of the two windows. For overview + detail interfaces, this factor is recommended to be below 25 or below 30.</li> <li>3. Overviews shall use at least one-sixteenth the size of the detail window.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Providing an overview window aids users in keeping track of their current position in the information space (also known as context +detail display). Navigation is more efficient because users may navigate using the overview window rather than using the detail window.</li> <li>• Interfaces with overviews present multiple views of an information space where some views show detailed information about the information space (called detail windows), while other views show an overview of the information space (called overview windows or overviews)</li> <li>• The size of the overview window influences how much information can be seen at the overview and how easy it is to navigate on the overview. However, a large overview window might take screen real estate from the detail window. It has been argued that the most usable sizes of the overview and detail windows are task dependent. For instance, a</li> </ul>		

large overview window, for example, is required for a monitoring task, while a diagnostic task might benefit from a large detail window.

- Note that there is a trade-off when using overview. Providing overviews can increase task completion time (more information to process) but user satisfaction is also increased.

## 5.7 Charts and Graphs

Charts and graphs are information visualization techniques for displaying data characteristics, values, as well as objective relationships between data sets (e.g., correlations). These techniques include all the traditional function graphs, icons and glyph displays, pseudo-colour, contour lines and vector maps, useful either for displaying entity-related data and spatial data. To determine the type of chart or graph to employ, identify the capability of the chart and graph application and understand the type of data that needs to be generated. Based on this analysis determine the type of graphical representation that can be used to reflect the data. Ensure that the application provides options for users to select data and gives opportunity for novice users to get more info to guide their choices.

5.7.1	Display Characteristics	<b>Source:</b> HFDS 2003 (8.3.3), HFES 200, COMDAT (14.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Displays of graphs shall be used to present assessment of trend information, spatially structured data, time critical information or relatively imprecise information.</li> <li>2. When users must compare graphic data across a series of charts, the same scale shall be used for each chart.</li> <li>3. The window shall include a title and shall be sized so that the entire graph is visible</li> <li>4. Users shall not have to scroll or resize a graph display window to view its contents.</li> <li>5. Charts and axes shall be clearly labelled, and important information shall be highlighted.</li> <li>6. When a user must compare graphed data to some significant level or critical value, a reference index or baseline shall be included in the display.</li> <li>7. When precise reading of a graph may be required, the capability shall be provided to supplement the graphic representation with the actual numeric values.</li> <li>8. Where graphs are presented, only a single scale shall be shown in each axis, rather than including different scales for different curves in the graph. If interpolation must be made or where accuracy of reading graphic data is required, the user shall be provided with</li> </ol>		

computer aids.

9. Linear scales shall be used in preference to logarithmic or other non-linear scales.
10. Gradations shall be at standard intervals; intervening gradations shall be consistent with the labelled scale.
11. Labels shall be used instead of legends or keys to identify the data.
12. The labels shall be oriented horizontally and located next to the data being referenced.
13. Gridlines shall be unobtrusive and shall not obscure the data presented in the graph.
14. Users shall be provided the capability to display or suppress gridlines as desired.
15. The same coding scheme shall be used consistently throughout an application.
16. Users shall be able to re-draw multiple graphs using the same scale to facilitate comparison.
17. For precise values, users shall be able to display the actual values on the graph and to zoom.

**Discussion:**

- No additional discussion required given the nature of the guidelines.

5.7.2	Line Graphs	<b>Source:</b> COMDAT (14.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Line graphs or curves shall be used for displaying relations between two continuous variables. This includes trend information, spatially structured information, time critical information, or relatively imprecise information.</li><li>2. The axes of the graph shall be clearly labelled and include the unit of measurement as appropriate.</li><li>3. The labels shall be in mixed-case letters and oriented left to right for normal reading.</li><li>4. The horizontal (x-axis) shall be used to plot time or the postulated cause and the vertical (y-axis) shall be used to plot a caused effect when these variables are appropriate.</li><li>5. Minimum/maximum values shall be shown on each axis, with up to nine intermediate</li></ol>		

<p>markings.</p> <ol style="list-style-type: none"> <li>6. The starting point of an axis shall be zero, with the gradations indicated in whole numbers unless zero is an inappropriate starting point and whole numbers are inappropriate.</li> <li>7. When graphed data represents only positive numbers, the graph shall be displayed with the origin at the lower left. When data includes negative values and the axis extend in both directions from a zero point, the origin shall be displayed in the centre of the graph.</li> <li>8. Each line or curve on a graph shall be labelled and coded and critical or abnormal data shall be coded.</li> <li>9. Multiple trend lines shall be presented on a single graph.</li> <li>10. Where users must evaluate the difference between two sets of data, that difference shall be plotted directly as a curve in its own right, rather than requiring users to compare visually the curves that represent the original data sets.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion required given the nature of the guidelines.</li> </ul>

5.7.3	Surface Charts	<b>Source:</b> COMDAT (14.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. When curves represent all of the portions of a whole, surface charts shall be used to display aggregated amounts.</li> <li>2. The area defined below the curves or lines in a surface chart shall be textured, shaded, or coloured.</li> <li>3. Data categories in a surface chart shall be ordered such that the least variable curves are displayed at the bottom and the most variable at the top.</li> <li>4. If space is available, labels shall be placed within the textured or shaded bands, the labels shall be easily readable within the bands.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion required given the nature of the guidelines.</li> </ul>		

5.7.4	Bar Charts and Histograms	<b>Source:</b> HFDS 2003 (8.3.3.6) COMDAT (14.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Bar charts shall be used when comparing a single measure (e.g., number of eligible recruits, thousands of dollars, etc.) across a set of several entities (e.g., geographic regions, level of education, religion, etc.) or for a variable sampled at discrete intervals. Histograms (bar charts without spaces between the bars) shall be used when there are a great many entities or intervals to be plotted. A bar chart (or histograms) is used to display discrete values of information.</li> <li>2. In a related series of bar charts, a consistent orientation of the bars (vertical or horizontal) shall be adopted.</li> <li>3. When data must be compared, bars shall be adjacent to one another. Adjacent bars shall be spaced such that a direct visual comparison can be made without eye movements</li> <li>4. A reference index shall be provided when displayed values must be compared with some critical value.</li> <li>5. Use of iconic representations of quantitative information (e.g., using a silhouette of a person to represent 1000 people) shall be avoided.</li> <li>6. When bars are presented in pairs, they shall be labelled as a unit, with a legend provided that distinguishes between the bars.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion required given the nature of the guidelines.</li> </ul>		

5.7.5	Flow Charts	<b>Source:</b> HFDS 2003 (8.3.3.11) COMDAT (14.5)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Flow charts shall be used for a schematic representation of sequences or processes.</li> <li>2. The steps in a flow chart shall be presented in a logical order (e.g., a process by sequence of activity or by decreasing importance to mission success).</li> <li>3. If there is no inherent logic, the steps shall be ordered to minimize the size of the flow</li> </ol>		

chart.

4. The path indicated in the flow chart shall be left-to-right, top-to-bottom, or clockwise.
5. Each decision point in the flow chart shall contain a single, simple decision.
6. The flow chart elements and lines shall be consistently coded to assist in understanding.
7. The flow chart shall provide direction indicators to indicate the sequence to be followed.
8. A legend shall describe each element and code; critical information and/or steps highlighted.

**Discussion:**

- No additional discussion required given the nature of the guidelines.

5.7.6.	Manipulation of Graphical Data	<b>Source:</b> COMDAT (14.6.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. When an application allows users to manipulate graphical data, the application shall allow the following:<ol style="list-style-type: none"><li>a. Users shall be able to select and edit graphical object attributes (e.g., colour, line thickness, and font size).</li><li>b. Users shall be able to enlarge or reduce graphical object sizes.</li><li>c. Users shall be able to fill enclosed graphical areas with colour and patterns.</li><li>d. Users shall be able to remove and restore selected elements of the display.</li></ol></li><li>2. If appropriate, applications shall automatically align graphical objects to an invisible rule or grid structure. Alignment shall be able to be turned on or off at the discretion of the user.</li><li>3. For most graphics data entry, pointing shall be a dual action, with the first action positioning the cursor at a desired position and the second action confirming that position to the computer. An exception may be a design that allow “free-hand” drawing of continuous lines.</li><li>4. Drawn objects shall be able to be easily repositioned, duplicated, and deleted.</li><li>5. When appropriate, users shall be able to group separate objects into a single, grouped</li></ol>		

<p>object (so that separate objects may, for example, be moved as a unit, or so that a complex object can be incrementally drawn).</p> <ol style="list-style-type: none"> <li>6. When editing graphic data, users shall be provided with the capability to change the size (scale) of any selected element on the display, rather than delete and recreate the element in a different size.</li> <li>7. A copy of the original graphics is retained until users confirm that the objects are to be changed. The objects are not modified automatically as users change them.</li> <li>8. Objects are displayed as they will be printed (or a print preview option is available to see this format).</li> <li>9. The user shall be able to change attributes, such as line, border, bullet size and shadow thickness, for improved viewing of charts, graphs and diagrams, but such changes would not affect the meaning.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion required given the nature of the guidelines.</li> </ul>

5.7.7	Creating and editing map graphics	<p><b>Source:</b> HFDS 2003 (8.3.2.3) COMDAT (14.6.2)</p>
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The system shall provide a set of appropriate drawing tools. Whenever drawing tools are provided they shall use the icons and options described in MSWUE.</li> <li>2. Drawing tools shall provide the operator with easy means to draw required tactical objects. New drawing tools and icons shall be provided if required.</li> <li>3. Objects shall emerge as they are drawn.</li> <li>4. Applications shall automatically complete figures (e.g., closure of a polygon) upon demand and shall draw lines between user-specified points.</li> <li>5. Users shall be able to draw objects such as lines, rectangles, ovals, and arcs.</li> <li>6. When line drawing is required, users shall be provided with aids for drawing straight line segments. When line segments must join or intersect, computer aids shall be provided to aid in such connection.</li> <li>7. When a user must draw figures, computer aids shall be provided for that purpose (e.g.,</li> </ol>		

templates, tracing techniques, stored forms).

8. When appropriate, users shall be able to constrain line drawing to exactly vertical or horizontal. For precise drawing, users shall be able to specify their geometric relations to other lines (e.g., parallel or perpendicular to another line).

**Discussion:**

- No additional discussion required given the nature of the guidelines.

## 5.8 Presentation of Information (Scheduling graph)

This section provides high-level guidance on the presentation of information. It is concerned with the organization of information and with attributes of graphical objects in order to improve user performance and satisfaction. The goal is to represent and arrange the objects and actions possible in a system in a way that facilitates perception and understanding.

5.8.1	Graphical Schedules	<b>Source:</b> COMDAT (13.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. A scheduled event shall be represented by an event object that shall be a line, bar or other object whose length is proportional to the amount of time necessary for a particular task or task element.</li><li>2. A graphical schedule shall be presented in a horizontal format and shall present the schedule tasks or resources on the left side or y-axis of a graph. Time shall be presented along the x-axis. Each schedule event shall be displayed by an object to the right of its associated task.</li><li>3. If appropriate to the application, different types of events or events undertaken shall be differentiated by colour coding, shading, or short alphanumeric designations on or above the event icons.</li><li>4. If a coding scheme is applied to a schedule, users shall have access to a key that describes the graphical coding technique utilized. It is recommended that no more than nine differently coded event icons be presented on a schedule at one time.</li><li>5. If frequent schedule changes are anticipated, users shall have the capability to reschedule an event object by directly manipulating the event object by means of the pointing device (e.g., dragging and dropping).</li><li>6. An alternative means (other than dragging and dropping) to locate an event object on a schedule shall also be supplied. For example, one alternative is to support direct text entry of an event object start and stop times.</li></ol>		

7. Users shall have the capability to select a schedule start and stop time to be displayed in a graphical schedule window. This duration time can be a superset of what can be displayed in the window at one time. Users shall also have the capability to display all or a subset of the pre-selected duration time. For example, a schedule duration may display a week, however, users may choose to display one or more days out of the pre-selected week.
8. If graphical schedules are overly cluttered or require a high level of precision, grid lines shall be used to correspond with individual tasks or resources and times.
9. If required by the application, users shall have the capability to display or suppress a gridline that indicates the present date and time on the schedule.
10. Users shall have the capability to select an individual event object and obtain additional information about that event.
11. If more than one event icon is used per task or resource, then a label shall be supplied for each event icon that is a part of the task or resource being scheduled. For example, a schedule may display planned and actual times or earliest, latest, and actual times.
12. If appropriate for an application, symbols shall be combined with event objects to display different scheduling attributes.
13. Where relationships among items cannot be captured conveniently with a regular tree structure, items shall be linked to an arbitrary number of other items in a network.
14. Network users shall be provided the shortest or least costly path(s) connecting two items or traversing the entire network.
15. Interface representation shall include node-and-link diagrams and square matrices of items with the value of a link attribute in the row and column representing a link.

**Discussion:**

- A graphical schedule displays timelines or scheduled events.

## 5.9 Alarms and Alerts

The use of alarms is a complex affair because alarms can have many different functions over and above that of alerting the operator to a new event. Alarms are potentially a rich source of information that can be used in all phases of responding to a change in system status and should therefore be considered as an aspect of the general process information. Alarms enable the operators to maintain control of the process and to be proactive as well as reactive. In order to be effective in the detection phase of human alarm handling they need to be alerting and as clear as possible in order that they are detected quickly and that they are not confused with other signals (Brown, 2002).

### 5.9.1 Alarms as information systems

Although the primary function of an alarm system is to attract and direct the attention of operators to system states that require some kind of response, the presentation of alarms—via the annunciators—constitutes a source of information that may be used in several other ways.

Having received an alarm signal and having recognised (detected) it as an alarm signal, the operator then needs to process this information through working memory in order to diagnose the alarm condition. Diagnosis is considered to be the most difficult part of the detection / diagnosis / correction sequence, with even simple faults being hard for operators to diagnose if the fault has not been seen before. By ensuring that alarm information is efficiently and effectively presented operators are more likely to be able to process the information accurately and quickly (Brown, 2002).

The following guidelines recommend general criteria for alarm implementation.

5.9.1.1	Characteristics	<b>Source:</b> HFDS 2003 (ch7)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. When to use. If equipment is not regularly monitored, an audio alarm shall be provided to indicate malfunctions or conditions that would cause personnel injury or equipment damage.</li><li>2. Alerting and warning systems shall be unambiguous, with a clear indication of the cause for the alert.</li><li>3. Alarms systems shall:<ol style="list-style-type: none"><li>a. alert the user to the fact that a problem exists,</li><li>b. inform the user of the priority and nature of the problem,</li><li>c. guide the user's initial responses, and</li><li>d. confirm in a timely manner whether the user's response corrected the problem.</li></ol></li><li>4. Alarms/alerts should indicate the degree of malfunction or emergency.</li><li>5. When a parameter value represents a fault in some modes and not in others, it should only be alarmed in the appropriate modes.</li><li>6. When part of a redundant system, unit of equipment, module, or component becomes inoperable, an alarm signaling the loss of redundancy shall be provided to the user immediately.</li><li>7. When alarm signals are based on user defined logic, the system should allow the users to</li></ol>		

access current alarm settings that are specified in terms of dimensions (variables) covered and which values (categories) are established as critical.

8. An alerting and warning system or signal shall provide the user with a greater probability of detecting the triggering condition than his or her normal observation would provide in the absence of the alerting or warning system or signal.
9. When necessary, users shall be able to request help and related information for the operation and processing of critical and non-critical alarms, messages, and signals.
10. Auditory as well as visual alarms shall be provided when the users work in an area with a high degree of ambient illumination.
11. All nonverbal audio signals shall be accompanied by a visual signal that defines the condition.
12. When used in conjunction with a visual display, an audio signal shall be supplementary or supportive, alerting and directing the user's attention to the appropriate visual display.

**Discussion:**

- Informative feedback should be presented to the operator in case of system failure or error; such as the likely cause and/or location of the failure. It is important for operators to understand why, and under what conditions, the system might make errors.

5.9.1.2	Implementation	<b>Source:</b> HFDS 2003 (ch7)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Alarms should be automatically organized and presented to the users in prioritized form, with the most significant alarms receiving the highest priority.</li> <li>2. The display of alarms with higher current operational significance should automatically override the display of alarms with lower current operational significance.</li> <li>3. When two or more incidents or malfunctions occur simultaneously, the one generating a message of higher priority shall be presented first. After presentation of the highest priority message, remaining messages shall be presented in descending order of priority.</li> <li>4. The number of priority levels for alarm messages should be limited to four.</li> <li>5. A message priority system shall be established so that a more critical message shall override the presentation of any message having a lower priority.</li> </ol>		

6. Warning signals shall be used to indicate the existence of a hazardous condition requiring immediate action to prevent loss of life, equipment damage, or a service interruption.
7. Caution signals shall be used to indicate conditions requiring awareness but not necessarily immediate action.
8. Caution signals shall be readily distinguishable from warning signals.
9. Alarm signals and messages shall be distinctive and consistent for each class of event.
10. Example. A signal alerting a user to an incoming message would be different from a signal alerting a user to a hazardous condition.
11. Processed alarm information should be simple enough that users can easily evaluate the meaning or validity of the resulting alarm messages.
12. System status indication generally should be presented on a separate display from the alarm indicators.
13. Filtering should only be used for alarms that have no current operational significance.
14. When a single alarmed event invariably leads to subsequent alarmed events, the primary alarmed event should be shown with the subsequent events suppressed, as long as it does not interfere with the user's tasks.
15. Users shall be able to sort alarm lists based on priority and time to aid in diagnosis of the event and in determining critical actions. Prioritise the alarms in the system to allow the operator to find the critical alarms in the event that numerous alarms are triggered during abnormal operations
16. When an alarm is suppressed, users should be able to access the alarm information that is not shown.
17. The method for accessing information on suppressed alarms should not be excessively complex.
18. Training techniques should be devised to ensure that users are exposed to all forms of alerts and possible combinations of alerts and that they understand how to deal with them.

**Discussion:**

- Prioritization of alarms can be based on the immediacy of required action and impact on overall safety.
- Caution - A signal that indicates the existence of a condition requiring attention but not immediate action.
- Warning - A signal that indicates the existence of a hazardous condition requiring immediate action to prevent loss of life, equipment damage, or a service interruption.

- Advisory - A signal that indicates a safe or normal configuration, condition of performance, or operation of essential equipment or attracts attention and imparts information for routine action purposes.
- Status indication is not intended to alert the user to the need for action. When status indication is presented together with alarm information, it can increase the demands on the user.
- Alarm filtering is a technique by which unnecessary alarms are eliminated. This differs from alarm suppression in which alarm messages are not displayed but are available to the user upon request.
- Complex processing can impact the user's ability to understand the constraints and limitations of alarm processing and the validity of resulting alarms. Users rely on the system's information. Thus, it is essential that the users understand the validity of the data, how they are processed, and the limitations of the system.

### 5.9.2 System Feedback

The system should continuously inform the operator about what it is doing and how it is interpreting the operator's input. For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest while for infrequent and major actions, the response should be more substantial.

The following sections describes how best to present feedback to the operator.

5.9.2.1	Immediate feedback	<b>Source:</b> COMDAT (7.5.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Control feedback responses to correct user input shall consist of changes in state or value of those display elements which are being controlled and shall be presented in an expected and logically natural form</li> <li>2. When users take an action, there shall be an immediate and visible response to the action.</li> <li>3. A visible response shall occur even if the result cannot be displayed immediately.</li> <li>4. An application shall provide visual cues that indicate when it can accept input, when it is temporarily unavailable, and when it is unavailable during extended processing.</li> <li>5. The appearance of an object shall indicate its availability.</li> <li>6. If an operation requires several actions, users shall be prompted with the actions to take.</li> <li>7. Applications shall ignore user actions made during periods when input cannot be accepted.</li> </ol>		

<p>8. The pointing device and/or the keyboard shall be disabled when input may be destructive.</p> <p>9. Although an application shall not allow users to override disabling, users shall be able to stop a process if desired (e.g., by selecting a cancel, or equivalent, push button).</p> <p>10. The current value of any parameter or variable with which the user is interacting shall be displayed.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>Feedback will enhance operator trust in the system. The transparency of system functioning (i.e., the properties of the system which allow the operator to understand its actions) will increase the predictability of the system (e.g. reliability of automatically detecting contacts) by ensuring the operator is cognisant of the limitations of the system. In addition, it is very important that operators understand why, and under what conditions, the system might make errors. Trust will grow if operators receive informative feedback in the event of a system error (e.g. explanation of system error).</li> </ul>

5.9.2.2	System response time	<b>Source:</b> COMDAT (7.5.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>System response times shall be consistent with operational requirements.</li> <li>Required user response times shall be compatible with required system response time.</li> <li>Required user response times shall be within the limits imposed by the total user tasking expected in the operational environment.</li> <li>The system shall give warning information when a command is invoked when that command will be time consuming or expensive to process.</li> <li>System response shall be within .2 seconds of user action; display shall take no more than .5-10 seconds.</li> <li>Requests for new displays may take between 2-10 seconds if an operation requires extensive processing.</li> <li>Error feedback shall be provided to users within 2 seconds of the time error was detected.</li> <li>When a user request takes more than 2 seconds to process, the pointer shape shall change to an hourglass.</li> <li>When response to a user request takes 5-15 seconds to process, an animated cursor shall be</li> </ol>		

presented to show progress.

10. When response to a user request will take 15 seconds to 1 minute to process, a message window shall be displayed immediately informing the user that processing is ongoing.
11. As a matter of operational necessity, all processing must be able to be interrupted by the users.
12. For delays exceeding 60 seconds, a countdown display shall show delay time remaining.
13. Where system overload or other system conditions will result in a processing delay, the system shall acknowledge the data entry and provide an indication of the delay to the user. If possible, the system shall advise the user of the time remaining for the process or of the fraction of the process completed.
14. During start-up, the system shall display a message window indicating its unavailability, change the pointer shape to an hourglass or watch, and disable input from the pointing device and keyboard.
15. When the system is ready, the message window shall disappear, the pointer shall return to a standard shape, and the input shall be enabled.
16. If appropriate, the system shall display status messages, e.g., response time, and unavailability.
17. If computer processing time requires delay of concurrent user inputs, and no keyboard buffer is available, keyboard lockout shall occur until the computer can accept the next transaction. An alert shall be displayed to indicate to the user that lockout has occurred.
18. When the computer is ready to continue after a response-time induced keyboard lockout, a signal to so indicate shall be presented (e.g., the cursor changes back to normal shape).
19. When keyboard lockout has occurred, the user shall be provided with a capability to terminate a transaction that has resulted in an extended lockout. Such capability shall act like an undo command that stops ongoing processing and does not reset the computer thereby losing prior processing.

**Discussion:**

- The system should always keep users informed about what is going on by using appropriate feedback. For instance, a progress bar and/or time can indicate the expected length for executing the actions as well as the ability to cancel the operation. In addition, the dialog box can provide a simple animation to let the user know that the action is being carried out.

## 5.10 Information Coding

Using codes can help designers reduce the amount of clutter (lack of order, poor spacing, and/or unnecessary displayed information) on a visual display by representing information in “short form” (or abbreviated form) using text and/or graphics. Employing codes can also benefit user performance in information-entry applications by increasing the speed of information entry and reducing the frequency of errors, whereas poorly coded information can cause a user’s dialogue with the system to be slower and error-prone. Information coding includes colour but also things like shape, size, orientation, special markers, and textual techniques (HFES 200).

Ensure that rules of coding are matched to end-users’ expectations and tasks. Involving users in the development process can increase user acceptance and understanding.

For the purpose of ease of use and keeping up to date with the latest guidelines, the following guidelines for coding information are directed to the HFES 200, Part 5, Chapter 6.

### 5.10.1 General Guidelines

These recommendations provide guidance for construction of codes. Codes are typically related to the type of intended user, the user's task, and/or the application. The types of codes depend upon a number of factors, one of which is the skill level of the intended user.

5.10.1.1	General Coding Principles	<b>Source:</b> HFES 200 (Part 5-6.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Codes shall be used that are perceptually distinct from each others</li><li>2. Codes shall be used consistently with the same meaning or the same function Note: If different applications are employed by the same user, then it is beneficial to task performance for codes to be used consistently with the same meaning or same function across applications.</li><li>3. Meaningfulness shall be built into codes however and whenever possible.</li><li>4. When the meaning of a code is not obvious to the user, then information about the meaning of the code shall be easily accessible (9241-12, 7.1.4).</li><li>5. Codes shall be assigned according to established standards or conventional meanings for an intended user group (e.g., postal code) For example, the maximum value of a horizontally oriented slider is at the rightmost position.</li><li>6. Rules of code construction shall be established for the specification of codes. They should be applied consistently and unambiguously</li><li>7. If the absence of information is important to the user’s task, then a code should be used to</li></ol>		

indicate the absence of this information, rather than removing a code For example, if a network connection is no longer available, then the icon representing the network connection is shown crossed out, rather than removed from the screen.

**Discussion:**

- Meaningfulness is increased when clear associations exist between coded information and its intended meaning. Preference should be given to mnemonic codes over arbitrary codes because mnemonic codes are meaningful Note: Task performance is more accurate and rapid if codes are meaningful.

**5.10.2 Alphanumeric Coding**

5.10.2.1	Apply alphanumeric coding	<b>Source:</b> HFES 200 (Part 5-6.2)
----------	---------------------------	--------------------------------------

**Guideline(s):**

1. Codes shall be short. Preferably, use six or fewer characters, while still being consistent with providing meaningfulness, unique codes, and the ability to add additional codes.
2. Alphabetic codes generally shall be used, rather than numeric codes, unless it can be shown that numeric codes offer greater meaningfulness to the intended users for a particular task For example, <http://www.hfes.org/> is used, rather than <http://65.164.1.100/>.
3. If alphabetic coding is used for input, then uppercase and lowercase letters shall have the same meaning, unless this is contrary to user expectations.
4. The length of abbreviations shall be as short as possible
5. If in a set of abbreviations of equal length some abbreviations can be shortened without ambiguity, then this should be permitted to minimize required keystrokes.
6. Truncation to construct codes shall be used, when this can be done without ambiguity Example: An example for truncation might be: always take the first three letters for commands (e.g., abbreviation = abb).
7. If an abbreviation must deviate from the rule of code construction (e.g., identical words, misleading), then the extent of the deviation shall be minimized. If more than 10% of all abbreviations are deviations, then the rule of code construction should be changed
8. Conventional and task-related abbreviations shall be used when they are required to meet user expectations (i.e., this is not affected by rule of code construction).

**Discussion:**

- There are inevitable trade-offs among these factors. For example, using the fewest number

of characters conflicts with the goal of supporting the ability to add additional codes.

- The length of abbreviations will depend on the number and similarity of words to be abbreviated.

### 5.10.3 Graphical Coding

Graphical coding provides rules for the design of symbols and considerations for improving the effectiveness of graphical coding. Some coding includes the construction of icons, coding with geometric shapes and using techniques for three-dimensional perception.

5.10.3.1	Apply graphical coding	<b>Source:</b> HFES 200 (Part 5-6.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. The number of perceptible levels or discernible degrees of coding shall be limited. For example, when using size coding, use a limited number of levels (generally not more than two or three).</li><li>2. Icons shall be constructed in such a way that they are easily discerned and discriminated.</li><li>3. Icons shall be easily and clearly comprehended.</li><li>4. The use of techniques to create the perception of three dimensions shall be considered to help users discriminate among different categories of information. For example, shading is used to make buttons appear “pushed in.”</li><li>5. Every category of information shall have a unique and discriminable geometric shape.</li><li>6. The overall number of different categories and geometric shapes to be displayed shall be minimized.</li><li>7. If coding by different appearances of lines is used, then variations in line type (e.g., solid, dashed, dotted) and line width (boldness) shall be clearly discriminable.</li><li>8. If line orientation is used for coding a direction or value, then contextual information shall be provided so that direction or values are accurately identifiable.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Coding with geometric shapes helps users discriminate among different categories of information on graphical displays.</li><li>• Approximately eight combinations of line types and line widths are discriminable</li></ul>		

## 5.10.4 Colour Coding

Section 6.4 of HFES provides recommendations for screen design with colours and considerations applying to the use of colour.

5.10.4.1	Apply colour coding	<b>Source:</b> HFES 200 (Part 5-6.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Colour shall never be the only means of coding because some people have colour perception deficits.</li><li>2. When it is not possible to make the colour code redundant, desaturated colours that result in different gray levels for people with colour deficits shall be used.</li><li>3. If colour coding is used, then the colours shall be reliably distinguishable by the user.</li><li>4. A legend showing colour coding and their associated meanings shall be displayed throughout the selection and decision making processes that rely on numerous levels of colour coding. Generally, six or more levels of colour coding necessitate a persistent legend.</li><li>5. Redundant coding techniques used to differentiate colours on a display (e.g., dot patterns) should be duplicated in the legend.</li><li>6. If it is necessary to violate the requirement for consistency in order to avoid violating the goal of restricting the number of colours used in an application, then cues should be provided to the user that the context in which the colour is being used is different and the meaning is different.</li><li>7. Familiar colour coding conventions shall be followed. Conventions that may be familiar to the user include: task conventions, such as the coding (e.g., black=0 and brown=1) used for resistors, cultural conventions, such as red for warning, or concrete identification, such as blue sky and red hot.</li><li>8. If colour codes are intended to be consistent with cultural conventions, then the following assignments shall be used, unless there are more appropriate industry-specific assignments:<ol style="list-style-type: none"><li>a. Red: Stop, Hot, Danger, Error, Extreme Warning, Alert, Emergency, Alarm</li><li>b. Yellow: Caution, Potential or Mild Warning, Requires Attention, Slow, Advise</li><li>c. Green: Go, Safe, Normal, Good, Proceed</li><li>d. Blue: Cold, Advisory</li></ol></li></ol>		

e. Gray: Inactive, Unavailable Option or Choice

9. If an application is going to be used internationally or is intended to support more than a single user population, then the colour coding shall either be culturally neutral or allow each user to easily select a palette that automatically maps the culturally appropriate colours to the relevant functions and/or meanings.
10. High contrast colours and colours that fit the conventions held by the user population should be used to signal a special and temporary state, such as a warning, or to identify emergency buttons. Note: This is one area where it is appropriate to use a highly saturated colour, because the state is temporary and the contrast with the generally less saturated interface is part of what will draw the user's attention to the warning. Example: Change the colour of the background of an entry field to yellow to indicate that it contains an error.
11. If colour is used to show values of a continuous variable, then changes in the current value should be shown by a corresponding change in one or more colour attributes. Note: If colour is used to code values of a more or less continuous variable, then it helps if there is an apparent sequence to the colour code.
12. To emphasize that items or categories of items are distinctly different, the colours used to code the items or categories should be distinctly different.
13. When the relative – rather than the absolute – values of a variable are important, gradual colour changes in hue, saturation, or brightness should be used to show the relative values of the single variable.
14. When variation in hue, saturation, or brightness is used for coding relative values, the assigned code values should be ordered so that the darkest and lightest shades correspond to the extreme values of the coded variable. Good choices include assigning short wavelengths to the low end of a continuum and long wavelengths to the high end, or using spectral order (e.g., blue, green, cyan, yellow, and red) or brightness order. Balance the brightness of spectral colours. Different hues are most appropriate for discrete rather than for continuous data; general changes in saturation or brightness are best to indicate continuous changes in data.
15. If colour is used for categorization, then all the items from the same category should be displayed in the same colour Example: A screen may have a set of meters implemented in software showing the states of various subsystems in a plant. Use one colour for all the meters that have passed critical, another for those reflecting systems that are normal, and a third for those that are approaching critical. The meters themselves will have colour, and if appropriate for the task, subsystems that are inter-related might all have the same colour (e.g., a dark gray). Independent, but interrelated, subsystems might have a different colour (e.g., a different shade of gray).
16. If categories of information are logically similar, then the same or related colours should be used when assigning colours to the categories. Example 1: The same colour might be used for the selected item in a menu and the title bar of a pop-up menu. Example 2: The same colour might be used as a background colour in a dialog box and as the background for

read-only text fields.

17. If colour is used to indicate a state, then a change in colour shall be used to indicate a change in state. An example is changing a green word reading “Normal” to a red word reading “Danger” when an alarm is triggered.

18. Indiscriminate use of colours should be avoided as it may cause displays to appear “busy” or cluttered and may reduce the effectiveness of colour coding

**Discussion:**

- Colour is a good auxiliary code and should be made redundant with some other coding technique. Coding techniques that can be used redundantly with colour hue include: line dot patterns, cross-hatching or textures, discriminable gray levels or brightness, shapes or location on the display, underlining, bold print, and other effects, and other media (e.g., auditory codes).
- Redundant coding is particularly important when colour is used to signal, or to enhance searching or classification. The goal is that when a specific colour is important for the user’s task, the colour is not the only method of coding the particular piece of information.
- Memory for items that must be recalled some time after being removed from the screen can be enhanced by providing redundant colour coding.
- Small changes in extreme reds and purples are more difficult to detect than corresponding changes in other colours, such as yellow and blue-green. Short wavelengths, therefore, are good for representing the lowest magnitude of change and long wavelengths the highest magnitude changes. When large or abrupt changes in a variable should be indicated, large hue changes are necessary.
- While the user will focus their attention on a change of colour if they see the change of colour as it is happening, they may not notice the change of colour if they are not looking at the screen. Therefore if the state change is important, make the redundant coding noticeable (e.g., use an auditory signal).

### 5.10.5 Coding with other visual techniques

The way information is presented on a display makes use of visual techniques for coding information. Some of these techniques include the use of markers and their position, luminance, and blinking. For detailed guidelines, refer to Section 6.5 of HFES.

5.10.5.1	Apply colour with other visual techniques	<b>Source:</b> HFES 200 (Part 5-6.5)
<b>Guideline(s):</b>  1. Aside from highlighting, special symbols also known as "markers" (e.g., *), should be considered for drawing attention to selected alphanumeric items.		

2. Different markers should be used to indicate single selection and multiple selection.
3. If areas in diagrams need to be distinguished, then filling the areas with different coding techniques (e.g., hatch, shading, dotting, etc.) should be considered, instead of colors.
4. Note: Consider texture coding together with color to provide redundant coding.
5. Markers should be used consistently.
6. Note: Where possible, do not use these symbols for any other purpose or display them under conditions where confusion might occur with other markers.
7. Markers should be positioned close to the items marked. However, the markers should not appear to be part of the displayed items. Markers and items should be designed and positioned in such a way that allows them to be identified clearly by users.
8. If blink coding is used, then it should be considered for applications where a displayed item implies an important task requirement for user attention.
9. If a blinking cursor is used, then only one other blink code should be used on the screen at the same time.
10. If highlighting by blinking is intended and if reading items is important, then an alternative method should be considered for highlighting the item. Example: A symbol is added to mark the item and the symbol is blinked, rather than the item.
11. Size coding (i.e., varying the size [height or width] of displayed characters or symbols) should be considered only for applications where displays have low overall density.
12. Luminance (i.e., brightness coding) should be used only for applications that require discrimination between two categories of displayed items.

**Discussion:**

- Blinking items are not easy to read and may cause fatigue, if used too much.
- Usually, at least two or three sizes can be readily distinguished for information categorization.

## 5.11 Symbology

### 5.11.1 General

The study team reviewed a selected collection of symbology sets in order to identify, based on operational experience within the C2 community, compatibilities and conflicts across these documented standards. The JCDS 21 applications were then subjected to the output of this comparison for the purpose of recommending the optimal use of symbology and colours.

A preliminary analysis was performed to review and assess relevant literature and implemented examples of symbology standards for domain areas such as military tactical objects, business intelligence, emergency response, and military operations other than war (MOOTW) as they apply to the JCDS 21 TDP tools and the C2 domain. The result of symbology analysis will determine which symbology set, or parts thereof, are most relevant to the current project and will be used to provide guidance within the JCDS 21 TDP HF Style Guide.

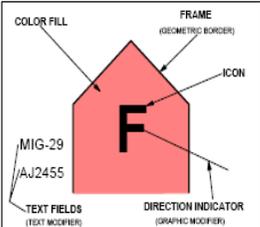
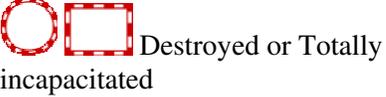
The intent of the symbology table (below) is to provide direction to designers during the development of the JCDS 21 TDP applications that will be used within short-term, domestic operations by military personnel and civilian responder communities. Given that the JCDS 21 TDP applications are still evolving the table is not intended to be used as a formal set of recommendations that must be adhered to during the early design process.

Below are the results of this preliminary analysis comparing the HLDS Symbols, Mil Standard 2525B and Command View symbology.

### 5.11.2 Symbol Composition

The variation in the composition of symbols across the different symbology sets is depicted in the following tables.

*Table 1: Symbology Comparison: Symbol Composition*

MIL-STD-2525B	HLDS	CommandView
		

Symbol composition across the standards and software applications is variable and impacts colour, framing patterns, shape, annotations outside symbol boundaries. Software applications permit user customization of symbology which is inconsistent with standard approaches to incident response.

JCDS applications are intended for domestic operations with short term durations that involve both military and civilian responders. To ensure continuity with respect to the composition of symbols, civilian symbology should be adopted, where possible, to minimize misinterpretation by civilian responders. Military (battalion level and above) and civilian officials (EMO) at strategic levels will resolve conflicts that result from overlapping use of symbol

### 5.11.3 Colour Schemes and Intent

The variation in the colour schemes and intent for the symbols across the different symbology sets is depicted in the following tables.

Table 2: Symbology Comparison: Colour Schemes

MIL-STD-2525B	HLDS	CommandView
 Pending (P)	 Destroyed or Totally incapacitated	 Operations
 Unknown (U)	 Operational, but filled to capacity or closed	 Planning
 Friend (F)	 Operational but partially damaged or partially incapacitated	 Exercise
 Neutral (N)	 Fully operational/open	 Intelligence
 Hostile (H)		 Op Centers
 Assumed Friend (A)		 OS Activity
 Suspect (S)		

Standards use colour but for different reasons:

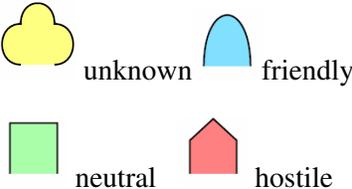
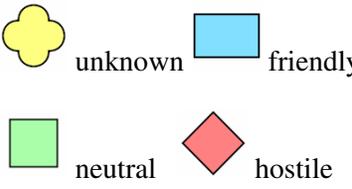
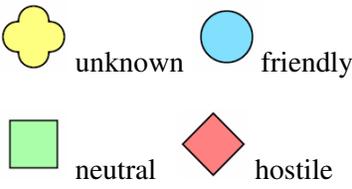
1. MIL-STD-2525B uses colour to label the unit, incident or others in terms of its status in comparison to friendly forces. Yellow symbols represent elements who's origins are UNKNOWN; BLUE/CYAN represent friendly elements or situations; GREEN represents neutral elements and finally RED symbols represent hostiles or enemy. Colours are also employed to represent the impact of meteorological events on mobility and operations (Green no impact, yellow some impact and Red major impact).
2. HLDS symbols use colour to indicate the status of the operations, elements and infrastructure. GREEN means Fully operational/open/no damage, BLUE means Operational, but filled to capacity or otherwise closed/not damaged; ORANGE Operational, but partially damaged or partially incapacitated; and RED Destroyed or Totally incapacitated
3. Command View uses colours to represent entry points to more specific information contained within a dashboard

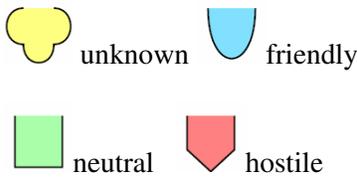
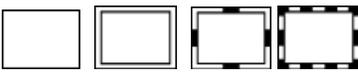
The discrepancy in colour utilization can potentially lead to confusion since MIL-STD-2525 units in red represent hostile entities whereas HLDS symbols that are red represent friendly units that have been destroyed or totally incapacitated. Command View uses colour also but it does not conflict with MIL-STD-2525B or HLDS/PSC as long as the symbols and colours continue to be used within the context of a Strategic/Executive Dash Board. When drilling down use of colour should be consistent with MIL-STD-2525B and HLSD/PSC

### 5.11.4 Shape Schemes and Intent

For each symbology set, shapes are used to differentiate between the different categories of elements as illustrated in the table below.

Table 3: Symbology Comparison: Shapes

MIL-STD-2525B	HLDS	CommandView
<p><u>Space and Air:</u> space/airborne based objects, weapons systems or tracks:</p>  <p><u>Ground:</u> ground based objects, units weapons systems &amp; facilities:</p>  <p><u>Sea Surface:</u> surface sea based objects, platforms, weapons systems &amp; ships:</p> 	<p>Shapes used to visually classify the symbols into their respective groups (Incidents, Natural Events, Operations, and Infrastructures).</p> <p><u>Incidents and natural events:</u> man made incidents either accidental or criminal in nature and natural disasters</p>  <p><u>Operations:</u> Organizations, services, capabilities or resources available during or implemented due to an emergency management situation</p>  <p><u>Infrastructure Background:</u> basic facilities, services, and installations needed for the functioning of a community or society, such as transportation and communications systems, water and power lines, and public institutions including schools, post offices, and</p>	<p>Maple Leafs are used to represent incidents, events, planning, exercises, operations, issues and intelligence</p> <p>Circles are used for operations centre activities and force generation</p> <p>Aircraft icons and ships are used for alert fighters, SAR aircraft, special events</p> <p>Squares are used to represent VIP, VOI, NATO Info</p> <p>Stars are used to represent US information</p> <p>A shield is used to represent NORAD events or NORAD operations</p>

<p><u>Sea Subsurface</u>: sea based subsurface objects, units weapons systems:</p> 	<p>prisons.</p> 	
--	--	--

For each symbology set, shapes are used to differentiate between the different categories of elements, specifically:

1. MIL-STD-2525B uses generic shapes to represent status of objects, weapons and facilities; Shapes vary across environments with exception of Neutral. Therefore requires compilation of 2 pieces of information ( environments and status)
2. HLDS uses Basic shapes are differentiated by border patterns applied to outside edges of symbols to represent categories (i.e., incidents, natural events, operations and infrastructures)
3. CommandView employs the Maple Leaf shape to represent all events and activities

Basic or generic shapes can be effectively used to denote critical information without requiring the formation of new cognitive associations (i.e., learning a new association for a Maple Leaf is not intuitive).

Manipulations of the Maple Leaf symbol require alteration of users' current and specific cognitive association with the well-known symbol. Not recommended as a symbology for use in JCDS 21 TDP due to non-intuitive application of symbology across multiple environmental contexts.

Symbols are effectively used to represent status; guidance on multiple categories can be provided if shape is distinctive

### 5.11.5 Symbol Frames

For each symbology set, shapes are used to differentiate between the different categories of elements as illustrated in the table below.

*Table 4: Symbology Comparison: Frames*

<b>MIL-STD-2525B</b>	<b>HLDS</b>	<b>CommandView</b>
Frames are consistent across symbols.	Outside edge of symbols is differentiated by border patterns representing (i.e., incidents,	No attributes are represented outside the symbol frames.

Additional attributes are represented immediately outside the symbol frame.	natural events, operations and infrastructures).  No attributes are represented outside the symbol frames.	
---	--	--

Differentiation of symbology that is based on frames may be misinterpreted by users. Solid frames or frames with minimal detail are recommended.

Additional information can be represented immediately outside the symbol frame providing it is obviously associated with the main component of the symbol.

Overall interpretation of the symbol requires compiling multiple pieces of information. The intent of the compiled symbology should be to facilitate decision making and should therefore be obvious to the users at all times, across all environments and in time critical situations.

## 5.12 Colour

Colours are used to convey specific information. The use of colour coding should be kept to a minimum. The use of too many colour codes will reduce the likelihood that significant colour coding will be interpreted quickly and accurately by users. Colour is only an effective coding scheme when an object has eight or less possible states. Colour is most advantageous when an operator must segregate or search for objects in a cluttered and unformatted display, like finding enemy contacts in a tactical plot (COMDAT)

It is assumed that the JCDS21 applications will be accessed in environments with ambient room lighting above 300 lux.

### 5.12.1 Colour Sets

5.12.1.1	Selecting colours	<b>Source:</b> COMDAT (17.1.2); HFES 200
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Colour coding for the windows and tactical display shall follow the guidance presented Appendix IV: Windows Colours and Appendix V: Tactical Display Graphics when ambient lighting is between 7-200 (20-300) lux, such as might be found in current Combat Information Centres (CICs). These colours were chosen to minimize operator eyestrain that can occur when viewing a monitor that is much brighter than the environment. (It shall be noted that such dark environments are not optimal for the viewing of colours, reading of printed documents, or face-to-face communications. Given that all of these activities commonly occur in CICs, an ambient lighting level of 100-500 lux is recommended for CICs.)</li> </ol>		

2. If ambient room lighting is above 300 lux then a light colour scheme shall be used.
3. The colours for windows and controls shall be consistent with the default Microsoft Windows settings.
4. In a dark adaptation environment then the dark colour set (See Appendix IV: Windows Colours and Appendix V: Tactical Display Graphics) shall be tailored for optimized usability under the specific lighting conditions.
5. Users shall be able to select from between colour sets: One set shall be provided for each of the lighting conditions in the operations room (e.g., normal operations room lighting and dark adaptation lighting).
6. The operator shall have the options to select between colour sets for each applicable lighting condition but shall not otherwise adjust the individual colours in the display.
7. The colour sets for the displays must be unambiguous and retain cognitively consistent colour coding conventions under all lighting conditions.
8. Changing to an alternate colour set shall not alter security classification colours.
9. In dim environments (most operations rooms, ship's bridge at night, submarines), a dark background is better for maintaining proper contrast ratios between the screen and the room objects.
10. If users must look back and forth between monitors, papers, status boards, windows, etc., the contrast ratio between any of these objects and the monitor's average luminance shall not exceed 40:1 with a preferred value of 20:1.
11. If a colour can be altered by a user, then the default set of colours should be retrievable and restorable.

**Discussion:**

- Consider providing users with colour palettes that have been demonstrated to provide appropriate levels of usability in order to increase the likelihood that users will not inadvertently hurt their own performance by inappropriate selections. For example, if the user is selecting the colour for a shape that will appear on a light coloured background, do not include the background colour as a choice on the palette.

## 5.12.2 Specific Use of Colours

5.12.2.1	Spectral Extremes	<b>Source:</b> COMDAT (17.1.3) HFES 200 (P5-7.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Colour pairs at spectral extremes (e.g., red/green, yellow/purple) shall not be used together.</li><li>2. High saturation of spectrally extreme wave lengths (e.g. red and blue) should not be used adjacently as the text or background colours for reading tasks.</li><li>3. Spectrally extreme colours that produce depth effects should not be presented for images to be continuously viewed or read.</li><li>4. For text, alphanumeric and symbols used in reading tasks that are presented in positive polarity:<ol style="list-style-type: none"><li>a. should not use spectrally extreme blue on a spectrally extreme red background;</li><li>b. should not use spectrally extreme red on a spectrally extreme blue background;</li><li>c. should avoid the adjacent display or overlapping of highly saturated spectrally extreme colours</li></ol></li><li>5. Use a minimum contrast ratio of 3:1 between images and backgrounds. With small images or thin lines that require sharply defined edges, provide a high luminance contrast with their background.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Colour pairs at spectral extremes can appear to vibrate when placed next to each other.</li><li>• Adjacent red and blue areas can produce unintended depth effects, size effects, or excessive accommodation. (HFES 7.2.3) Spectrally extreme red on a spectrally extreme blue background may create chromostereopsis. Chromostereopsis is a phenomenon of visual perception which makes it difficult to focus on an image that combines two colours as each colour is of a different wavelength of light focus.</li><li>• Two visual objects that differ in chroma (brightness and dominant wavelengths) will appear to be at different distances from the viewer.</li><li>• Spectrally extreme can be (<math>v' &lt; 0,2</math>) for blue and (<math>u' &gt; 0,4</math>) for red.</li><li>• Magenta is typically composed of a mixture of the primaries red and blue, therefore magenta alphanumerics will appear unfocussed to some users, especially to those wearing glasses.</li><li>• Visual effects of spectrally extreme colours are more obvious at low ambient illumination</li></ul>		

(e.g., 250 lux or less).

5.12.2.2	White text	<b>Source:</b> COMDAT (17.1.3)
<b>Guideline(s):</b> 1. Pure white text shall not be displayed on a pure black background.		
<b>Discussion:</b> <ul style="list-style-type: none"><li>• This combination produces an effect known as halation, which makes text less readable.</li></ul>		

5.12.2.3	Blue	<b>Source:</b> COMDAT (17.1.3) HFES 200 (P5-7.2)
<b>Guideline(s):</b> 1. Very saturated or pure shades of blue shall not be used for text or for any critical information. 2. Saturated blue should be avoided for displaying text or symbols on a dark background. 3. Spectrally extreme blue should not be used on a dark background a. Spectrally extreme blue ( $v' < 0,2$ ) on a dark background (particularly black on CRTs) does not usually maintain the required luminance contrast specified in clause 5.16, ISO 9241-3. In addition, if other colours are present (e.g., red), blue can sometimes exceed the depth-of-field of the eye. 4. Avoid saturated blue for large areas of text. 5. Saturated blue can be used to de-emphasize areas and as a background.		
<b>Discussion:</b> <ul style="list-style-type: none"><li>• The human eye has difficulty focusing on blue.</li><li>• Small saturated blue elements are often difficult to reliably discriminate and bring into clear focus.</li></ul>		

5.12.2.4	Adhere to proper use of red	<b>Source:</b> COMDAT (17.1.3) HFES 200 (P5-7.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Red shall be used to indicate critical or irreversible options.</li> <li>2. Red shall be used to alert the operator that application/process is inoperative until corrective action is taken.</li> <li>3. Spectrally extreme red on a dark background should be avoided and should not be used on a spectrally extreme blue background.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Like spectrally extreme blue, using spectrally extreme red (<math>v' &gt; 0,4</math>) makes it difficult to maintain adequate luminance contrast</li> <li>• Using spectrally extreme red on a spectrally extreme blue (<math>v' &lt; 0,2</math>) background may create chromostereopsis.</li> </ul>		

5.12.2.5	Yellow	<b>Source:</b> COMDAT (17.1.3); HFES 200
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Yellow codes shall be used only to alert to situations involving caution, recheck, or unexpected delay.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Use of yellow for this type of coding is consistent with North American cultural conventions.</li> </ul>		

5.12.2.6	Background and foreground	<b>Source:</b> COMDAT (17.1.3) HFES 200 (P5-7.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Colours with high saturation, as well as bright white, should be avoided as background colours.</li> <li>2. To better discriminate and identify colours, systems and applications should use an achromatic background behind chromatic foreground images.</li> <li>3. If foreground colours are used on a neutral background (e.g. white, gray, or black) then foreground colours that are far apart in chromacity should be chosen to improve the user's ability to distinguish among them. (e.g. yellow can be confused with white).</li> <li>4. If the background is dark, then light colours should be used for the foreground. If the background is light, then dark colours should be used for the foreground.</li> <li>5. To maximize colour visibility, use a black or dark gray background.</li> <li>6. For white backgrounds, use saturated colours for thin lines and small images to maximize legibility and produce sharp edges.</li> <li>7. Black shall not be used as the background for colour-coded items.</li> <li>8. Background colours shall be an unsaturated hue (e.g., tan, off-white) and shall not be pure white.</li> <li>9. If continuous reading is required, then medium contrast and desaturated colours that are not spectral extremes should be used.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• A good background colour is a light gray. Dark colours also make good backgrounds. If dark characters are displayed, use more subtle, pastel-like colours for backgrounds. A background in the complement colour of the main image will minimize visual afterimages.</li> <li>• Achromatic foreground colours such as white, black, or dark gray preserve their intended colour appearances better than light or medium gray achromatic foreground colours on chromatic backgrounds. Medium achromatic values (such as light or medium gray) used for thin images (such as normal font alphanumeric characters and lines) typically appear like a desaturated value of the colour background. The same medium achromatic values in bold fonts and thick lines typically appear like a desaturated value of the complement of their background colour.</li> <li>• Differences in colour based on varying amounts of blue in the colour's mixture are</li> </ul>		

particularly difficult to notice.

- Maximize the contrast between objects that should be distinguished and minimize the contrast between objects that need not be distinguished.
- On displays with a 50 or 60 Hz refresh rate, high-luminance backgrounds show annoying flicker effects. To minimize flicker, ensure that refresh rates are in the range of 60 to 80 Hertz, for 1 fL luminance levels.
- Use a minimum contrast ratio of 3:1 between images and backgrounds. With small images or thin lines that require sharply defined edges, provide a high luminance contrast with their background.
- With dark backgrounds, use desaturated colours for thin lines or small images. For white backgrounds, use saturated colours for thin lines and small images to maximize legibility and produce sharp edges.
- Consider using muted background colours, such as dark gray rather than black, and off-white rather than white. However, use colours with obvious differences in hues and brightness levels.

5.12.2.7	Symbology Colour Coding	<b>Source:</b> COMDAT,
<p><b>Guideline(s):</b></p> <p>1. Symbology shall be colour coded using conventions including the following unless superseded by NTDS or other explicitly specified symbology conventions:</p> <ul style="list-style-type: none"><li>a. Hostile: Red</li><li>b. Friend: Blue</li><li>c. Unknown: Yellow</li><li>d. Neutral: Green</li></ul>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Refer to Section 5.10.4 for additional guidelines regarding colour coding for other situations.</li></ul>		

5.12.2.8	Use colours to draw attention	<b>Source:</b> HFES 200 (P5-7.3.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. If colour is used to draw attention to a specific object on a screen: <ol style="list-style-type: none"> <li>a. the object should be the brightest and most highly saturated object on the screen, contrasting greatly with the background and with other objects on the screen</li> <li>b. the colour should be unique for the target</li> <li>c. the colour should be a warm colour (e.g. red, orange) rather than a cool colour</li> </ol> </li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Brightness and contrast are the most important principles, and then the warmth of the colour.</li> </ul>		

### 5.12.3 Colour Guidelines

5.12.3.1	Colour Guidelines	<b>Source:</b> COMDAT (17.1.4); HFES 200
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Colour shall be used only when it will increase operator performance or situational awareness.</li> <li>2. Borders between different land types, water, and coastal borders shall be consistent with the conventions presented in Appendix V: Tactical Display Graphics</li> <li>3. A maximum of three land and three water colours shall be used. Exceptions may include topographic and bathometric data if appropriate shading does not negatively impact operations.</li> <li>4. The number of colours used to code the information for graphical displays shall not exceed nine since users have difficulty discriminating among more than this number of colours.</li> <li>5. The number of colours used to code the information for alphanumeric display shall not exceed seven, and only four codes shall be displayed at any one time.</li> <li>6. Colour shall be used for coding, to highlight, aid in grouping, or to clarify relationships.</li> <li>7. When colour is used to enhance action icons, the colours shall be consistent with Microsoft</li> </ol>		

Windows conventions.

8. Colour coding may be employed to differentiate between classes of information in complex, dense, or critical displays.
9. If six or more colours are used, a persistent legend should be part of the display.
10. Colour shall be used consistently within an application and between applications in a system.
11. Colours used for displaying status shall be the same throughout the application and restricted to that function.
12. Colour used in predefined message windows shall only be used in the symbol/icon and shall not be used for text.
13. If colour is used to convey meaning, it shall be used redundantly and shall not be the only coding technique.
14. If colour coding is used, each colour shall represent one category of displayed data.
15. If the operator must focus on different types of objects in the display for different tasks, the operator shall be given the ability to use variable coded symbology (e.g., only colour code contact symbols or only colour code IFF symbols).
16. Slight shade changes in colour shall not be used to show gradation or choice. Exceptions may include topographic and bathometric data if appropriate for operations.
17. The background colour behind the text shall not be used to show a change in the system status. This is because changing the background colour usually reduces the readability of the text. Instead, change in system status shall be identified by changing the colour of an object next to the text.
18. Colour code customization shall be only permitted for information that is not tactically significant.
19. Unobtrusive colours shall be used to display information used infrequently. Warm colours (those with longer wavelengths, such as red or orange) shall be used to convey action or the requirement for a response .
20. Cool colours (those with shorter wavelengths, such as blue or green) shall be used to convey status of background information.
21. Wavelengths above 650 nm shall be avoided if the operators include protanopes.

**Discussion:**

- Colour should never be the only means of coding because approximately 8% of the

general population has difficulty in distinguishing differences in colour.

- Other coding techniques recommended include: line dot patterns, cross-hatching or textures, discriminable grey levels or brightness, shapes or location on the display, as well as underlining, bold print, and other effects.
- Memory for items that must be recalled some time after being removed from the screen can be enhanced by providing redundant colour coding.
- A persistent legend ensures the operator is not forced to remember the colour coding scheme.
- Refer to Section 5.10.4 for additional guidelines regarding colour coding for other situations.

### **5.13 Security**

Much of the information available in C2 systems is sensitive and secret in nature. Therefore, access to this information must be secured. Security provisions should enable user to gain secure access to applications, move within or between applications within the same session, and log out of applications. Security measures should also permit user to identify themselves as specific users on the network or as belonging to a group of users (e.g. Group password protected). Security measures should provide the consistent access and/or multiple simultaneous users. Environmental context in which the applications are accessed should not limit the extent to which the applications may be used. For more detailed information associated with enabling security within a Windows-based environment please refer to Annex D.

## 6 References

---

- [1] Department of Defense (2007). DoD Interface Standard – Common Warfighting Symbology, MIL-STD-2525b, w/ Change 2, March 2, 2007.
- [2] Department of Defense (1996). DoD Interface Standard – Symbols for Army Systems Display, MIL-STD-1477C(MI) 30 September 1996
- [3] Department of Defense (1999). DoD Design Standard – Criteria, Human Engineering, MIL-STD-1472F, 23 August 1999
- [4] Department of Homeland Security Working Group, Symbology Reference <http://www.fgdc.gov/HSWG/index.html>, September 14, 2005.
- [5] Department of Defense. (1997). Department of Defense Intelligence Information Systems (DoDIIS). Retrieved 02 October 2008 from <http://www.fas.org/irp/program/core/dodiis.htm> .
- [6] Defence Research and Development Canada (DRDC) –Toronto. (2004). Canadian Command Decision Aiding Technology (COMDAT) Operator Machine Interface (OMI) Style Guide, Version 2.0.
- [7] Dal Sasso Freitas Carla, M., Luzzard, Paulo R. G., Cava, Ricardo A., Winckler, Marco A. A., Pimenta, Marcelo S., Nedel, Luciana P. Evaluating Usability of Information Visualization Techniques.
- [8] Harrison, B. et al, “An Experimental Evaluation of Transparent User Interface Tools and Information Content”, Proceedings of UIST 95, November 1995
- [9] Harrison, B. and Vicente, K. (1996). An Experimental Evaluation of Transparent Menu Usage, Proceedings of CHI 96, May 1996.
- [10] HFES 200 Human Factors Engineering of Software User Interfaces, Published by Human Factors and Ergonomics Society (HFES), Santa Monica, USA, Canvass Version, 2006. <http://www.hfes.org/Publications/ProductDetail.aspx?ProductId=76>
- [11] Ahlstrom, V. and Longo, K. (2003). Human Factors Design Standards (HFDS) – For Acquisition of Commercial Off-the-Shelf Subsystems, Non-Developmental Items, and Developmental Systems., The NAS Human Factors Group , Research, Development, and Human Factors Laboratory Building 28FAA WJHTC Atlantic City International Airport, NJ 08405.
- [12] Human Factors for Designers of Equipment, 24 May, 1996.
- [13] Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. Proc. ACM CHI'94 Conf. (Boston, MA, April 24-28), 152-158.

- [14] Linwood, J. and Minter, D. (2004). *Building Portals with The Java Portlet API*. Springer-Verlag New York, Inc.
- [15] Microsoft Vista User Experience Guidelines, <http://msdn2.microsoft.com/en-us/library/aa511440.aspx>, USA, 2007.
- [16] Microsoft Corporation. (2006). *Windows Mobile Design Guidelines 5.0*. Retrieved 02 October 2008 from <http://msdn2.microsoft.com/en-us/library/ms879586.aspx>.
- [17] Microsoft Developers Network (MSDN). Retrieved 02 October 2008 from <http://msdn.microsoft.com/en-us/default.aspx>
- [18] Moraga, M., Calero, C., Piattini, M., and Diaz, O. (2007). "Improving a portlet usability model", *Software Quality Journal*, Volume 15, Number 2, June 2007, pp. 155-177.
- [19] Tognazzini, B. (n.d.). *First Principles of Interaction Design*. Retrieved 21 November 2008 from <http://www.asktog.com/basics/firstPrinciples.html>
- [20] World Wide Web Consortium (W3C). (2006). *Mobile Web Best Practices 1.0 User Interface Guidelines*. Retrieved 02 October 2008 from <http://www.w3.org/TR/2006/WD-mobile-bp-20060412/>.
- [21] Visual Analytics Inc. Website (2008). Retrieved 02 October 2008 from <http://www.visualanalytics.com/products/visualinks/vlFeatures/placements/centrality.cfm>.
- [22] i2 Inc. Website (2008). Retrieved 02 October 2008 from [http://www.i2inc.com/products/analysts\\_notebook/](http://www.i2inc.com/products/analysts_notebook/).
- [23] Niwa, Y. and Hollnagel, E. (2001). *Enhancing Operator Control by Adaptive Alarm Presentation*. *International Journal of Cognitive Ergonomics*, 2001, 5(3), 367–384

This page intentionally left blank.

# Annex A Windows

---

## A.1 Windows

### A.1.1 Primary Windows

A.1.1.1	Characteristics	<b>Source:</b> COMDAT (5.2.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. A primary window shall contain a title bar, window menu, window menu button, Minimize Control, Maximize/Restore Control, and resize borders.</li><li>2. The left side of the title bar shall contain navigation tools with a window menu button as the minimum navigation tool.</li><li>3. The name of the application shall be left justified next to the window menu button icon.</li><li>4. The right side shall contain manipulation tools with minimize, maximize, and close buttons as the minimum manipulation tools.</li><li>5. A fixed window is a special case of a primary window. A fixed window may contain a title bar but this is not a requirement if space is limited and the purpose of the fixed window is clear. It shall not contain minimize, maximize, or close buttons, or resize borders.</li><li>6. When multiple modes of operation exist, a means shall be provided to remind the user of the current mode. Modal operations shall be avoided.</li><li>7. The content of the application area shall be grouped into functional sub-areas of like information. These sub-areas shall take advantage of standard window sizes if possible.</li><li>8. Each sub-area shall be separate from other sub-areas by spacing or a framing (lines, shadow, etc.) and shall be labelled unless it will be unambiguously understood by the operator.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Primary windows provide the screen area under which most applications run. (See Figure 10.1: Components for Primary Windows for an illustration of the standard components (e.g., title bar, window menu, minimize/maximize/close buttons) residing within the primary windows.).</li></ul>		

A.1.1.2	Application	<b>Source:</b> COMDAT (5.2.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The application primary window shall be the first window displayed when an application is launched.</li> <li>2. When an application primary window is minimized, all of its secondary task windows or dialog boxes shall also be minimized.</li> <li>3. The application primary window shall be the only window that can close the application.</li> <li>4. An application primary window shall contain the following elements: Window Menu Control, Title Bar, Title, Minimize Control, Maximize Control, Menu bar, Resize Borders, and as an option, Standard Menus.</li> <li>5. If the application primary window is the only primary window in an application, then the following standard menus shall be included in a menu bar: File, Edit, and Help. If the application primary window is used in conjunction with primary task windows, then an Application Menu with the same name as the application and a Help menu shall be included in the application primary menu bar.</li> <li>6. An application primary window shall follow the Microsoft Window menu style.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• An application primary window is a specific case of a primary window. The application primary window provides application control. When an application consists of an application primary window only, the application primary window provides access to the application top-level tasks. If an application primary window is used in conjunction with several primary task windows, the application primary window provides application control and management of primary task windows.</li> <li>• All of the requirements in this section have been edited to be consistent with Microsoft Windows styles.</li> </ul>		

A.1.1.3	Primary Task Window	<b>Source:</b> COMDAT (5.2.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A primary task window shall be used to display primary data and controls for a top-level task.</li> </ol>		

<ol style="list-style-type: none"> <li>2. A primary task window shall be closed or minimized independently from other primary task windows and the application primary window.</li> <li>3. When a primary task window is minimized, all of its secondary task windows or dialog boxes shall also be minimized. When a primary task window is minimized processing in the window shall continue.</li> <li>4. When a primary task window is closed, processing in it shall stop.</li> <li>5. A primary task window may be closed but the application shall not be shut down via a primary task window.</li> <li>6. A primary task window shall contain the following elements: Window Menu Control, Title Bar, Title, Minimize Control, Maximize/Restore Control, Menu bar, and Resize Borders.</li> <li>7. A primary task window shall have a standard Microsoft window menu.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• The primary task window provides primary working areas for data display and manipulation of top-level tasks.</li> </ul>

A.1.1.4	Fixed Windows	<b>Source:</b> COMDAT (5.2.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Fixed windows shall be fully visible to the operator at all times. Fixed windows are windows that are vital to the operation of the system and must be rapidly time-shared by the operator.</li> <li>2. Fixed windows shall be automatically loaded upon system initialization and cannot be closed or minimized.</li> <li>3. Fixed windows cannot be re-sized or re-located by the user.</li> <li>4. All fixed windows shall be tiled but do not have to occupy the entire display area.</li> <li>5. No window shall overlay a fixed window.</li> <li>6. The title bar and sizing frame are removed from fixed windows since they cannot be closed, moved or sized.</li> <li>7. The following shall be fixed windows:</li> </ol>		

- a. Message area (Alert Display)
- b. Main tactical display (e.g., contact location, tracks, and route planning)
- c. Tactical display function access and viewing
- d. Primary track amplification
- e. Primary function access (e.g., map controls)

**Discussion:**

- Fixed windows are special cases of primary windows and are necessary in Maritime combat systems. Fixed windows are used for vital information that must be available to the operators at all times. These windows cannot be closed and must not be obscured by other windows. An example of a fixed window is a CCS tactical display. Because these windows are special cases, they do not have the usual capabilities of application windows. For example, they shall not have controls that permit them to be sized, relocated, or closed.

### A.1.2 Secondary Windows

A.1.2.1	Characteristics	<b>Source:</b> COMDAT (5.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The display, controls, and function of each type of secondary window implemented in the application shall be consistent with the Microsoft Windows styles (see MSWUE Chapter 9). Types of secondary windows include the following.             <ol style="list-style-type: none"> <li>a. Dialog Windows - provides an exchange of information or dialog between the user and the application. Typically, a dialog window (or box) is used to obtain additional information from the user that is needed to carry out a particular command or task. An example of a Dialog Window is the commonly-used Microsoft Open Dialog window which is presented when the user chooses to open a file from within an application.</li> <li>b. Prompt Windows</li> <li>c. Message Window</li> <li>d. Error Message Windows - informs the user of a serious problem that requires intervention or correction before work can continue.</li> <li>e. Information Message Windows - presents general information to users such as the results of a command.</li> </ol> </li> </ol>		

- f. Question Message Windows - queries the user prior to performing certain actions.
  - g. Warning Message Windows - alerts the user to a condition or situation that requires the user's decision and input before proceeding. Examples include an impending action with potentially destructive, irreversible consequences.
  - h. Working Message Windows
  - i. Selection Window
2. Secondary windows shall be used for short-term interaction with data and controls that support the primary task of a primary task window or an application primary window.
  3. A secondary window shall contain a title bar. A secondary window shall contain resize borders and a maximize button if users are able to resize the window.
  4. Secondary windows shall contain either a menu bar or an action area. An action area is more common and is provided when the number of possible user actions is five or less. A menu bar is provided only if the number of possible user actions is greater than five or if the user requires access to the File or Edit menus.
  5. Message windows contain a message area and an action area. Users shall be able to move message windows but not close, minimize, or resize them. A message window may be modal or modeless depending on its purpose. The default action of all message windows shall be non-destructive.
  6. The secondary window title bar shall contain navigation tools with a window menu button as the minimum navigation tool. The navigation tools shall be located on the left end of the title bar.
  7. The name of the application shall be left justified to the end of the window menu button, followed by a colon and then the file name.
  8. The right end of the window title bar shall contain manipulation tools with a close button as the minimum manipulation tool.
  9. When a secondary window is opened, it shall appear in front of the parent window. The parent window shall stay displayed.
  10. When a secondary window is closed, its children shall be closed but its parent shall not be affected.
  11. A secondary window shall not shut down an application.
  12. A secondary window shall not be opened at application start-up.
  13. A secondary window shall contain the following: window menu control, title bar, and title.

14. The window menu for a secondary window shall only contain the following: close and move.
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Secondary windows are called from primary windows to display information to or obtain information from the user.</li> </ul>

## A.2 Window Design Guidance

A.2.1	Ensure consistency in design across windows	<b>Source:</b> COMDAT (6.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A consistent organizational scheme for key elements shall be used in all application windows.</li> <li>2. The same window design shall be employed whenever users perform the same basic task.</li> <li>3. Different or distinctive elements may appear in a window to fit the task being performed, but these elements shall be consistent across windows within an application.</li> <li>4. Groups of windows with similar functions (e.g., a group of alert windows) shall be ordered by frequency of usage, with the most frequent at the top.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• The same information should be presented in the same location on all screens and dialogue boxes and it should be formatted in the same way to facilitate recognition (e.g., identical terminology should be used in prompts, menus, and help screens).</li> </ul>		

A.2.2	Arrange information to match user actions	<b>Source:</b> COMDAT (6.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A window shall be designed so users can manipulate objects in ways that support task performance.</li> <li>2. Objects shall be arranged so users can move quickly and easily among them.</li> <li>3. Pointer movement and keystrokes needed to perform a task shall be minimal.</li> </ol>		

4. Window layout shall support natural scanning order (from left to right and top to bottom).
5. Window layout shall be logical to users and appropriate to actions executed.
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>Information objects and operations should be accessed in a sequence that matches the way operators will most effectively and productively perform tasks with minimal error.</li> </ul>

A.2.3	Arrange information by importance	<b>Source:</b> COMDAT (6.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>The most important information and controls are shall be the upper left part of the window.</li> <li>The objects in a window shall be arranged to accommodate the possibility that users may resize the window, if the window is resizable.</li> <li>Task-critical information is visually set apart from other information in a window. At least one character line shall be left blank above and below critical information; at least two character spaces shall be left blank to the left and right of critical information.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>Visually differentiating task-critical information ensures that this data is more easily perceived by the user.</li> </ul>		

A.2.4	Design windows to minimize memory load	<b>Source:</b> COMDAT (6.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>Users shall be able to perform the task called for in a window without referring to external information.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>Providing all the necessary information to complete a task in a single location reduces the requirement for the operator to remember additional information requirements.</li> </ul>		

A.2.5	Ensure proper usage of modal windows	<b>Source:</b> COMDAT (6.5)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Modal dialogs shall be used when the system must inform the user of a situation before proceeding (e.g., potential loss of data).</li> <li>2. Modal dialogs shall be used when a task cannot proceed without additional information.</li> <li>3. Dialog windows that are system modal shall appear as the top-most window.</li> <li>4. Dialog windows that are full application modal shall appear as the top-most window when any window in the application that spawned that dialog window is selected.</li> <li>5. Dialog windows that are primary application modal shall appear as the top-most window when the window that spawned the dialog window is selected.</li> <li>6. Message dialogs shall be primary or full application modal.</li> <li>7. If a modal window is used it shall contain all of the information necessary for the operator to make the decision.</li> <li>8. When a dialog box is displayed it shall reflect the current state of the system. If a dialog box is modeless, then any changes to the application shall be updated in the dialog box.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• A window is either modal or modeless. There are three levels of modality as follows: <ul style="list-style-type: none"> <li>– Primary Application Modal. Primary application modal dialogs allow the user to interact with any other window on the display except for the window that is acting as the parent for the modal dialog window.</li> <li>– Full Application Modal. Full application modal dialogs allow the user to interact with any window on the desktop except those that are a part of the same application as the modal window.</li> <li>– System Modal. In the system modal dialogs the user is prevented from interacting with any other window on the display.</li> </ul> </li> <li>• Modal dialogs shall only be used to help the user structure and keep track of a task and to prevent user errors</li> <li>• Modal windows are used when the user must address the contents of the window before further processing can continue or to ensure the user has received an important message.</li> <li>• Modal windows should only be used when it is critical that the system obtain further information or that the user acknowledges certain information before anything else can begin.</li> </ul>		

A.2.6	Widget Selection	<b>Source:</b> COMDAT (6.6)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Radio buttons shall be used when selecting an option from up to six mutually exclusive options.</li> <li>2. An option menu or combination box shall be used when more than six mutually exclusive options (no more than 10-12) are available.</li> <li>3. Option menus are used to display small lists of mutually exclusive options. Option menus may be used in the place of a list or radio buttons when only one list selection can be made and when space is limited. See MSWUE for the appearance and behaviour of option menus.</li> <li>4. A list box shall be used when selecting from a large group of options (more than 12).</li> <li>5. If the operator must select a numeric value then the following shall be used: <ol style="list-style-type: none"> <li>a. Scale or counter for continuous values within a range</li> <li>b. Standard Combination Box for a standard set of values normally selected (and space is available)</li> <li>c. Drop-Down Combination Box for a standard set of values normally selected and there is a need to conserve space</li> </ol> </li> <li>6. If the operator must select a single option from a group of options the following shall be used: <ol style="list-style-type: none"> <li>a. Radio button if there are six or fewer options and space is available</li> <li>b. Standard Combination Box if there are more than six options and space is available</li> <li>c. Drop-Down Combination Box to conserve space or to avoid clutter</li> </ol> </li> <li>7. If the operator may select multiple options from a group of options the following shall be used: <ol style="list-style-type: none"> <li>a. Check box if seven or fewer options are available</li> <li>b. List if more than seven items are available</li> </ol> </li> <li>8. If the operator must transfer or copy items from one group to another a Transfer List shall be provided.</li> </ol>		

9. Radio buttons or an option menu shall be used when the set of options is not likely to change.
10. A list box shall be used when the options might change.
11. Push buttons shall be used for frequently used actions, and when the pointing device is already moving.
12. Pop-up menus shall be used only when it is critical to the application that users be able to access functions without moving the pointing device.
13. Pop-up menus shall not be used as the only method available for accessing operations.
14. Option menus shall be used when setting values or choosing from a set of related items.
15. Vertical scroll bars shall be provided if the list is longer than the viewable list box area. Horizontal scroll bars shall be avoided if possible. A horizontal scroll bar shall only be used if most items in a list are considerably shorter than a few longer items and screen space is limited.

**Discussion:**

- Employing the appropriate widgets for the required action allows the user to understand their intended functionality and therefore improve the overall usability of the application.

A.2.7	Coding Critical Information	<b>Source:</b> COMDAT (6.7)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Capitalization shall not be the sole indication of critical information in a window.</li> <li>2. When special symbols are used to signal critical conditions, they shall only be used for that purpose.</li> <li>3. Bolding/brightening, colour coding, etc., shall be used to focus attention on critical information.</li> <li>4. Coding shall be employed to differentiate between items of information and to call the user's attention to changes in the state of the system. Coding shall be used for critical information, unusual values, changed items, items to be changed, high priority messages, special areas of the display, errors in entry, criticality of command entry, and contacts. Consistent, meaningful codes shall be used. Coding shall not reduce legibility or increase transmission time.</li> <li>5. Colour shall be used for alerts, and users shall be required to respond before an alert is</li> </ol>		

terminated.

6. When a user's attention must be directed to a portion of a graphic display showing critical or abnormal data, that feature shall be highlighted with some distinctive means of data coding.
7. When a special symbol is used to mark a word, the symbol shall be separated from the beginning of the word by one space.
8. A maximum of two levels of intensity (brightness) shall be used. Brightness shall only be used as a code if the objects coded are adjacent. Each level shall be separated from the nearest other level by not less than a 2:1 ratio.
9. Users shall be alerted to critical information in an inactive or minimized window.

**Discussion:**

- Refer to Section 5.10 for additional guidance on information coding techniques.

## A.3 Navigation and Selection

### A.3.1 Navigation

A.3.1.1	Window mode	<b>Source:</b> COMDAT (7.1.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. In modeless windows, users shall be able to interact with other windows while the modeless window is displayed on the screen.</li><li>2. Modal windows are used when the operator must address the contents of the window before further processing can continue or to ensure the user has received an important message. Modal windows shall only be used when it is critical that the system obtain further information or that the user acknowledge certain information before anything else can begin.</li><li>3. Modal windows shall not restrict interaction with windows higher in the hierarchy unless it is critical that the system obtain further information or that the user acknowledge certain information before anything else can begin.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Unless it is necessary for command completion or if it is important to prevent further interaction until a condition is satisfied, modeless rather than modal dialog boxes should be used.</li></ul>		

A.3.1.2	Maintain consistent navigation mechanisms	<b>Source:</b> W3C
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Consistent navigation mechanisms shall be provided within and across applications and platforms.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Implementing consistent navigation mechanisms across a service helps users orient themselves as well as identify navigation mechanisms more easily.</li> <li>• Users of devices that do not have pointing devices have to scroll between hyperlinks using the keypad. Intelligent grouping, perhaps optimized through adaptation according to usage patterns, can assist usability.</li> <li>• A “drill-down” method, based on major headings, can often provide an effective means of navigation; because of the linearized arrangement of content, small screen size and lack of pointing device, it is often useful to provide a means to jump entire sections of content.</li> </ul>		

A.3.1.3	Input focus policy	<b>Source:</b> COMDAT (7.1.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Only one window on the screen shall have input focus at any time.</li> <li>2. Users shall assign focus explicitly particularly if overlapping windows are implemented, and shall do so either with the pointing device or from the keyboard.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Input focus will need to be readily visible to the operator in order to avoid confusion and the potential for errors.</li> </ul>		

A.3.1.4	Assigning Focus to a Window with a Pointing Device and the Keyboard	<b>Source:</b> COMDAT (7.1.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Users assign focus by moving the pointer into a window and clicking the S button.</li> <li>2. If users click in an empty window area, the window frame shall highlight and the window shall be raised to the front.</li> <li>3. If users click in the title bar, the frame shall highlight and the window shall be raised to the front and shall receive input focus.</li> <li>4. If users click on an object within a window, the window frame shall highlight, the window shall be raised, and an object shall be selected.</li> <li>5. A window is raised by clicking anywhere in the window. The window is then moved to the top of the window hierarchy (except for any toolbars or pallets associated with the window being raised) and is given input focus. A window may also be raised by selecting it from the open window menu.</li> <li>6. &lt;Alt&gt; + &lt;Tab&gt; and &lt;Alt&gt; + &lt;Shift&gt; + &lt;Tab&gt; shall move the focus forward and backward through window families (i.e., primary windows and window icons).</li> <li>7. &lt;Alt&gt; + &lt;F6&gt; and &lt;Alt&gt; + &lt;Shift&gt; + &lt;F6&gt; move the focus forward and backward through windows in a family.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• The system should provide a keyboard alternative to standard pointing devices that enables keyboard (or keyboard equivalent) control of pointer movement and pointer button functions in parallel with the standard pointing device.</li> </ul>		

### A.3.2 Navigation within Windows

A.3.2.1	Pointing Device Navigation for Controls	<b>Source:</b> COMDAT (7.2.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Placing the pointer on a control and clicking the S button shall move the location cursor to the object and shall give the object focus.</li> <li>2. Pressing the &lt;Ctrl&gt; key and clicking the S button on an object shall select the object and keep the current object selected.</li> </ol>		

<ol style="list-style-type: none"> <li>3. Autoscrolling shall be available when the pointer is on a scrollable control such as a text block or a list.</li> <li>4. The means shall be provided to readily move the cursor to the head or the foot (end) of a file.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion provided given the nature of the guidelines.</li> </ul>

A.3.2.2	Keyboard Navigation for Controls & Graphic Objects	<b>Source:</b> COMDAT (7.2.2-3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Cursor tab controls or other provisions for establishing and moving readily from field to field shall be provided for the purpose of editing programs or tabular data.</li> <li>2. The &lt;Ctrl&gt; + &lt;Tab&gt; and &lt;Ctrl&gt; + &lt;Shift&gt; + &lt;Tab&gt; keys shall move the location cursor to the next and previous tab group, respectively.</li> <li>3. The &lt;Tab&gt; and &lt;Shift&gt; + &lt;Tab&gt; shall move to the next and previous tab group except in the multi-line text widget.</li> <li>4. The location cursor shall be on the default or first control (i.e., top left most) when it moves to a tab group.</li> <li>5. The location cursor shall skip a tab group if none of the controls can have keyboard focus.</li> <li>6. The Up, Down, Left, and Right arrow keys shall move the location cursor between controls in the tab group with focus.</li> <li>7. Moving the location cursor to a control shall not change the state of the control.</li> <li>8. The direction of cursor movement within a window shall be from upper left to lower right and shall wrap between the first and last tab groups in the window.</li> <li>9. Arrow keys shall move the location cursor one increment at a time (e.g., to the next line in text, to the next item in a list); &lt;Ctrl&gt;+Arrow keys move the location cursor one large increment (e.g., move the text cursor to the next word instead of the next character).</li> <li>10. The virtual &lt;Home&gt; and &lt;End&gt; keys shall move the cursor to the leftmost/rightmost element in a control.</li> <li>11. &lt;Ctrl&gt; + &lt;Home&gt; and &lt;Ctrl&gt; + &lt;End&gt; shall move the cursor to the beginning/end element</li> </ol>		

in a control.

12. The virtual keys <PageUp>, <PageDown>, <PageLeft> (or <Ctrl> + <PageUp>), and <PageRight> (or <Ctrl> + <PageDown>) shall scroll the elements in the control up, down, left, or right one page minus one unit of information (e.g., one line of text) at a time.
13. The title “Window” shall be in all application pull-down menu bars to allow for the selection of hidden windows. The <Alt> + <Tab> keyboard accelerator shall shuffle through all open windows in an application.
14. Keyboard focus shall remain on the element where it was before scrolling began even when the location cursor may not be in view.
15. When keyboard action alters the element with focus, scrolling shall occur so the element is in view.

**Discussion:**

- Keyboard navigation, in conjunction with cursor navigation, allows the user to select a preferred option for interacting with the application.

### A.3.3 Object Selection

A.3.3.1	Pointing Device Selection Methods	<b>Source:</b> COMDAT (7.3.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Click the S button on an object to select it. When another object is selected, previously selected objects are deselected.</li><li>2. To select multiple objects, click the S button on objects one at a time while holding down the &lt;Ctrl&gt; key. The objects are highlighted and selected.</li><li>3. Where text has been specified to become the subject of control entries (e.g., for underlining, bolding, moving, copying, or deleting), the affected segment of text shall be highlighted to indicate its boundaries.</li><li>4. To select a range of contiguous objects, the pointer shall be positioned on the first object in the range to be selected, and then the S button shall be selected to set the anchor for the range. Any other object that was selected shall be deselected. The pointer shall be dragged until it is on the last object in the range, and the S button shall be released to complete the selection.</li><li>5. To extend a range selection the user shall position the pointer on the object that shall be the last one in the selection, and hold down the &lt;Shift&gt; key while clicking the S button on the pointing device. The objects in the revised selection range (defined from the original</li></ol>		

anchor to the current pointer position in one-dimensional collections, and defined by the diagonal from the anchor to the current pointer position in two dimensional collections) shall be reselected, and any elements removed from the selection shall return to normal appearance.

6. To add or remove a non-contiguous object to or from a range selection, the pointer shall be positioned on the object, and then the <Ctrl> key shall be held down while clicking the S button on the pointing device. If previously unselected, the object shall be selected and highlighted. If previously selected, the object is deselected and shall return to a normal appearance. The other elements in the selection shall remain highlighted.
7. A bounding box shall appear when dragging the pointer over elements in a two dimensional collection.

**Discussion:**

- Multiple selections provide an efficiency mechanism whereby the user can changes the attributes of several like objects simultaneously as opposed to changing the attributes on individual objects.

A.3.3.2	Keyboard Selection Methods	<b>Source:</b> COMDAT (7.3.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Add mode shall be used to select one element or multiple objects one at a time. Normal mode shall be used to select multiple contiguous objects.</li> <li>2. The location cursor shall be a solid rectangle in normal mode and shall be a dotted rectangle in add mode.</li> <li>3. Add mode and normal mode are used to select multiple non-contiguous objects.</li> <li>4. &lt;Space&gt; (and the virtual key &lt;Select&gt;) shall select an object or multiple objects one at a time.</li> <li>5. &lt;Space&gt; (and the virtual key &lt;Select&gt;) shall set the anchor for selecting a range of contiguous objects.</li> <li>6. &lt;Shift&gt; + &lt;Space&gt; (and &lt;Shift&gt; + &lt;Select&gt;) shall extend selection from anchor to last object selected.</li> <li>7. &lt;Shift&gt; + &lt;F8&gt; shall toggle between add mode and normal mode.</li> <li>8. &lt;Ctrl&gt; + &lt;/&gt; shall select all of the objects in a collection.</li> </ol>		

<p>9. &lt;Ctrl&gt; + &lt;I&gt; shall deselect all of the objects in a collection.</p> <p>10. Cancel shall undo a selection action and return the objects to their normal appearance.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• The keyboard and the pointing device shall be used to make selections. The normal mode and the add mode are used to make keyboard selections. The normal mode is used for making simple contiguous selections from the keyboard. The add mode is used for making more complex selections which may be disjoint. In normal mode, the location cursor and the highlight are on the same element and move together when the arrow keys are used. In add mode, the location cursor moves independently of the highlight.</li> </ul>

A.3.3.3	Other Selection Methods	<b>Source:</b> COMDAT (7.3.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A default action in a window is executed by double clicking when making a selection.</li> <li>2. Pressing &lt;Enter&gt; or &lt;Ctrl&gt; + &lt;Return&gt; shall invoke the default action after making a selection in a window.</li> <li>3. The &lt;Return&gt; key shall invoke the default action in a window if the focus is on an object other than multi-line text.</li> <li>4. Extended selection shall be available in text.</li> <li>5. In a text widget a single click shall place the text cursor, a double click shall select a word, and a triple click shall select a line of text.</li> <li>6. The virtual &lt;Cancel&gt; key shall cancel the action being executed and shall return the object to its state prior to action.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• For users who may experience difficulties in operating pointing devices (such as mice, etc.), equivalent techniques should be provided to achieve the same results with a keyboard or keyboard equivalent device. For example, to open a document the user can double click on the document icon with the pointing device. Alternatively, the user can select the document icon with the tab key, select a menu item to open the document with the cursor keys, and activate it with the return key.</li> </ul>		

### A.3.4 Object Transfer

A.3.4.1	Object Transfer Overview	<b>Source:</b> COMDAT (7.4.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Individual objects or a collection of objects shall be able to be transferred in a window or to another window in the same application as well as into other applications. This shall be accomplished by two or more of the following four techniques:<ol style="list-style-type: none"><li>a. Clipboard transfer</li><li>b. Primary transfer</li><li>c. Quick transfer</li><li>d. Drag transfer (drag and drop)</li></ol></li><li>2. The transfer techniques to be implemented include at minimum clipboard transfer and drag transfer techniques.</li><li>3. For each transfer technique the following operations shall be generally available: Copy, Move, and Link.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Providing redundant object transfer techniques allows the operator to employ a technique conducive to their personal preferences.</li></ul>		

A.3.4.2	Clipboard Transfer	<b>Source:</b> COMDAT (7.4.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. When inserting characters, words or phrases (e.g., editing), items to be inserted shall be collected in a buffer area and displayed in the prescribed insert area of the screen for subsequent insertion by user command.</li><li>2. Clipboard transfer shall allow users to move and copy objects by transferring them first from their current location to a temporary clipboard and then from the clipboard to a new location. Clipboard transfer can be used, for example, to transfer text (e.g., strings of characters, blocks of text) and graphics (e.g., lines, shapes, entire figures) within or between windows or applications.</li></ol>		

3. A clipboard move operation shall consist of cut and paste operations; a clipboard copy operation shall consist of copy and paste operations.
4. Clipboard transfer operations shall be available whenever an object that can be edited has keyboard focus. Applications shall provide access to these operations as menu options, push buttons in a window, or other buttons, and users shall be able to execute the operations either with the pointer or from the keyboard.
5. Access to clipboard transfers shall be implemented in a consistent fashion throughout the application. Access to clipboard transfer shall be available only when it is appropriate to the task being performed by users. Users shall be able to view the contents of the clipboard and shall be informed (e.g., in a message window) when they attempt to cut or copy an object whose size exceeds the capacity of the clipboard. The clipboard is only displayed at the request of the user.
6. The clipboard transfers operations of Cut, Copy, and Paste shall be performed using the Edit menu of an application.
7. Clipboard transfer shall be available whenever an editable object has keyboard focus.
8. A clipboard transfer operation shall be invoked from pull down or pop-up menus and have standard keyboard bindings. Access to clipboard transfer shall be provided in consistent fashion throughout an application.
9. Keyboard accelerators shall be available to perform other editing operations (e.g., Clear, Delete).
10. Users shall be capable of viewing the clipboard contents and shall be informed when they cut/copy an object of excessive size for the clipboard.
11. Cut operations shall be performed by the following: Selecting Cut, <Ctrl> + <X>, and <Shift> + <Delete>).
12. Cut shall clear the previous contents of the clipboard, store a copy of the source object in the clipboard, and remove the source object from the window.
13. If the cut object is a graphic, the space where the graphic object was previously located shall be left blank. If the cut object is text, the remaining text shall be compressed in the text widget.
14. Copy shall clear the previous contents of the clipboard and store a copy of the source object in the clipboard; the source object shall stay in its original location.
15. Copy operations shall be performed by the following: selecting Copy, <Ctrl> + <C>, <Ctrl> + <Insert>).
16. Paste shall duplicate the object in the clipboard to a new location.

<p>17. Selecting Paste, &lt;Ctrl&gt; + &lt;V&gt; and &lt;Shift&gt; + &lt;Insert&gt;) shall perform a paste operation.</p> <p>18. If the clipboard content is text, paste shall copy the clipboard contents to the location of text cursor and existing text appears to the left of cursor. If existing text is selected, paste shall copy the clipboard contents to the location of the selected text and remove the selected text. The pasted object shall remain in the clipboard until another object is cut/copied.</p> <p>19. If the clipboard content is graphic, paste shall copy the clipboard contents to the pointer location in a window with input focus. The pasted object shall remain in the clipboard until another object is cut/copied into it.</p> <p>20. A Copy Link entry in the Edit menu shall be used to place a link in the clipboard to selected elements of the target component so that the link can be placed in a destination by subsequent use of the Paste or Paste Link.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• The clipboard transfer technique transfers an object selection from a source to the clipboard and then from the clipboard to the destination.</li> </ul>

A.3.4.3	Primary transfer	<b>Source:</b> COMDAT (7.4.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Primary transfer operations shall include primary copy, primary move, and primary link.</li> <li>2. A primary transfer operation can be invoked from Pull Down or Pop-up Menus and have standard keyboard bindings. Access to primary transfer shall be provided in consistent fashion throughout an application.</li> <li>3. Primary transfers shall also be invoked using the T button.</li> <li>4. The default operation for primary transfer using the T button is copy.</li> <li>5. In an editable collection, a Primary Copy is performed by the following: T button click, &lt;Ctrl&gt; + T button click, and &lt;Alt&gt; + &lt;Ctrl&gt; + &lt;Insert&gt;.</li> <li>6. In an editable collection, a Primary Move is performed by the following: &lt;Shift&gt; + T button click and &lt;Alt&gt; + &lt;Shift&gt; + &lt;Delete&gt;.</li> <li>7. In an editable collection, a Primary Link is performed by the following: &lt;Ctrl&gt; + &lt;Shift&gt; + T button click.</li> </ol>		

8. Editing commands (such as move, copy, and delete) shall be provided for adding, inserting, or deleting text/program segments.
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• The primary transfer technique transfers the primary selection directly to a destination without using the clipboard for immediate storage of data.</li> </ul>

A.3.4.4	Quick transfer	<b>Source:</b> COMDAT (7.4.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The quick transfer operations of quick copy, quick cut, and quick link are available.</li> <li>2. Quick transfers shall be invoked using the &lt;Alt&gt; + T button.</li> <li>3. The default operation for quick transfer using the T button is copy.</li> <li>4. Text components must support quick transfer.</li> <li>5. If quick transfer is supported, &lt;Alt&gt; + T button motion or &lt;Alt&gt; + &lt;Ctrl&gt; + T button motion must temporarily select elements in the specified range and, on release, must copy them to the insertion position of the destination component.</li> <li>6. If quick transfer is supported, &lt;Alt&gt; + &lt;Shift&gt; + T button motion must temporarily select elements in the specified range and, on release, must move them to the insertion position of the destination component.</li> <li>7. If quick transfer is supported, &lt;Alt&gt; + &lt;Ctrl&gt; + &lt;Shift&gt; + T button motion must temporarily select elements in the specified range and, on release, must place a link to them at the insertion position of the destination component.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• A quick transfer technique allows the user to indicate a range of objects (called a secondary selection) that are transferred to the destination component.</li> </ul>		

A.3.4.5	Drag transfer (Drag and Drop)	<b>Source:</b> COMDAT (7.4.5)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. A drag and drop operations shall be executed by pressing the primary button while pointing to an object, moving the input device while holding the button down, and then releasing the button at the destination.</li> <li>2. The default transfer operation shall be move.</li> <li>3. &lt;Cancel&gt;, or equivalent, shall cancel a drag operation and return the object being dragged to the original location.</li> <li>4. Elements moved within a component shall remain selected after they have been moved.</li> <li>5. Dragging a set of selected elements shall drag the entire collection.</li> <li>6. Dragging in overlapping elements shall occur on the highest draggable element in the stack.</li> <li>7. The pointer shape shall change to a drag icon during the drag operation and then back to pointer at the completion of the operation.</li> <li>8. The drag icon shall contain a source indicator and may contain operations and state indicators.</li> <li>9. When users must repeatedly move objects by means of a drag transfer action and the drag destination is to the same or a default location, implement redundant transfer methods shall be used. Redundant transfer methods shall include the capability to drag a set of selected objects or double-click on an object as a command accelerator for the drag action.</li> <li>10. A drag move operation shall be executed by holding down &lt;Shift&gt; while dragging the object using the T button. If no modifier key is used (that is, just the T button is pressed on an object) the default operation is a move.</li> <li>11. A drag copy operation shall be executed by holding down &lt;Ctrl&gt; while dragging the object using the T button.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Movable or copyable text and objects shall be able to be moved or copied to a new location by using the pointer and the drag function.</li> <li>• If a metaphor is used, its representation should be sufficiently recognizable. For example, to delete a document in an office environment, the user can select the document icon, drag it over to the waste paper can and drop the document in the can in order to “throw it</li> </ul>		

away”.

### A.3.5 Interactive Controls

A.3.5.1	Object-Action Selection	<b>Source:</b> COMDAT (7.5.1)
<b>Guideline(s):</b>  1. Users shall first select an object, and then they shall select an action to perform on that object.		
<b>Discussion:</b> <ul style="list-style-type: none"><li>• Enable the user to interact directly with objects and minimize the use of indirect techniques. Whether they are dragging an object to relocate it or navigating to a location in a document, users should see how their actions affect the objects on the screen.</li><li>• Direct manipulation allows people to feel that they are directly controlling the objects represented by the computer. According to the principle of direct manipulation, an object on the screen remains visible while a user performs physical actions on the object. When the user performs operations on the object, the impact of those operations on the object is immediately visible. For example, a user can move a file by dragging an icon that represents it from one location to another or can position a cursor in a text field by directly clicking the location where the cursor should be placed</li><li>• This form of interaction is common with all Microsoft applications; therefore, the users can leverage their past experiences when working with JCDS21 applications.</li></ul>		

A.3.5.2	User control interaction	<b>Source:</b> COMDAT (7.5.2)
<b>Guideline(s):</b>  2. Applications shall execute an action only in response to explicit user input.  3. Users may take actions that will interrupt or terminate a process.		
<b>Discussion:</b> <ul style="list-style-type: none"><li>• Allow the user to initiate and control actions. The user should always feel in control of the software rather than feeling controlled by it. People learn best when they're actively engaged. Too often, the system acts and the user merely reacts within a limited set of</li></ul>		

options. In other instances, the system “takes care” of the users by offering only those alternatives that "protect" them from making detailed decisions or destroying data. This approach mistakenly puts the system, not the user, in control.

A.3.5.3	Error detection	<b>Source:</b> COMDAT (7.5.5)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The application shall not execute a user-requested action that is considered invalid. Instead, an error message shall be displayed.</li> <li>2. A capability shall be provided to facilitate detection and correction of errors after keying in but before entering the information into the system. While errors shall be detected early, error checking shall occur at logical data entry breaks (e.g., at the end of data fields rather than character-by-character) in order to avoid disrupting the user.</li> <li>3. User errors shall be minimized by use of software checks of user entries for validity of item, sequence of entry, completeness of entry, and range of value.</li> <li>4. When users make multiple errors with a single action, they shall be notified of each error. Multiple occurrences of the same error shall not appear in separate windows.</li> <li>5. When users make multiple errors with a single action, they will be provided with an immediate description of the error and the total number of additional errors detected, as in a word processing spell checker. There shall be some means for the user to request and correct sequential display of error messages.</li> <li>6. To prompt for correction of an error in stacked commands, the system shall display the stacked sequence with the error highlighted. Where possible, a procedure shall be provided to correct the error and salvage the stack.</li> <li>7. A computer-detected error, as well as the error message, shall be continuously displayed until the error is corrected.</li> <li>8. When an error is repeated, feedback shall show that an attempted correction was processed.</li> <li>9. Users shall be required to correct only an invalid action and not to repeat the entire sequence. The system shall permit correction of individual errors without requiring re-entry of correctly entered commands or data elements.</li> <li>10. After making a correction users shall be able to execute the same action for re-entry that was used for the original entry.</li> <li>11. Error messages shall first state the problem followed by a statement of appropriate</li> </ol>		

solutions.
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>Refer to Section 5.9 for guidance supporting the design and implementation of alarms and error messages.</li> </ul>

A.3.5.4	Undo Capability	<b>Source:</b> COMDAT (7.5.6)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>Users shall be allowed to undo the most recent selection or action unless the selection or action required explicit destruction.</li> <li>Users shall be allowed to execute a multi-level undo.</li> <li>In addition to being able to undo commands, users shall be able to deselect objects, return an object to its prior state before an action was executed, and retrieve information that was removed from the screen.</li> <li>Users shall be allowed to re-do the most recent, or specified “undone” transaction.</li> <li>Irreversible actions (actions that cannot be undone) shall be labelled and clearly separated from actions that are reversible.</li> <li>If an action cannot be labelled as irreversible, the user shall be presented with a warning and asked to confirm the irreversible action.</li> <li>When an undo option is provided, the wording shall change dynamically to reflect the action that can be undone. The undo option shall always begin with Undo. For example, if the most recently executed action is a cut, the undo option shall be worded Undo Cut.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>Forgiveness assists users with recovering from errors when they occur as well as encourages exploration through an application. One way to address error recovery is through user correction. By making actions reversible, for instance, users can easily undo actions without repercussions. Users quickly learn to rely on the existence of undo; therefore, make it continually available as a generic command that undoes a special category of user actions.</li> </ul>		

## Annex B Controls

---

### B.1 General

Controls are graphic objects that represent the properties or operations of other objects. Each control has a unique appearance and operation designed for a specific form of interaction. The following is a list of controls that JCDS21 application and tools might have.

1. buttons
2. combo boxes
3. checkboxes
4. fields
5. labels
6. list box
7. scroll bars
8. context menus
9. explorer tree views
10. group boxes
11. selection lists
12. application menus
13. toolbars
14. dialogs in general
15. tab controls
16. property sheets
17. tables

General guidelines regarding control characteristics, actions and areas are presented in the following sections. Specific guidelines pertaining to the characteristics, appearance, and interactions are reserved for the MSWUE.

B.1.1	Characteristics	<b>Source:</b> COMDAT (8.1)
<b>Guideline(s):</b> <ol style="list-style-type: none"><li>1. All of the types of controls in a window shall be identifiable solely based on their appearance.</li><li>2. All controls with the same function shall have the same appearance.</li><li>3. Text/graphics in a window shall be clearly different in appearance from standard controls.</li><li>4. Controls performing similar or related functions shall be physically grouped together.</li><li>5. Grouped controls shall be framed and shall be clearly labelled to indicate the functions they perform.</li><li>6. Controls that are temporarily unavailable shall be dimmed and shall not be available for selection.</li><li>7. Controls that are never available to users shall not appear in a window.</li></ol>		

8. Lines surrounding a group of controls shall follow the look and labelling of Microsoft Windows styles.
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• It is easier for a user to recognize something as opposed to recalling it; therefore, the function of the control should be obvious and explicit so that users' can take advantage of their knowledge of other applications.</li> </ul>

B.1.2	Actions	<b>Source:</b> COMDAT (8.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Control actions shall be minimized, consistent, make minimal memory demands of the user, and be sufficiently flexible to adapt to different user needs.</li> <li>2. Very frequent or safety-critical tasks shall be identified and shall be accomplished with a single operator action such as a keystroke or button press.</li> <li>3. Frequent or critical tasks shall be identified and shall be accomplished with no more than three operator actions (keystrokes, button presses, etc.)</li> <li>4. User acceptance of stored data or defaults shall be possible by a single confirming keystroke.</li> <li>5. Control actions to be selected from a discrete set of alternatives shall have those alternatives displayed prior to the time of selection.</li> <li>6. User control inputs shall result in a positive feedback response displayed to indicate performance of requested actions.</li> <li>7. If a default button is defined it shall be non-destructive. A destructive action is an action that cannot be reversed (e.g., missile launch), or cannot be routinely reversed without extensive or special procedures (e.g., unerase a file), and therefore shall be preceded by a user confirmation in a dialog box. The confirmation shall usually be the OK button (or equivalent), unless the action is destructive wherein the Cancel button shall be used as the default. The default can only be used when the default button is active.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Human factors analysis techniques are critical prior to application design in order to identify the operators' goals and tasks. This ensures the resulting design takes into consideration frequency and criticality of the operator tasks.</li> </ul>		

B.1.3	Control Areas	<b>Source:</b> COMDAT (8.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. The display shall include visual affordance for the operators.</li> <li>2. Controls for operations that are potentially destructive shall not be co-located with frequently used or confusable controls.</li> <li>3. Frequently used and critical controls shall be located at anchor points within the display or shall be organized in order of priority.</li> <li>4. Controls shall have a large active area so that the operator does not have to access a small target area.</li> <li>5. Direct access to the desired levels of controls shall be obtained without requiring intervening steps.</li> <li>6. Controls shall be sufficiently large to be accessed easily under expected operational conditions.</li> </ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• If the operator must search for a control then the operator's attention is devoted to the interface rather than the operational task. New layouts and designs should provide good visual affordance to improve navigation within the tactical controls. New controls should also address the sequence with which operators access the functions and should support the operational task flow.</li> </ul>		

B.1.4	Action icons	<b>Source:</b> COMDAT (8.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Action icons shall have unique graphic images so that users recognize the action performed. Action icons shall not conflict with action icons that are already defined.</li> <li>2. Action icon graphics in an application shall be consistent with icons in other applications in the system and shall be consistent with the icons that are already defined.</li> <li>3. Developers shall use a short text label in addition to a graphic for action icons. This is especially important if the function being represented is highly abstract.</li> </ol>		

4. Colours used in images in action icons shall be similar to other system colours and used in a consistent manner.
5. Graphics for action icons representing opposite actions shall be designed to mirror each other.
6. Graphics shall be presented in a common style and oriented consistently within the button.
7. An action icon shall be large enough for a user to see and understand the graphic and text label. An action icon shall also be large enough to be easily selectable via the pointing device.
8. Action icons shall be grouped by task relevance or frequency of use.
9. Action icons shall duplicate, but do not replace selections in a pull-down menu.

**Discussion:**

- If the operator must search for a control then the operator's attention is devoted to the interface rather than the operational task. New layouts and designs should provide good visual affordance to improve navigation within the tactical controls. New controls should also address the sequence with which operators access the functions and should support the operational task flow.

## **B.2 Menus and Navigation**

Navigation facilitates movement from one application or web page to another. Menus provide a means for the operator to navigation and access functions in a hierarchical fashion that requires decreased display space and decreased cognitive requirements. While this means accessing functions requires less memorization, it is generally slower for experienced operators who have already learned common function keys. The deeper a function is located in a menu hierarchy the longer it will take to access that function and the more difficult it will be to remember how to navigate to that function. The Microsoft Windows visual layout shall be followed.

## Annex C Special Functions and Formats

### C.1 Date/Time and Latitude/Longitude

C.1.1	Date/Time and Latitude/Longitude	<b>Source:</b> COMDAT (13.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. All dates shall be presented to and supplied by the operator in the DD MMM YY (10 MAY 68) format. The characters of the month shall be capitalized. The spaces between day and month and month and year may be omitted. If the day or year is a single digit it must be preceded by a zero. The notation 01JAN00 would mean January 1, 2000..</li> <li>2. Time shall be presented to and supplied by the operator in the HH:MM:SSZ (09:38:30Z) format. HH is the hour in a 24-hour day, MM is the minute with a leading zero if necessary, and SS is the second with the leading zero if necessary. Z is the time zone with Zulu time as the default. The seconds are optional. All colons are required and the Zulu (Z) shall be capitalized.</li> <li>3. The display shall include Zulu time and local time; Zulu time shall be presented above local time.</li> <li>4. The date time group shall be presented on the COMDAT status bar on the right-hand edge. (Note: Although the time is presented in the lower right hand corner of the screen in Microsoft Windows; the upper right hand corner was preserved in the COMDAT OMI to maintain consistency with other combat systems.)</li> <li>5. If the operator must know the time in multiple time zones, the application shall provide a separate time for each zone or provide the operator the ability to change the time zone displayed.</li> <li>6. A date time group shall be presented to and supplied by the operator in the DD HH:MM:SSZ MMM YY format (seconds are optional) where DD is the day with the leading zero if necessary, HH is the hour with a leading zero if necessary, MM is the minute with a leading zero if necessary, SS is the second with a leading zero if necessary, and Z is the time zone with Zulu as the default with MMM as the abbreviated month and YY is the last two digits of the year. All colons and spaces are required. (Note: DODSG, JMCIS, GCCS call for a similar display, without the colon separators or space between the day and the time.)</li> <li>7. Latitude and longitude information shall be displayed in separate fields.</li> <li>8. The latitude label may be abbreviated to Lat. Latitude shall be presented to and supplied by the operator in the DD°MM`SS.T" (09°06`30.3"N) format. A two-digit degree and the hemisphere (N or S) are required. Minutes, seconds, and tenths of seconds are optional. If</li> </ol>		

provided, minutes and seconds each must contain two digits. All symbols are required if preceded by a number and the hemisphere (N or S) shall be capitalized. A hyphen may be substituted for the degree and minute symbols DD-MM-SS.T (09-06-30.3) The system should know in which hemisphere it is located and use this as the default value.

(Note: CSFAB states that the abbreviation may be LAT or Lat. Lat is the preferred abbreviation because the abbreviation is not an acronym.)

9. The longitude may be abbreviated to Long. Longitude shall be presented to, and supplied by, the operator in the DDD°MM'SS.T" (090°06`30.3"E) format. DDD is the degrees of longitude, MM is the minutes of longitude and is optional; SS is the seconds of longitude. A three-digit degree and the hemisphere (E for East or W for West) are required. Minutes, seconds, and tenths of seconds are optional. If provided, minutes and seconds must each contain two digits. Seconds of longitude are only presented if minutes are presented. All symbols are required if preceded by a number. The hemisphere (E or W) shall be capitalized. A hyphen may be substituted for the degree and minute symbols DDD-MM-SS.T (090-06-30.3). The system should know in which hemisphere it is located and use this as the default value.
10. A latitude/longitude group shall be presented to and supplied by the operator in the DD°MM'SS.T" / DDD°MM'SS.T" (09°06`30.3"N /090°06`30.3"E) format. All symbols are required if preceded by a numeric value. Hyphens may replace degree and minute symbols and the second symbol may be omitted. Latitude shall always precede longitude by either being above or to the left of the longitude value. Two possible latitude/longitude groupings are shown below. The degrees and hemisphere is required in both latitude and longitude. Minutes, seconds, and tenths are optional in both but shall follow the guidance described in this document.
  - e. Lat: 09°06`30.3"N
  - f. Long: 090°06`30.3"E
  - g. Lat/Long: 09°06`30.3"N / 090°06`30.3"E

**Discussion:**

- Date/Time Groups. For Date/Time groups, the display is easier to read if colons are used to separate the time groups.
- Date Formats. The appropriate format for the date information depends on the use to which the information will be put. The YYMMDD format does not disambiguate the month and day fields across international conventions and thus is not recommended. The DD MMM YY format provides an advantage because it is consistent with the order of the elements in the preferred Date Time Group format. If the date is to be conveyed to other entities that are expecting the YYMMDD format then the international conventions affect the display of choice. If the operators must report or otherwise convey the two-digit number for the month, then support for the conversion shall be provided.

## C.2 List-to-List Transfers

C.2.1	List-to-list transfers	<b>Source:</b> COMDAT (13.4)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. A task window used to copy items from a source list (master list) to a destination list shall provide one Add and one Remove button. If many items are in a list an Add All and/or a Remove All button may also be included. The labels Add and Remove may be altered to more closely fit the task for which the list-to-list transfer is being used.</li><li>2. List to list transfers shall follow the Microsoft Windows conventions.</li><li>3. Vertically arranged push buttons shall be located between the two lists and shall be used to invoke the functions Add All, Add, Remove, and Remove All. Add All shall be on the top, then Add, then Remove, and Remove All on the bottom. Selecting one or multiple items from the original list and selecting Add shall move these items to the destination list. Selecting Add All shall move all items from the original list to the destination list. Selecting one or multiple items from the destination list and selecting Remove shall move these items to the original list. Selecting Remove All shall move all items from the destination list to the original list.</li><li>4. In some tasks it may be preferable to copy instead of move items from the original list to the destination list. Items shall then be marked with an asterisk to show they have already been copied. Multiple copies of the same items shall not be allowed in the original or destination lists. The push buttons from top to bottom shall read: Copy All, Copy, Remove, Remove All.</li><li>5. The Add/Copy push button is only enabled when an item is selected in the original list. The Remove push button is only enabled when an item is selected in the destination list. The Add/Copy All and Remove All push buttons are always enabled unless there are no items in the original or destination lists respectively. If an item is selected in the original list, the Add/Copy button shall be the default. If an item is selected in the destination list, the Remove button shall be the default.</li><li>6. Double clicking an object in either list shall move or copy it to the other list.</li><li>7. The Remove button shall be enabled and the Add button shall be disabled if an item in the destination list is the only selected item.</li><li>8. The Add button shall be enabled and the Remove button shall be disabled if an item in the source list is the only selected item.</li><li>9. The user shall not be allowed to remove an item from the source list.</li><li>10. Moving an item from the source list to the destination list in a list-to-list transfer box shall</li></ol>		

<p>not permanently remove the item from the source list.</p> <p>11. List-to-list transfer windows shall support drag transfer actions (drag and drop) from one list to another.</p> <p>12. The Add and Remove buttons shall include a graphic showing the direction of transfer.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• A list-to-list transfer is used to copy (or move) items from a source list (master list) to a destination list.</li> </ul>

### C.3 Message Handling

C.3.1	Message handling window	<b>Source:</b> COMDAT (13.5.1)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Except for broadcast communication systems, the transmitter of each message in inter-user communications shall be identified automatically.</li> <li>2. Message preparation windows shall follow the same design as data entry windows.</li> <li>3. Users shall be given basic message header fields that are supported in specifying the message address.</li> <li>4. Option menus may available for selecting from limited sets of frequently used terms.</li> <li>5. When replying to a message, the appropriate addressee(s) shall be provided automatically.</li> <li>6. Users shall have the capability to build and maintain lists of common addresses and select from these lists when preparing messages, if that functionality is required.</li> <li>7. If functionality is required by an application, addresses are checked prior to transmission; users can correct errors before sending.</li> <li>8. Preformatted standard forms are available and format control during entry is automatic.</li> <li>9. Users shall be able to specify data, incorporate data files, and save during preparation/completion.</li> <li>10. Each individual data group or message shall contain a descriptive title, phrase, word or similar device to designate the content of the group or message.</li> <li>11. The notification of the message (or alert) shall indicate, at minimum, an indication of the number of alerts in each category (e.g., flash messages, system messages, or tactical</li> </ol>		

messages).

**Discussion:**

- A message window (sometimes called a message box) is a secondary window that provides users with non-critical information, progress information about lengthy processes, alerts to unusual events, and/or warnings of potential dangers. Message windows may be modal or modeless

C.3.2	Message transmission	<b>Source:</b> COMDAT (13.5.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Message transmission procedures shall be designed to minimize the user actions required.</li><li>2. Users shall have the capability to initiate message transmission directly (e.g., select a transmit push button) or can set a transmission time.</li><li>3. If a message cannot be sent immediately, it shall be queued automatically.</li><li>4. If functionality is required by an application, users may assign message priorities and cancel or terminate a transmission.</li><li>5. Status feedback shall be available that confirms messages sent and indicates failures.</li><li>6. Users shall be able to specify what feedback they want to receive, an automatic log of this information shall be maintained.</li><li>7. If functionality is required by an application, undelivered messages are saved in the event that there is transmission failure.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• No additional discussion provided given the nature of the guidelines.</li></ul>		

C.3.3	Message receipt	<b>Source:</b> COMDAT (13.5.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Users shall be informed when high-priority messages are received.</li> <li>2. Message notification shall not interrupt operator tasks, but shall provide an indication of urgency.</li> <li>3. The presence of non-system generated messages that do not require immediate operator intervention shall be indicated.</li> <li>4. The messages shall be presented in order of priority followed by the most recently received messages.</li> <li>5. The highest priority messages shall be indicated with a colour coded with a red (or magenta) symbol in the message notification area.</li> <li>6. The title bar of a message window shall be colour coded to match the priority code of the message (Red for Flash messages)</li> <li>7. If functionality is required by an application, incoming messages shall be automatically queued by time and priority; logs shall be maintained.</li> <li>8. Users can review summary information, display messages, save/file, and discard.</li> <li>9. A message shall be displayed in a text window and users can scroll, save, and print the message.</li> <li>10. During login, a system shall provide users with a list of new messages received.</li> <li>11. Data transmission functions shall be integrated with other information handling functions within a system. A user shall be able to transmit data using the same computer system and procedures used for general entry, display and other processing of data.</li> <li>12. Procedures for preparing, sending and receiving data and messages shall be consistent from one transaction to another, and consistent with procedures for other information handling tasks.</li> <li>13. The data transmission procedures shall minimize memory load on the users by providing computer aids for automatic insertion of standard information, such as headers and distribution lists.</li> <li>14. Users shall be able to interrupt message preparation, review, or disposition and then resume any of those tasks from the point of interruption.</li> </ol>		

<p>15. Where message formats conform to a defined standard or are predictable in other ways, stored forms shall be provided to aid users in message preparation.</p> <p>16. Users shall be able to incorporate an existing data file in a message, or to combine several files into a single message for transmission and to combine stored data with new data when preparing messages for transmission. It shall not be necessary to re-enter any data already entered for other purposes.</p> <p>17. When users must specify the address for messages, prompting shall be provided to guide the user in the process.</p> <p>18. Users shall be provided with an on-line directory showing all acceptable forms of message addressing for each destination in the system, and for links to external systems.</p> <p>19. Computer aids shall be provided so that a user can search an address directory by specifying a complete or partial name. It shall also be possible to extract selected addresses from a directory for direct insertion into a header in order to specify the destination(s) for a message.</p>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• Consistent positioning of messages allows user to look for and find the messages more easily.</li> </ul>

## C.4 Spell Checking

C.4.1	Spell checking	<b>Source:</b> COMDAT (13.6)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. Spell checking displays and operation shall be consistent with Microsoft Windows styles.</li> <li>2. A task window used to perform spell checking on a text document or message segment shall display the unknown word to the user in an uneditable text field.</li> <li>3. An unknown word identified by the spell checker shall appear in an editable text field so that the user can edit the unknown word.</li> <li>4. The system shall provide a default dictionary of common words, abbreviations, etc., used in the application. This dictionary shall be tailored for the application task.</li> <li>5. The user shall be able to view suggestions from the dictionary on how to correct an unknown word.</li> <li>6. The user shall be able to select a word from the list of suggestions and have the option of</li> </ol>		

editing the word before replacing the unknown word.

7. The user shall have the option of replacing all instances of a given unknown word with the specified corrected word or replacing each instance individually. If a user chooses to globally replace all instances of a given unknown word, the spell checker shall automatically replace the word without acknowledgment from the user and shall not interrupt the spell checking process.
8. The user shall be allowed to skip an unknown word without changing it.
9. The user shall be given the option to skip all instances of an unknown word for the remainder of the document or to skip the current instance only.
10. The user shall be able to specify the direction of the spell check relative to the current position.
11. The user shall be able to cancel the spell check operation at any time.
12. Spelling and other common errors shall not produce valid system commands or initiate transactions different from those intended. When possible, the system shall recognize common misspellings of commands and execute the commands as if spelling had been correct.
13. Computer-corrected commands, values, and spellings shall be displayed and highlighted for user confirmation.

**Discussion:**

- Spell checking can highlight possible errors and provide suggestions for corrections.

## C.5 Imagery Manipulation

C.5.1	Imagery appearance	<b>Source:</b> COMDAT (13.8.1)
<b>Guideline(s):</b> <ol style="list-style-type: none"><li>1. An imagery window shall contain an imagery display and a set of tools for manipulating the image.</li><li>2. An imagery window shall include identifying information about the image displayed and this information shall be displayed in a standard location.</li><li>3. When an entire image does not fit on a screen, the users shall be provided with the capability to scroll, pan, and zoom.</li></ol>		

<ol style="list-style-type: none"> <li>4. An imagery window shall contain resize handles and resizing the window shall have an effect on the image contained in the window.</li> <li>5. An imagery system shall permit users to access and retrieve an image for display from a directory of images.</li> <li>6. An imagery system shall permit users to specify the name of the image or search the directory of images for images matching user-defined criteria, including wild card searches.</li> <li>7. An imagery system shall permit users to print in full-resolution or low-resolution an image displayed in an imagery window.</li> </ol>
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion provided given the nature of the guidelines.</li> </ul>

C.5.2	Imagery viewing functions	<b>Source:</b> COMDAT (13.8.2)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. An imagery window shall permit users to roam or pan an image in both the horizontal and vertical direction.</li> <li>2. When panning or roaming an image, automatic, jump, manual, and patterned roam control shall be available and an imagery system shall permit users to specify the rate, direction, and area of interest as appropriate.</li> <li>3. As users pan or roam an image, the imagery system shall permit users to tag specific areas of interest for later recall while the image remains in the display area.</li> <li>4. An imagery system shall permit users to zoom an image in predefined increments or in user-defined increments.</li> <li>5. An imagery system shall support image chipping. Image chipping allows users to select and designate regions of an image for storage.</li> <li>6. An imagery system shall permit users to create and manipulate annotations to display with an image. Annotations shall include text, lines, icons, geometric shapes, colours, and patterns.</li> <li>7. Image annotations shall be edited, deleted, repositioned, resized, saved, and retrieved without altering the underlying imagery data.</li> </ol>		

8. When an image is manipulated (e.g., resized, rescaled, etc.), the scale and orientation of image annotations shall be adjusted to accommodate the changes. However, textual annotations shall remain constant.
<p><b>Discussion:</b></p> <ul style="list-style-type: none"> <li>• No additional discussion provided given the nature of the guidelines.</li> </ul>

C.5.3	Advanced imagery viewing functions	<b>Source:</b> COMDAT (13.8.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"> <li>1. An imagery system shall provide image enhancement filtering capabilities to control the dynamic range of imagery data by modification of intensity and contrast.</li> <li>2. An imagery system shall permit users to select and examine (e.g., zoom, roam) a region of interest within an image leaving the remainder of the image unaffected.</li> <li>3. Imagery windows shall provide measurement functions for computing lengths, areas, and volumes from dimensions or angles. Imagery windows shall also provide the capability to perform isotropic pixel correction (i.e., convert rectangular pixels to square pixels for display purposes).</li> <li>4. Imagery windows shall permit users to create a mirror view of the image so the image can be adjusted if the negative was inverted when scanned.</li> <li>5. Imagery windows shall provide a default image rotation where vertical objects are oriented toward the top of a window and an automatic north rotation where north is oriented at the top of the window. Imagery windows shall also permit users to rotate images in user-defined increments.</li> <li>6. When an image is manipulated in an image window, the scale and orientation of annotations shall be adjusted according to the type and nature of the symbol.</li> <li>7. An imagery system shall permit users to plot user-selected geographic data on the image, including frame-by-frame plotting to animate the data in either a forward or reverse time direction.</li> <li>8. An imagery system shall permit users to register (i.e., transform an image so that it aligns with either another image or a map projection) geo-referenced images acquired from the same or different sensors and display them together.</li> <li>9. An imagery system shall permit users to perform concurrent geometric manipulations on</li> </ol>		

separate geo-referenced windows that have overlapping geographic coverage.

10. Users shall be able to "leave" together multiple geo-referenced windows and manipulate (e.g., roam, zoom, rotate) all the slaved windows concurrently and relatively (i.e., with the window centres maintained at a common centre latitude/longitude position), despite differences in the amount and distance coverage between windows

**Discussion:**

- No additional discussion provided given the nature of the guidelines.

## Annex D Security and Simulation

---

### D.1 Security and Simulation Characteristics

D.1.1	Design for system security	<b>Source:</b> COMDAT (11.1.1)
<b>Guideline(s):</b> <ol style="list-style-type: none"><li>1. Data shall be protected from unauthorized use, potential loss from equipment failure, and user errors.</li><li>2. Automated measures shall be provided to minimize data loss from intruders in a system or from errors by legitimate users.</li></ol>		
<b>Discussion:</b> <ul style="list-style-type: none"><li>• Security safeguards are the protective measures and controls that are prescribed to meet the security requirements specified for a system. Those safeguards may include but are not necessarily limited to: operational procedures, physical security, or hardware and software features.</li></ul>		

D.1.2	Design for system simulation	<b>Source:</b> COMDAT (11.1.2)
<b>Guideline(s):</b> <ol style="list-style-type: none"><li>1. When simulated data and system functions are provided real data shall be protected and real system use shall be clearly distinguished from all simulated operations.</li><li>2. In applications where either real or simulated data can be displayed, a clear indication of simulated data shall be included.</li></ol>		
<b>Discussion:</b> <ul style="list-style-type: none"><li>• No additional discussion provided given the nature of the guidelines.</li></ul>		

## D.2 System Login

D.2.1	System login	<b>Source:</b> COMDAT (11.2), HFDS 2003 (ch11)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. Where two or more users must have simultaneous access to the computer program or data processing results from multiple personal equipment interfaces, the operation by one person shall not interfere with the operations of another person unless mission survival may be contingent upon pre-emption.</li><li>2. Provisions shall be made so that a pre-empted user can resume operations at the point of interference without information loss.</li><li>3. Users shall complete a login procedure before the system functions can be accessed.</li><li>4. In applications where users must login to the system, login shall be a separate procedure that must be completed before a user is required to select among any operational options.</li><li>5. Users shall be provided with feedback relevant to the login procedure, the feedback shall indicate the status of the inputs.</li><li>6. A system shall provide access to only those applications to which a user is allowed access.</li><li>7. Operators must be able to login in any operational role so as to assume any role within the operations room as required.</li><li>8. If appropriate, users shall also complete a login for individual or groups of applications.</li><li>9. If the system is unavailable, a message shall be displayed indicating the system status and when the system will be available.</li><li>10. If a user cannot login to a system, a prompt shall be provided to explain the reason for this inability.</li><li>11. Login processes shall require minimum input from the user consistent with the requirements prohibiting illegal entry.</li><li>12. A login window shall be displayed on the screen when users begin a session on a system.</li><li>13. Appropriate prompts for login shall be automatically displayed on the user's terminal with no special action required other than turning on the terminal.</li><li>14. A login window shall contain two text fields, one each for entering user identification and password.</li></ol>		

15. User identification procedures shall be as simple as possible, consistent with adequate data protection.
16. The password shall not be echoed on the display. An asterisk (\*) shall be displayed for each character when inputting secure passwords during login.
17. When passwords are required, users shall be allowed to choose their own passwords since a password chosen by a user will generally be easier for that individual to remember. Guidelines for password selection shall be given so that users will not choose easily guessable ones. (Note: Before requiring passwords as part of the login procedure, the extent of security measures required should be determined.)
18. Users shall be allowed to change passwords whenever they choose; all passwords shall be changed at periodic intervals (not to exceed six months).
19. The text fields shall not provide cues as to the number of characters required for a password.
20. The layout and format of objects in the login window shall have the system title or logo centred at the top. The Login Name and Password shall be on two separate lines below the title. The window shall have button controls at the bottom that shall include, at minimum, the Login button in addition to the usual controls.
21. Users shall enter a valid identification and password before a session is initiated.
22. If an invalid identification and/or password is entered, an error message shall be displayed.
23. Users who fail repeatedly to login shall be locked out of the system and informed that they must contact the system administrator.

**Discussion:**

- Identification is the process that enables the security safeguards to recognize a user name (usually through a machine-readable name) as an identical match to a name previously listed in an authorized user file.
- Authentication is the act of identifying and confirming the eligibility of a station, originator, or user to access specific categories of information. Authentication is a measure designed to provide protection against fraudulent entry or transmissions by establishing the validity of a transmission, message, station, or originator.
- Authorization is granting, to a user or user group, the right of access to a program, a process, or information.

## D.3 System Logout

D.3.1	System logout	<b>Source:</b> COMDAT (11.3)
<p><b>Guideline(s):</b></p> <ol style="list-style-type: none"><li>1. To terminate a session, users shall select logout from the system menu.</li><li>2. Users shall be prompted to save data prior to logout if there is any unsaved data.</li><li>3. Users shall be prompted to logout of any applications that required they login.</li><li>4. During logout, all processing in application windows shall stop, and all windows shall be closed.</li><li>5. When logout is complete, the initial login window shall be displayed.</li><li>6. If there is auto logout, the system shall incorporate the standard length of user inactivity before logout occurs.</li><li>7. Users, or the system administrator, shall be permitted to modify the time before auto logout occurs. (Note: The provision was modified to specify the system administrator.)</li><li>8. A message is displayed during inactivity indicating the action needed to avoid auto logout.</li><li>9. For auto logout, unsaved data shall be saved, with a message indicating logout and file name.</li><li>10. If a partial hardware/software failure occurs, the program shall allow for orderly shutdown and establishment of a checkpoint so restoration can be accomplished without loss of computing performed to date.</li></ol>		
<p><b>Discussion:</b></p> <ul style="list-style-type: none"><li>• Exceptions may be required with respect to closing all applications windows upon system logout. This implementation will be dependent on the operational context.</li></ul>		

This page intentionally left blank.

## List of symbols/abbreviations/acronyms/initialisms

---

C2	Command and Control
C4I	Component Command, Control, Communications, Computer, and Intelligence
COMDAT	Canadian Command Decision Aiding Technology
COTS	Commercial off the shelf
DRDC	Defence Research & Development Canada
FAA	Federal Aviation Administration
GUI	Graphical User Interface
HF	Human Factors
HFDC	Human Factors Design Guide
HFDS	Human Factors Design Standard
HFES	Human Factors Engineering of Software
JCDS 21	Joint Command Decision Support for the 21 <sup>st</sup> Century
JIMP	Joint, Interagency, Multinational, Public
MSWUE	Microsoft Windows User Experience
OMI	Operator Machine Interface
PDA	Personal Digital Assistant
PUM	Portlet Usability Model
TDP	Technology Demonstration Project
URL	Universal Resource Locator
W3C	World Wide Web Consortium
WMBP	Windows Mobile-Based Platforms
ZUI	Zoomable User Interface

# Distribution list

---

Document No.: DRDC Toronto CR 2009-047

## **LIST PART 1: Internal Distribution by Centre**

DRDC Toronto

- 1 Wenbi Wang
- 1 Kevin Trinh

---

2 TOTAL LIST PART 1

## **LIST PART 2: External Distribution by DRDKIM**

- 1 Library and Archives Canada
- 1 DRDC Valcartier – Dany Dessureault (Dany.Dessureault@drdc-rddc-gc.ca)
- 1 DRDC Valcartier – Adel Guitouni (Adel.Guitouni@drdc-rddc.gc.ca)

---

3 TOTAL LIST PART 2

**5 TOTAL COPIES REQUIRED**

This page intentionally left blank.

# UNCLASSIFIED

<b>DOCUMENT CONTROL DATA</b> (Security classification of the title, body of abstract and indexing annotation must be entered when the overall document is classified)		
<b>1. ORIGINATOR</b> (The name and address of the organization preparing the document, Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's document, or tasking agency, are entered in section 8.)  Publishing: DRDC Toronto 1133 Sheppard Ave. W., Toronto, ON Performing: Contractor CAE Professional Services, 1135 Innovation Dr., Suite 300, Ottawa, ON  Monitoring: Contracting:		<b>2. SECURITY CLASSIFICATION</b> (Overall security classification of the document including special warning terms if applicable.)  <b>UNCLASSIFIED</b>
<b>3. TITLE</b> (The complete document title as indicated on the title page. Its classification is indicated by the appropriate abbreviation (S, C, R, or U) in parenthesis at the end of the title)  Joint command decision support 21st century technology demonstration: Human factors style guide (U) (U)		
<b>4. AUTHORS</b> (First name, middle initial and last name. If military, show rank, e.g. Maj. John E. Doe.)  CAE Professional Services		
<b>5. DATE OF PUBLICATION</b> (Month and year of publication of document.)  March 2009	<b>6a NO. OF PAGES</b> (Total containing information, including Annexes, Appendices, etc.)  169	<b>6b. NO. OF REFS</b> (Total cited in document.)  23
<b>7. DESCRIPTIVE NOTES</b> (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)  Contract Report		
<b>8. SPONSORING ACTIVITY</b> (The names of the department project office or laboratory sponsoring the research and development – include address.)  Sponsoring: Tasking:		
<b>9a. PROJECT OR GRANT NO.</b> (If appropriate, the applicable research and development project or grant under which the document was written. Please specify whether project or grant.)  15a	<b>9b. CONTRACT NO.</b> (If appropriate, the applicable number under which the document was written.)  W7701-5-3182-AT11	
<b>10a. ORIGINATOR'S DOCUMENT NUMBER</b> (The official document number by which the document is identified by the originating activity. This number must be unique to this document)  DRDC Toronto CR 2009-047	<b>10b. OTHER DOCUMENT NO(s).</b> (Any other numbers under which may be assigned this document either by the originator or by the sponsor.)	
<b>11. DOCUMENT AVAILABILITY</b> (Any limitations on the dissemination of the document, other than those imposed by security classification.)  Unlimited distribution		
<b>12. DOCUMENT ANNOUNCEMENT</b> (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, when further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.)  Unlimited announcement		

UNCLASSIFIED

## UNCLASSIFIED

### DOCUMENT CONTROL DATA

(Security classification of the title, body of abstract and indexing annotation must be entered when the overall document is classified)

13. **ABSTRACT** (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

(U) The aim of the Joint Command Decision Support for the 21st Century Technology Demonstrator Project (JCDS 21 TDP) is to demonstrate a Joint Net-enabled, Collaborative Environment to achieve Decision Superiority at the strategic and operational levels of command. JCDS 21 applications are intended for short-duration, domestic operations that may require military and civil responders to work together to resolve the impact of natural or man-made incidents. This document is intended to leverage recent background research which identified relevant documentation that could be used as a basis for developing a high-level usability framework for the JCDS 21 TDP Style Guide; this Style Guide contains general guidance for the development of Graphical User Interfaces (GUIs) for use in a Windows-based environment and within a joint Command and Control (C2) environment.

This work was completed under sub-contract to Fujistu Consulting and with cooperation from Defence Research & Development Canada (Toronto) (DRDC-Toronto) and Prolity.

(U) Le but du Projet de démonstration de technologies – Aide à la décision des commandements interarmées pour le XXI<sup>e</sup> siècle (PDT ADCI 21) est de démontrer qu'un cadre collaboratif réseautique peut permettre à un commandement interarmées d'obtenir la supériorité décisionnelle aux niveaux stratégique et opérationnel. Les applications du PDT ADCI 21 sont conçues pour les opérations nationales de courte durée qui peuvent nécessiter une certaine collaboration entre les intervenants civils et militaires pour faire face à l'impact d'une catastrophe naturelle ou causée par l'homme. Le présent document vise à tirer profit des récents travaux de recherche qui ont identifié la documentation qui pourrait servir de base à l'élaboration d'un cadre d'utilisabilité de haut niveau pour le guide de style du PDT ADCI 2. Ce guide de style contient des directives générales sur le développement d'interfaces graphiques personnalisées (GUI) conçues pour être utilisées dans un environnement Windows et dans un environnement de commandement et de contrôle (C2) interarmées. Ce travail a été effectué en sous-traitance par Fujistu Consulting, en collaboration avec RDDC Toronto et Prolity.

14. **KEYWORDS, DESCRIPTORS or IDENTIFIERS** (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

(U) Joint command, decision support, human factor style guide

UNCLASSIFIED