

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

**WEB-BASED INFORMATION MANAGEMENT SYSTEM FOR
THE INVESTIGATION, REPORTING, AND ANALYSIS OF
HUMAN ERROR IN NAVAL AVIATION MAINTENANCE**

by

Anthony R. Boex

September 2001

Thesis Advisor:
Associate Advisors:

John K. Schmidt
George A. Zolla
Robert C. Figlock

Approved for public release; distribution is unlimited

Report Documentation Page

Report Date 30 Sep 2001	Report Type N/A	Dates Covered (from... to) -
Title and Subtitle Web-Based Information Management System for the Investigation, Reporting, and Analysis of Human Error in Naval Aviation Maintenance		Contract Number
		Grant Number
		Program Element Number
Author(s) Boex		Project Number
		Task Number
		Work Unit Number
Performing Organization Name(s) and Address(es) Research Office Naval Postgraduate School Monterey, Ca 93943-5138		Performing Organization Report Number
Sponsoring/Monitoring Agency Name(s) and Address(es)		Sponsor/Monitor's Acronym(s)
		Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited		
Supplementary Notes		
Abstract		
Subject Terms		
Report Classification unclassified		Classification of this page unclassified
Classification of Abstract unclassified		Limitation of Abstract UU
Number of Pages 316		

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2001		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE. Web-Based Information Management System for the Investigation, Reporting, and Analysis of Human Error in Naval Aviation Maintenance				5. FUNDING NUMBERS
6. AUTHOR(S) Boex, Anthony R.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000				8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Safety Center, Norfolk VA 23411 Federal Aviation Administration Washington, DC 20591 National Aeronautics and Space Administration, Moffett Field, CA 94035-1000				10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE
13. ABSTRACT (maximum 200 words) The purpose of this thesis is to examine the development of a Web application to display, analyze, and produce reports of human error involvement and patterns in aviation maintenance mishaps. The Human Factors Analysis and Classification System-Maintenance Extension (HFACS-ME) taxonomy, a framework for classifying and analyzing the presence of maintenance errors that lead to mishaps, is the foundation of this tool. The target audience for this tool includes safety and maintenance personnel, mishap investigators, and safety analysts. A review of five areas is needed to produce the Web-based prototype: (1) client/server architectures, (2) database management systems, (3) Web application design, (4) application coding, and (5) usability considerations for a Web/database tool. Collectively, these topics provided a foundation to develop an effective and user-friendly prototype, referred to as HFACS-ME Web. A usability study was conducted using potential end-users. The participants were given both written procedures to navigate through the prototype and an exit survey. The results of the survey, including objective and subjective responses, indicate strong user support for the HFACS-ME Web prototype in concept and implementation and suggest that the training and analysis capability it provides may contribute to a reduction in maintenance errors and ultimately a decreased mishap rate.				
14. SUBJECT TERMS Aviation Mishaps, Maintenance Mishaps, Maintenance Error, Human Factors, Information Management System, Client/Server Systems, Internet Technology, Web Design, Database Management Systems.				15. NUMBER OF PAGES 316
				16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified		20. LIMITATION OF ABSTRACT UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**WEB-BASED INFORMATION MANAGEMENT SYSTEM FOR THE
INVESTIGATION, REPORTING, AND ANALYSIS OF HUMAN ERROR IN
NAVAL AVIATION MAINTENANCE**

Anthony R. Boex
Commander, United States Navy
B.S., University of Washington, 1985

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

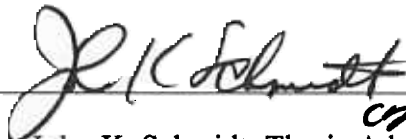
**NAVAL POSTGRADUATE SCHOOL
September 2001**

Author:

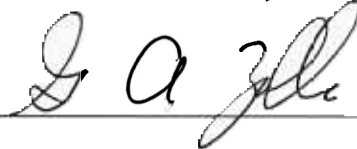


Anthony R. Boex

Approved by:

 *commcusa*

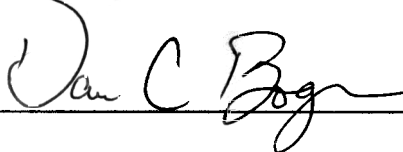
John K. Schmidt, Thesis Advisor



George A. Zolla, Associate Advisor



Robert C. Figlock, Associate Advisor



Dan Boger, Chairman
Information Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The purpose of this thesis is to examine the development of a Web application to display, analyze, and produce reports of human error involvement and patterns in aviation maintenance mishaps. The Human Factors Analysis and Classification System-Maintenance Extension (HFACS-ME) taxonomy, a framework for classifying and analyzing the presence of maintenance errors that lead to mishaps, is the foundation of this tool. The target audience for this tool includes safety and maintenance personnel, mishap investigators, and safety analysts. A review of five areas is needed to produce the Web-based prototype: (1) client/server architectures, (2) database management systems, (3) Web application design, (4) application coding, and (5) usability considerations for a Web/database tool. Collectively, these topics provided a foundation to develop an effective and user-friendly prototype, referred to as HFACS-ME Web. A usability study was conducted using potential end-users. The participants were given both written procedures to navigate through the prototype and an exit survey. The results of the survey, including objective and subjective responses, indicate strong user support for the HFACS-ME Web prototype in concept and implementation and suggest that the training and analysis capability it provides may contribute to a reduction in maintenance errors and ultimately a decreased mishap rate.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION	1
A. OVERVIEW.....	1
B. PURPOSE	6
C. PROBLEM STATEMENT	7
D. SCOPE AND LIMITATIONS	7
E. DEFINITIONS	8
F. CHAPTER ORGANIZATION	10
II. TWO-TIER VERSUS THREE-TIER CLIENT/SERVER ARCHITECTURE..	11
A. OVERVIEW.....	11
B. TWO-TIER ARCHITECTURE	11
C. THREE-TIER ARCHITECTURE	12
D. ARCHITECTURE COMPARISON	13
III. THREE TIER CLIENT/SERVER COMPONENTS	27
A. OVERVIEW.....	27
B. WEB COMMUNICATIONS PROTOCOLS.....	27
C. WEB BROWSER, AKA THE HTML CLIENT	28
D. WEB SERVER.....	32
E. DATABASE SERVER	37
F. SECURITY ISSUES	45
IV. METHODS.....	49
A. RESEARCH APPROACH.....	49
B. HUMAN FACTORS AND ANALYSIS CLASSIFICATION SYSTEM– MAINTENANCE EXTENSION (HFACS–ME) WEB.....	50
C. DATA COLLECTION.....	55
D. DATA TABULATION	57
E. DATA ANALYSIS	57
V. RESULTS.....	59
A. SAMPLE	59
B. DEMOGRAPHIC INFORMATION.....	60
C. PARTICIPANT SATISFACTION WITH THE HFACS–ME WEB PROTOTYPE.....	62
VI. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS	73
A. SUMMARY	73
B. CONCLUSIONS	73
C. RECOMMENDATIONS	74
APPENDIX A. HFACS–ME DATABASE SCHEMA	81
APPENDIX B. HFACS–ME WEBSITE SCREEN SHOTS.....	83
APPENDIX C. HTML AND ASP SOURCE CODE.....	95
APPENDIX D. SQL SERVER 2000 CODE.....	179
A. SQL SERVER 2000 STORED PROCEDURES.....	179
B. SQL SERVER 2000 VIEWS	271
APPENDIX E. HFACS–ME WEBSITE USABILITY EVALUATION	279

APPENDIX F. HFACS–ME WEBSITE USABILITY EVALUATION EXIT	
SURVEY	287
LIST OF REFERENCES	291
INITIAL DISTRIBUTION LIST	295

LIST OF FIGURES

Figure 1. Naval Aviation Mishaps, FY 1951–1999 (From NSC, 1999).....	1
Figure 2. Two-Tier Architecture Diagram (After SYBEX, 2001).....	12
Figure 3. Three-Tier Architecture Diagram (After SYBEX, 2001).....	13
Figure 4. Comparison of ADO, OLE DB, and ODBC (From http://www.microsoft.com/data/archb.htm)	35
Figure 5. Factor Analysis Layout.....	53
Figure 6. Exit Survey, Part II, Statement One, Response Distribution.....	63
Figure 7. Exit Survey, Part II, Statement Two, Response Distribution	64
Figure 8. Exit Survey, Part II, Statement Three, Response Distribution	65
Figure 9. Exit Survey, Part II, Statement Four, Response Distribution	66
Figure 10. Exit Survey, Part II, Statement Five, Response Distribution	67
Figure A-1. HFACS–ME Database Schema	81
Figure B-1. HFACS–ME Web Homepage.....	83
Figure B-2. Data selection Page.....	84
Figure B-3. Mishap Query Results Page.....	85
Figure B-4. Mishap Expanded Details Page	86
Figure B-5. Mishap Factor Analysis Results Page.....	87
Figure B-6. Graph Axis Selection Page	88
Figure B-7. Graph Data Table.....	89
Figure B-8. Graph Results Page.....	90
Figure B-9. HFACS–ME Taxonomy Legend	91
Figure B-10. Report Selection Page.....	92
Figure B-11. Report Results Page	93
Figure B-12. HFACS–ME Contact Page	94

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. The levels of HTML support.(From SYBEX, 2001)	30
Table 2. Participant Computer Configuration for Usability Study	59
Table 3. Participant Maintenance Affiliation (n=13).....	60
Table 4. Number of Hours Per Day Participants Use Specific Software Applications (n=12).....	61
Table 5. Participant Web Browser Usage Distribution (n=12).....	61
Table 6. Normal Operating System of the Participants (n=12).....	62

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS

ACL	Access Control List
ADO	ActiveX Data Object
AGM	Aircraft-Ground Mishap
AKA	Also Known As
API	Application Program Interface
ARPA	Advanced Research Project Agency
ASO	Aviation Safety Officer
ASP	Active Server Page
DBMS	Database Management Systems
DHTML	Dynamic HTML
DOD	Department of Defense
DON	Department of the Navy
FAA	Federal Aviation Administration
FM	Flight Mishap
FOQA	Flight Operational Quality Assurance
FRM	Flight-Related Mishap
FY	Fiscal Year
GB	Gigabyte
GUI	Graphical User Interface
HFACS	Human Factors Analysis and Classification System
HFACS–ME	Human Factors Analysis and Classification System–Maintenance Extension
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
IE	Internet Explorer
IETF	Internet Engineering Task Force
IIS	Internet Information Server
IP	Internet Protocol
IT-21	Information Technology for the 21st Century
JSP	Java Server Pages
MB	Megabyte
ME	Maintenance Extension (see HFACS–ME)
MEDA	Maintenance Error Decision Aid
MEIMS	Maintenance Extension Information Management System
MS	Microsoft
MSDE	Microsoft Data Engine
NASA	National Aeronautics & Space Administration
NMCI	Navy–Marine Corp Intranet
NSC	Naval Safety Center
NPS	Naval Postgraduate School
NTFS	NT File System
NTSB	National Transportation Safety Board
ODBC	Open Database Connectivity

PHP	Hypertext Preprocessor
QMB	Quality Management Board
RDBMS	Relational Database Management System
SIMS	Safety Information Management System
SQL	Structured Query Language
SSL	Secure Socket Layer
TCP	Transport Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
TEAM	Tools for Error Analysis in Maintenance
T-SQL	Transact SQL
URL	Universal Resource Locator
USN	United States Navy
VBScript	Visual Basic Script
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

ACKNOWLEDGEMENT

The author would like to acknowledge and express gratitude to Captain (sel) John Schmidt, Professor George Zolla, and Professor Robert Figlock for their tremendous assistance, prudent direction, sincere dedication, and good humor throughout the conduct of this study.

THIS PAGE INTENTIONALLY LEFT BLANK

DEDICATION

I would like to take this opportunity to express my sincere thanks to those people who have encouraged me throughout this project.

To my parents, Harold A. Boex and Mary Helen Boex, regardless of the challenges that I have faced in life, I have always received your love, encouragement and support. I am eternally grateful, for without you I would not have achieved nearly as much.

My love and heartfelt thanks go to my three sons Braden, Colin, and Sutton, and my amazing wife Cindy (who is pregnant with our little girl). You have always supported me and this endeavor has been no exception. You have allowed me to take occasional “leaves of absence” from family life in order that I may work towards the completion of this thesis and to pursue the career that I love. Thank you for your unwavering support, encouragement and understanding. I love you all.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. OVERVIEW

The Department of the Navy (DON) is in the midst of a major transformation as it attempts to come to terms with the demands of maintaining operational readiness in the face of diminishing budgets and reduced manning. Diminishing operating and procurement budgets mean that Naval Aviation is for the most part “making do” with existing aircraft. This means that squadrons are looking to maximize aircraft availability by minimizing errors that result in aircraft loss or damage.

Naval Aviation achieved great success over the years in reducing its Class A Flight Mishap (FM) rate (see Figure 1). Despite this achievement, the proportion of mishaps attributed to human error has remained relatively constant (Nutwell & Sherman, 1997). In 1996, senior Naval leadership established a Human Factors Quality Management Board (QMB) with an objective to reduce human error involvement in Naval Aviation Class A FMs by 50 percent by fiscal year (FY) 2000 (QMB, 1997). Because aircrew error is a contributing factor in 60 percent of Class A FMs, it became the initial focus of the QMB.

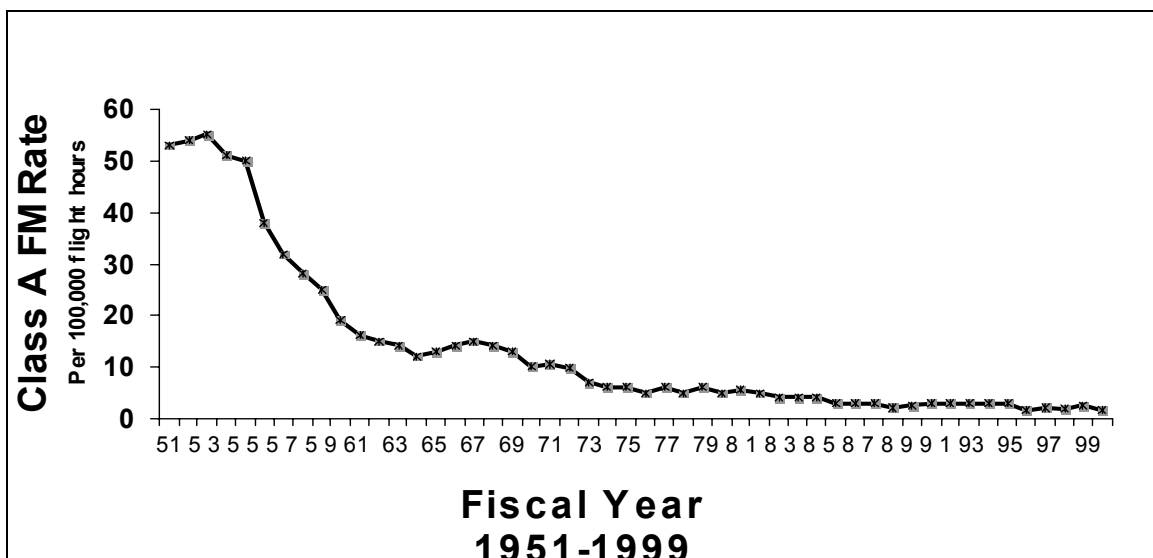


Figure 1. Naval Aviation Mishaps, FY 1951–1999 (From NSC, 1999)

As part of this project, the Naval Safety Center (NSC) developed the Human Factors Analysis and Classification System (HFACS) to capture aircrew errors and contributing factors in Naval Aviation mishaps. The goal of HFACS is to identify areas for potential intervention by fully describing factors that are precursors to mishaps. The resulting taxonomy identifies both active failures and latent conditions within four categories: (1) unsafe acts, (2) pre-conditions for unsafe acts, (3) unsafe supervision, and (4) organizational influences. NSC adopted HFACS to analyze aircrew error in Naval Aviation mishaps and targeting areas for intervention. (DON, 2001) Although Naval Aviation had its lowest Class A FM rate in FY 1999, the QMB's 50 percent reduction goal achieved (NSC, 1999). Consequently, the QMB expanded its focus to include the reduction of maintenance error in aviation mishaps.

Although HFACS facilitated identification of mishap factors, its restricted focus on only aircrew errors limited its utility. Over the past decade, nearly one in five Naval Aviation Class A FMs was partially attributable to maintenance error (Schmorrow, 1998). The present operating environment underscores the need to address maintenance error and its causes. HFACS was adapted and adopted to cover maintenance operations, and the extension was successfully used to examine major maintenance mishaps (Schmidt, Schmorrow, & Hardee, 1997), all mishaps (Schmidt, Schmorrow, & Figlock, 2000), and incidents and injury (Schmidt, Figlock, & Teeters, 1999) data. Naval Aviation included the Maintenance Extension of HFACS (HFACS-ME) in the most recent revision of the Naval Aviation Safety Program Instruction (DON, 2000). HFACS-ME contains four categories: (1) Management Conditions, (2) Working Conditions, (3) Maintainer Conditions, and (4) Maintainer Acts.

A review of 470 Naval Aviation mishaps by Schmorrow (1998) determined HFACS-ME was an effective classification tool for determining trends in maintenance mishaps. Further analysis by Fry (2000) using 595 mishaps validated Schmorrow's model and provided insights into the validity of maintenance error forecasting. Using the model, the predicted impact of maintenance mishaps on future operational readiness and mission capabilities can be better understood facilitating decisions concerning allocation of resources.

Unfortunately, human error has not been served well by conventional accident investigation methods (Marx, 1998). Normally they end once error is identified without trying to understand why it occurred. An easy-to-use tool that enables a user to query the stored data and perform analysis across multiple dimensions could more effectively address and identify patterns of maintenance error using HFACS–ME. The target audience for such a tool would include safety personnel (e.g., data entry and retrieval by organizational safety officers, other safety and training personnel, maintenance officers, maintenance supervisors, etc.), mishap investigators—for data retrieval (e.g., Aircraft Mishap Board members, squadron safety officers, etc.), and analysts (e.g., from the Naval Safety Center, training organizations, maintenance/safety documentation managers, and command/wing safety officers, etc.).

Boeing Aircraft Corporation took the view that human error is inevitable, and as such, they sought to enhance their ability to manage error in order to mitigate its consequences and learn what systemic factors contribute to its occurrence. (Graeber, 1999). In 1994 Boeing developed an event-driven tool to reduce maintenance related accidents by assisting investigators in the identification of accident contributing factors and recommendations for corrective actions. The Maintenance Error Decision Aid or “MEDA” (Hibit & Marx, 1994) supports human-centered error investigation in an attempt to encourage users to change their paradigm about maintenance error. MEDA is based on process improvement, and discourages the practice of simply punishing the person who commits the error. Investigators establish contributing factors in an event and make recommendations for process improvements. Once improvements are made, this information is provided to affected employees. Organizations using MEDA cite success in the form of reductions in maintenance related incidents, improved maintenance practices, however, Marx (1998) notes that MEDA and human factors based investigation methods in general is not being widely used. Of 92 commercial carriers using MEDA, only 6 are in the United States.

Both Galaxy Scientific and BF Goodrich have created software applications for MEDA to transform it from a pencil and paper collection method to the information age. Galaxy developed “TEAM”—Tools for Error Analysis in Maintenance (www.galaxyscientific.com, 2000) and BF Goodrich (1997) followed with a hybrid

system that incorporates MEDA and later another system called Aurora (www.hfskyway.faa.gov, 1999). These applications allow the user to collect, organize, analyze, and report data through an interactive graphical user interface system. Users are able to enter new or update existing error data, create reports (e.g., MEDA forms, contributing factors/error summaries, and audit information/checklists, etc.), and update information on corrective actions being taken.

The Flight Operational Quality Assurance (FOQA) program is a voluntary Federal Aviation Administration (FAA) sanctioned effort to share aggregated safety data from flight data recorders, across commercial airline carriers (www.faa.gov, 2000). The intent is to provide a means to examine industry wide trends and use the derived information to enhance personnel training, operational procedures, maintenance and engineering, air traffic control, and airport safety. Presently, 230 total aircraft consisting of 13 aircraft types are electronically collecting/sharing FOQA data (Reuters, 2000). An impetus for sharing under the banner of safety is that shared FOQA data is not used for enforcement purposes except under egregious circumstances.

The level of cooperation in FOQA has not been successfully extended to the hangar bay and flight line in terms of sharing maintenance error and incident data. The FAA and NTSB both concur that this is an essential part of the overall safety equation for increasing commercial aviation safety. One major problem standing in the way is having a common process/taxonomy for capturing, recording, and archiving accident/incident/error data for aggregate and trend analysis.

Boeing's MEDA, Galaxy's TEAM tool, and BF Goodrich's software tool all attempt to achieve a vehicle for not only capturing mishap information, but also for sharing data across the industry. Unfortunately, although used by some of Boeing's customers (e.g., BF Goodrich in its re-work facility, etc.), MEDA has not been adopted as an industry standard (Marx, 1998). Consequently, a need exists to develop a tool encompassing accident data collection, organization, analysis, and reporting.

Previous research work focused on the feasibility of the HFACS–ME as a taxonomy framework for the investigation, collection, and analysis of maintenance related mishap data. Fry (2000) developed the Maintenance Error Information Management System (MEIMS)—a software tool to facilitate data collection and analysis

of maintenance mishaps using the HFACS–ME taxonomy. Wood (2000) further refined Fry’s rudimentary MEIMS tool and developed it into a working prototype for Naval Fleet testing and evaluation. A usability study of the prototype MEIMS tool determined that it could be an effective tool, not only in determining trends, but providing information for mishap prevention efforts. Wood identified the need for MEIMS to incorporate improved HFACS–ME definitions, improved user interface, and simplified data entry procedures. McCracken (2000), using an improved version of MEIMS and a newly developed User’s Guide, conducted further usability testing with Aviation Safety Officer trainees. He concluded that users of MEIMS, when properly trained, could utilize the program as an effective tool in mishap prevention planning and training.

One significant drawback to the current MEIMS tool is that it runs as a stand-alone application with the underlying database stored locally on the computer running the MEIMS application. While this application provides for responsive data queries, analysis, report generation and local data entry, it has several shortcomings, including issues relating to data currency, application/database deployment and upgrade, and platform compatibility. During the course of this study, Nelson (2001) and Flanders and Tufts (2001) developed an advanced MEIMS tool that partially addressed the latter two issues and added an investigation module to facilitate mishap data entry.

In the current implementation of the NSC database, authorized personnel wishing to query the database must submit query requests via email or phone to technicians who manually key in the search criteria. After the query is executed the results are forwarded to the requestor via email or hardcopy. This process may take several hours to weeks, depending upon the query backlog and the requestor’s location. By developing a web-based prototype, the end user will be able to easily access the HFACS–ME database in a fraction of the time and obtain up-to-date information that can then be used in training, hazard identification, and trend analysis to prevent possible future incidents. The resulting application would take advantage of a centrally managed and secure database while providing the ability to make the information available to the greatest number of authorized users.

The central database also addresses three key factors that inhibit efficiency in knowledge management. First, storing all HFACS–ME data in a central database

prevents *incompleteness of information*. This ensures that the information presented to the user represents the "whole" picture and is not skewed by sampling error. Second, the ability of the end-user to view data that cuts across aircraft type boundaries and allows users in one aviation community to gain insight into maintenance errors affecting other communities. Under the current mishap message distribution system, messages are addressed to specific aircraft type communities, whereas maintenance error information may be applicable to a much broader audience. The end user's ability to aggregate data at various levels (e.g., aircraft type, mission category, or all Naval Aviation, etc.) prevents the *asymmetry of knowledge* perpetuated under the current system. Finally, people usually get knowledge from their organizational neighbors. This *localness of knowledge* is affected not only by the physical proximity to the knowledge source, but the ease of access and the trust extended to the source. The web-based interface addresses the proximity and access issues directly and database administration and security aspects influence the trust issue. (Davenport, 1998)

The future growth of web-based capabilities provided by the Navy–Marine Corps Intranet (NMCI) and Information Technology for the 21st Century (IT–21) infrastructures will provide the logical path for NSC to provide access to the Aviation Safety. This study also explored the possibility of incorporating the web-based tool into a civilian variant, which would presumably use the Internet as the logical communication medium to share safety related data across the industry. Currently, the HFACS–ME Web and MEIMS share a common HFACS–ME database and are designed to accommodate either military or civilian data. The biggest challenge will be the political and legal frameworks needed to encourage honest reporting when human error occurs (Graeber, 1999). The usefulness of the web-based HFACS–ME database will depend on whether the system adds value to the underlying mishap data, and ultimately, whether the end user gains knowledge leading to effective intervention and mishap prevention.

B. PURPOSE

The intent of this study is to develop and evaluate a Web-based HFACS–ME safety information management system that will facilitate the organization, query, analysis, and reporting of maintenance errors and contributing factors that lead to Naval

Aviation mishaps, equipment damage, and personnel injury. The goal of this project is to provide a tool that will enable safety, maintenance, and management personnel to query and analyze maintenance mishap data, regardless of geographic location or time of day. With the knowledge gained from this analysis, the user may be able to better understand and prevent errors, which may lead to personal injury, equipment damage and loss of valuable resources.

C. PROBLEM STATEMENT

In order to continue to reduce Naval Aviation mishaps there needs to be an understanding that all mishaps are not caused solely by aircrew error. The analysis of maintenance mishaps offers an increased opportunity to reduce target mishaps and enhance readiness. The HFACS–ME taxonomy was adapted to classify causal factors that contribute to maintenance mishaps. A modern database tool is essential in more effectively addressing and identifying patterns of human error using HFACS–ME. However, there is no such tool available today.

This thesis investigates the following questions: (1) How can a web-based tool be used to query the NSC aviation safety database of aircraft mishaps? (2) Will aviation safety and maintenance personnel effectively use a web-based tool to query and analyze aviation maintenance error data? (3) Will the web-based implementation make the aviation safety database easily accessible to authorized users, regardless of the user's location? (4) Will Active Server Pages (ASP), Hypertext Markup Language (HTML) pages, and Structured Query Language (SQL) provide required functionality, security, data integrity and display capabilities? (5) Can this tool be converted for commercial airline use?

D. SCOPE AND LIMITATIONS

Naval Fleet personnel, primarily Aviation Safety Officers, evaluated the prototype HFACS–ME web application. Based upon the Naval Aviation safety data it contains, the HFACS–ME Web prototype is designed for use by Naval Aviation squadrons, maintenance organizations, and support organizations. The Federal Aviation Administration (FAA) and the National Aeronautics & Space Administration (NASA)

have provided funding in support of related research in anticipation of crossover use by other military branches and civilian airlines. The data contained in the HFACS–ME database was manually transcribed from reports of actual Naval Aviation mishaps that occurred between 1990 and 2000. Only maintenance related mishaps caused by human error were included. Due to the anonymous nature of the Usability Study and Exit Survey conducted at the conclusion of this project, there is no means to probe deeper into a participant comment or concern about the HFACS–ME Web prototype nor is there a means to judge the motivation of the respondents.

E. DEFINITIONS

This study uses the following definitions:

Aircraft Mishap Board. Group of officers appointed to investigate and report on an aviation mishap (DON, 2001).

Aviation mishap rate. Number of aviation mishaps per 100,000 flight hours (DON, 2001).

Aviation Safety Officer. Principal advisor to Naval Aviation squadron commanding officers on all aviation safety matters (DON, 2001)

HFACS. Human Factors Analysis and Classification System. System designed to help analyze Naval Aviation mishaps focusing on aircrew error (DON, 2001).

HFACS–ME. Human Factors Analysis and Classification System–Maintenance Extension. HFACS adaptation to classify causal factors that contribute to maintenance mishaps (Schmidt, 1996).

HFACS–ME Web. Human Factors Analysis and Classification System–Maintenance Extension Web Application. Prototype tool developed for this thesis.

HFQMB. Human Factors Quality Management Board. Established by Naval Aviation senior leadership to reduce human error involvement in Naval Aviation Class A flight mishaps (QMB, 1997).

Mishap. A Naval mishap is an unplanned event or series of events directly involving naval aircraft, which result in \$20,000 or greater cumulative damage to naval aircraft, other aircraft, property, or personnel injury (DON, 2001).

Mishap Categories. Naval aircraft mishap categories are defined below (DON, 2001):

Flight Mishap (FM). A mishap in which there is \$20,000 or greater DOD aircraft damage or loss of a DOD aircraft, and intent for flight for DOD aircraft existed at the time of the mishap. Other property damage, injury, or death may or may not have occurred.

Flight Related Mishap (FRM). A mishap in which there was less than \$20,000 DOD aircraft damage, and intent for flight (for DOD aircraft) existed at the time of the mishap, and \$20,000 or more total damage or a defined injury or death occurred.

Aircraft Ground Mishap (AGM). A mishap in which no intent for flight existed at the time of the mishap and DOD aircraft loss, or \$20,000 or more aircraft damage, and/or property damage, or a defined injury or death occurred.

Mishap Severity Class. Mishap severity classes are based on personnel injury and property damage (DON, 2001):

Class A. A mishap in which the total cost of property damage (including all aircraft damage) is \$1,000,000 or greater; or a naval aircraft is destroyed or missing; or any fatality or permanent total disability occurs with direct involvement of naval aircraft.

Class B. A mishap in which the total cost of property damage (including all aircraft damage) is \$200,000 or more, but less than \$1,000,000 and/or a permanent partial disability, and/or the hospitalization of five or more personnel.

Class C. A mishap in which the total cost of property damage (including all aircraft damage) is \$20,000 or more but less than \$200,000 and/or injury results in five or more lost workdays.

Naval Aircraft. Refers to US Navy, US Naval Reserve, US Marine Corps, and US Marine Corps aircraft.

OPNAVINST 3750.6: The Naval Aviation Safety Program. US Navy instruction outlining Naval Aviation's safety program. Revision R-2001, (DON, 2001).

F. CHAPTER ORGANIZATION

Chapter II contains a literature review of two-tier and three-tier client/server architectures and compares the strengths and limitations of each. Chapter III discusses the planning and development of a Web-based application to allow an authorized user to query, display, analyze and report human error involvement and look for patterns in aviation maintenance mishaps. Chapter IV details implementation of the HFACS–ME Web prototype and the methods used in this study. Chapter V discusses final prototype design and the results of a preliminary Usability Evaluation. Lastly, Chapter VI contains Summary, Conclusions, and Recommendations.

II. TWO-TIER VERSUS THREE-TIER CLIENT/SERVER ARCHITECTURE

A. OVERVIEW

This study examines the development of a Web application to display, analyze, and produce reports of human error involvement and patterns in aviation maintenance mishaps. The literature review includes textbooks, research articles, and masters theses pertaining to: (1) client/server architectures, (2) database management systems (DBMS), (3) Web application design and development including database interface, (4) application coding, and (5) usability considerations for an effective Web/database tool. Collectively, these information sources provide a foundation to develop an effective and user-friendly prototype maintenance error analysis and reporting tool, hereafter referred to as HFACS-ME Web.

Historically, the most common method for users to access information in a database is with a two-tiered client/server architecture. This architecture presents an easy-to-use interface for the user (normally by supporting a Graphical User Interface (GUI)), circumventing the need for the casual database user to learn complex database manipulation languages, such as Structured Query Language (SQL). In many cases, these applications could be replaced with a three-tier client/server architecture using a browser, Web server and a database server. There are substantial advantages to the Web approach, but there are serious obstacles that must be overcome before the Web technology can completely replace the need for client/server applications. Two-tier and three-tier applications may appear essentially the same to the end user, but there are some very fundamental differences in how they work that have an important impact on their functionality.

B. TWO-TIER ARCHITECTURE

In a two-tier application, the application program runs on the end user's computer (the client) and communicates with the server (i.e., database server, etc.) through a network or modem connection. In a database client/server application, the client can pass SQL statements through a Transmission Control Protocol/Internet Protocol (TCP/IP) connection and if necessary, a database specific protocol (e.g., SQLNet for ORACLE,

etc.), to the database. The results are returned to the client machine via the same middleware protocols and is displayed to the user. This scenario is depicted in Figure 2.

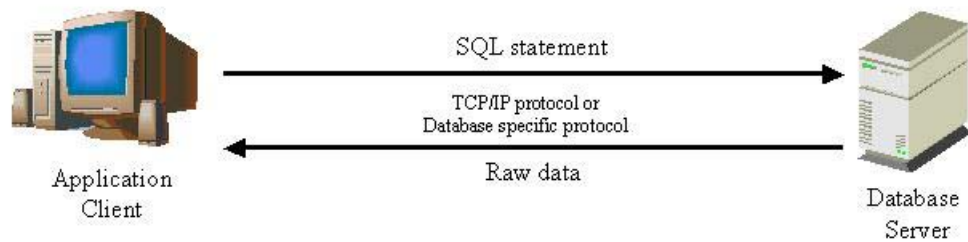


Figure 2. Two-Tier Architecture Diagram (After SYBEX, 2001)

C. THREE-TIER ARCHITECTURE

The three-tier interface operates differently than two-tier applications. Similar to two-tier applications, three-tier applications also display information in a GUI. However, in a Web interface the GUI is provided by the Web browser that runs on the client machine, as opposed to the GUI being provided by the particular programming language used in a two-tier application. In this scenario, the Web server provides an additional layer between the client and the database server. The user specifies a Uniform Resource Locator (URL), which uniquely identifies a particular Web server to connect to and a Hypertext Markup Language (HTML) file to view or a program to run on that server. Between the Web browser and the Web Server, TCP/IP is used as the underlying communications protocol (for Internet communications) and the request for the web page is handled by the HTTP protocol which rides on top of TCP/IP. The Web server can then interpret the URL and service the HTTP Request. If the request requires database access, the request for data, normally a SQL statement, is sent to the database server either using TCP/IP, a proprietary database protocol, or an interprocess communication protocol called Named Pipes. Named Pipes is a protocol within the Windows NT (and UNIX) operating systems that allow processes running on the same machine communicate with one and other. This allows the database server to be configured to not accept TCP/IP communications thereby making it less vulnerable to exploitations from the network. Once the data request has been serviced, the Web server formats the retrieved data into HTML, and sends it to the client machine to be displayed in the browser. See Figure 3 for a diagram of this configuration.

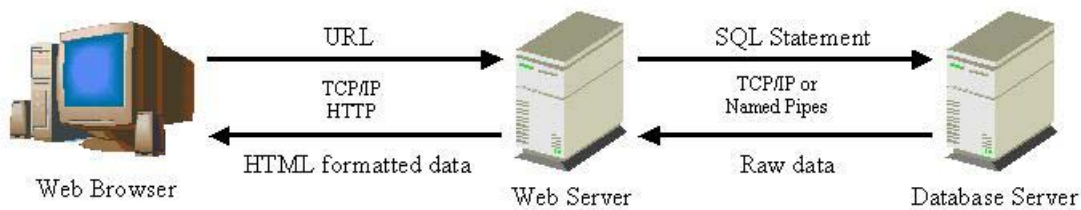


Figure 3. Three-Tier Architecture Diagram (After SYBEX, 2001)

D. ARCHITECTURE COMPARISON

As many application programmers, project managers, and other information system professionals have learned over the years, two-tier database applications are inherently difficult to build, deploy, and manage. Billions of dollars have been invested in client/server technology, and many tools and products have been developed in support of two-tier applications, but it remains a problematic field. The characteristics of Web technology make it possible to circumvent many of the problems associated with two-tier development. Although the advantages of the Web approach over two-tier architecture may seem overwhelming, there remain many obstacles for Web technology to overcome before it can completely replace two-tier technology.

1. Installation and Maintenance

One of the biggest problems that plagues two-tier applications is that of deployment and maintenance. Someone possessing administrator privileges on the machine must install the application program executables, along with various other system files that are necessary to run it, on each end user's machine. This usually requires that a complex installation routine and error handling procedures be written, which is then distributed to each of the users. It is often the case that end-users have different hardware and software configurations, which may cause conflicts with the client software as it is being installed and makes it necessary for the organization deploying the application to provide extensive technical support to its users. In addition, if a system is being developed for use by a client running on a different platform and/or utilizing different operating systems, the application must be ported to each of the intended platforms. Once the application has been deployed, any software changes require that the

system be re-installed on each user's machine. All of these factors make the administration of a two-tier application a very tedious process.

By using a Web application, these problems are avoided. The programs to retrieve and manipulate data are stored in one place: either on the Web server or in the database itself. This eliminates the need to install software on each end user's machine and makes software changes much easier. Furthermore, users running on different platforms can use the same programs. There is no need to write different versions for different platforms as Web browsers are available for a wide variety of platforms and HTML is machine independent (although presentation of HTML content on different browsers is rarely identical due to subtle differences in HTML Standard implementation). These advantages make Web technology a very desirable alternative to two-tier development.

2. Connectivity

Another problem that two-tier developers face is that of connectivity. Client/server applications have to deal with users connecting by modems, Internet connections, and local network connections as well. Most commercial Database Management Systems (DBMS) have their own network software, which require expensive site licenses for each user. Three-tier applications do not have these problems. The client simply needs to have Internet or modem access to the Web, and the application developers need only worry about the connectivity between the Web server and the database. Furthermore, because Web applications do not maintain a persistent connection to the database, the database server can handle more users at a time.

3. User Interface

One final benefit of the Web approach is that nearly all users, no matter what their level of sophistication, are generally comfortable using a Web browser. This is the part of the appeal of the World Wide Web (WWW) that has made it so popular. The fact that many people are already familiar with the use of a browser eases part of the learning curve associated with the deployment of any application. This will make it easier to train end-users on the use of the system. Like any application, a poorly written Web

application will be more difficult to use than a well written one, but the standardized features of a Web browser provide users with a familiar framework.

4. State Maintenance

Probably the biggest obstacle that Web developers have encountered is the stateless nature of the HTTP server. Whereas two-tier applications have a program running throughout the various screens of the application, Web applications consist of completely independent pages and have no persistent program memory to save user state information. Every connection is negotiated from scratch, not just at the page level, but for every element on the page. Each HTTP exchange is a completely independent event. Therefore, information entered into one HTML form must be saved by the associated server application in a place where it can be accessed by subsequent programs in a conversation. Basically, there are only two places to store state information—on the client or on the server. This is not to say that the information has to be saved on the Web server itself, but on the server side of the browser-server relationship. As a simple example of why this is a problem, suppose a corporate application requires the user to select from a list of customers on the initial screen in order to display various information about that customer on the next screen. In the two-tier application, the customer name and possibly an identifier would simply be stored as variables in memory and that information would be accessible in subsequent screens. In a Web application, however, the information must be explicitly passed on to the next page, as each Web page has no knowledge of the of the previous pages. In this example, there is only a small amount of state information to be passed to the next page, but even in a relatively simple Web application the amount of data that must be stored can quickly escalate to unmanageable levels. Six methods currently are employed by developers as workarounds to this problem: (1) Session variables, (2) Cookies, (3) QueryString variables, (4) Hidden form variables, (5) files, or (6) database tables.

Examples of each method will use the following scenario: a database table exists which contains demographic information about registered users (e.g., UserIDs, sign-ons, names, etc). When a user signs on, the sign-on value entered by the user is used to look up information in the database. Now assume the application needs access to the user's

demographic data for each page in the application. To be able to retrieve the user's information, the application needs to somehow make, and keep, an association between the user's browser and the information related to that browser.

a. Maintaining State With Cookies

Cookies are highly efficient from the server point of view because they push the burden of maintaining data onto the browser. You can write persistent or temporary (per-session) cookies, depending on your needs. However, if you rely on cookies there are six considerations to be considered:

(1) Cookies increase network traffic. Each time a cookie is stored on a browser, the browser subsequently sends the cookie name and value(s) back to the server for each request. Assume the maximum amount of information (approximately 4 KB) of information is stored in each cookies set. If there are 20 users on the site, it is unlikely to have a significant impact on network traffic; but if there are 2,000 users, each making requests 4 times per minute, network traffic has increased to 32 megabyte (MB) per minute. While that may not bring the network and server to a halt, the scalability of this method needs to be considered.

(2) Cookies increase the time required to service a request. The server must parse the cookie string and create the Request.Cookie collection for each request. Therefore, the more values stored in the cookie string, the longer it takes the server to parse the string, thus increasing the response time and reducing scalability.

(3) Cookies are machine-dependent. When a persistent cookie is written to a computer, the browser will return that cookie value the next time a user visits the application. But, if the person uses a different computer to access the application, the cookie value will not be there. Consider an individual who accesses a travel reservations site from work, makes a reservation, and then accesses the same site from home. If the site relies on cookies to maintain state, the user's reservation information will be on the work computer, but not on the home computer.

(4) Cookies are browser-dependent. The problem is actually worse than described in the previous paragraph. Cookies belong to a specific browser.

Suppose a user has two browsers. After accessing the travel application with Microsoft Internet Explorer™ (IE), the user later (that same day) returns to the site with Netscape Communicator™. The two browsers maintain cookies independently. That means that any cookie you write with IE is not sent with Netscape, and vice versa. Therefore, the application will not have access to the information stored in cookies for visits by other browsers, even if the machine and user are the same.

(5) Cookies may disappear. Perhaps worst of all, because browsers store persistent cookies in directories accessible to the users, those users can delete the files, or the machine can crash, or the user can rebuild the machine. In other words, critical data cannot be stored in cookies because you cannot rely on the information to be present in the future.

(6) Users control cookies. Finally, even with per-session cookies, there is a small chance that a person may start a Web application with cookies enabled, but turn cookies off during the session. That means the application may lose cookie values in the middle of the session—not an attractive scenario for critical applications.

The point of this example is not to suggest that cookies should not be used; it is that cookies are not necessarily the best place to store data. They are limited in size, are not guaranteed to be present between sessions (or even during a session), and not all browsers accept them. Nevertheless, cookies are the method that Microsoft chose to associate a browser with server-side data,—they work most of the time. In spite of these drawbacks cookies do have some advantages. They can be created and read easily with both client and server-side code, they work in all modern browsers, and they can be encrypted for added security.

If only a small amount of information needs to be stored between requests and there is not much concern if the information is lost, cookies are an excellent way to maintain state because they do not take up any memory on the server. Permanent cookies (as they are called because they are not stored in volatile memory) can also be written to store data on the client's hard drive. Permanent cookies make it possible to store state between sessions. By default, the browser sends all cookies set by an application back to

that application, but there are a number of methods to specify cookie use in virtual directories and alternate domains.

The browser stores and transmits cookies as text files, so they need to be encrypted if the information is sensitive, such as sign-on and password information. Cookies can be manually encrypted or the browser can encrypt the data automatically by using the Secure attribute. This tells the browser to store cookies in encrypted form, and to send them only to sites that support Secure Sockets Layer (SSL) encryption.

b. Maintaining State With QueryString Variables

There has been a great deal of press coverage (mostly negative) about how unscrupulous Web sites are impinging on privacy by storing unwanted information on your computer in cookies. Therefore, some of the more paranoid individuals have improved their privacy by turning cookies off. For clients that will not accept cookies, Web applications cannot store anything in the Session object because the ASP engine will not find the SessionID cookie, and (if Sessions are enabled for the application) will try to create a new Session cookie for each request. Of course, cookies cannot be used to maintain state in this case either. Instead, information can be passed through the QueryString collection.

QueryString data is text appended to the URL. On the server, the data is received in the Request object as the Request.QueryString collection. As is typical with Web communications, the raw data is in the key=value form like cookies, form variables, etc. Ampersands separate the key-value pairs. A question mark separates the entire query string from the URL. The following URL example contains two QueryString values- LastName and FirstName:

`http://MyServer/MySite.com?LastName=Doe&FirstName=John`

Most browsers support up to about 1024 characters in QueryString data. The server parses the QueryString data into a collection, so it can be retrieved using keys or indexes. That makes it extremely easy to retrieve the data sent by the browser. QueryString parameters can be added to HTML and sent to the browser by concatenating

strings together to produce a valid URL. As subsequent pages are displayed the information needs to be carried forward. All that concatenation can become painful when you're trying to keep track of multiple variables. Especially, if trying to append enough information to keep track of 20 or 30 critical variables in an application!

There are some undesirable characteristics of using QueryString data to maintain state. First, the user (and other observers) can see the data because it appears in the address bar. Therefore, this method should never be used to send any private information unless encryption is used. Second, if QueryString data is used and the data contains spaces or other non-alphanumeric characters, the `Server.URLEncode` method must be applied to the query string before appending it to a URL to avoid errors.

Finally, because the information in the QueryString on the browser appears in the address field, users can change the QueryString data, which can lead to errors in the Web application. Therefore, one cannot always depend on the validity of the information in the QueryString variables

c. Maintaining State With Hidden Form Variables

Another way of maintaining state on the client is to use hidden form variables. To do this, a Web developer creates a `<form>` tag and inserts the hidden form state variable which needs to be passed back to the server. The form needs to be submitted by the client using the POST method so the browser returns the values to the server. This can be accomplished by a Submit button for each page, or by using client-side script to submit the form when a navigation event-such as a click on a link or button-occurs.

```
<form name="frmHidden" method="post">  
<input type="hidden" name="ID" value="2817">  
<input type="submit" value="Submit">  
</form>
```

When the user clicks the Submit button, the form will send the value `ID=2817` to the server. The Web application can retrieve the value using the `Request.Form` collection, and then use the value in a form on the next page. To submit the form from a link, use client-side script such as this:


```

<script language="javascript">
function doSubmit()
{
document.frmHidden.submit();
}
</script>
<form action="mypage.asp" name="frmHidden" method="post">
<input type="hidden" name="ID" value="2817">
<input type="submit" value="Submit">
</form>

```

When the user clicked on the following link, the javascript subroutine above would be called, which would submit the form to the target ASP page.

```

<a href="mypage.asp" onClick="doSubmit();">Click Here</a>

```

Then the ID value is passed to mypage.asp, and retrieved by the application using a call to Request.Form("ID"). The process of setting and retrieving the ID variable could continue on subsequent pages as long as the variable value was needed on the server to maintain state.

Hidden form variables are in some ways better than cookies or QueryString variables. There is no size restriction on hidden form variables, they are not visible in the browser's address field, and users cannot change them; but they are visible to users savvy enough to use the View Source feature of the browser and therefore should not be used for values you do not want the user to see (e.g., answers to test questions, etc.) unless you encrypt them.

d. Maintaining State With Session Variables

Session variables are the easiest way to maintain state and are not limited to a specific data type. Once the UserID is stored in a Session variable, then whenever the user makes a request, the value of the Session("UserID") variable is retrieved from server memory and the user's information are retrieved from the database, as seen in the example.

```
Set R = conn.execute("SELECT UserID FROM Users " & _  
"WHERE UserID=" & session("UserID"), ,adCmdText)
```

Unfortunately, retrieving database data is an expensive operation in terms of both time and server resources. It may not be efficient to perform a database lookup for each user request. If the application retrieved the data during the sign-on request it could store the entire row in Session variables at that time. That way, the application would not need to go back to the database for each subsequent request. Instead, a Session variable for each field in the table row could be created for that user. For example:

```
session("UserID") = rs("UserID").Value  
session("Usersignon") = rs("Usersignon").Value  
session("UserLastName") = rs("UserLastName").Value  
Session("UserFirstName") = rs("UserFirstName").Value  
session("UserEMail") = rs("UserEmail").Value
```

Now, the application can have access to the user data from anywhere in the application without returning to the database. In essence, it has cached the demographic data for this user in the Session object. This is absolutely the easiest way to cache data for an individual user, and probably the one used most often. There are only a couple of problems with it. Using Session variables forces the server to access a single object for each request. Therefore, the use of Session variables must force the server to serialize requests, at least during retrieval of the sub-dictionary containing the Session variables associated with that user's SessionID.

Sessions are a mixed blessing. They make development extremely easy, but Microsoft itself recommends that you should use another method besides Sessions to maintain state if you want to build Web sites that will scale to large numbers of users. Consider a Web farm where more than one server is used to service requests. If Session variables are used, some method is needed to route requests from one browser to the same server for each request. If such a method is not used (and the ASP engine does not currently provide one), the Session information either will not be available or will be different on each server.

Finally, although it may not always be an issue, developers need to be aware of the memory requirements when using Session variables. Storing a few small strings and numbers for each user will not use much memory, but storing entire record sets, even as arrays will use up large amounts of memory in a hurry. If the server runs out of RAM, it will begin spooling the data out to virtual memory, and that will definitely adversely affect the application's response time and scalability.

e. Maintaining State With Files

Although reading a file for each request may sound like a totally inefficient method for maintaining state, it is not as bad as it sounds. The Web server caches files—in fact, site administrators can set the number of files the server caches to suit the resources available on it. Because requests tend to come in clusters—a single user will use an application, then quit—so file-caching services will likely improve the speed of file reads and writes.

While the application still needs to rely on one of the previous forms of data caching to make the association between the browser and the file, any of the previously mentioned methods may be used. For example, a UserID may be saved as a cookie, placed in the QueryString, saved as a Session variable, or submitted it as a hidden form variable. The result is that the UserID value is obtained and that value is used to open a file, allowing additional user information to be retrieved. A significant advantage of this method is the ability to associate a reasonably large amount of data with a single pointer value. Unfortunately, as with any promising solution there are drawbacks. For example, the files become harder to manage when the number of users grows. There is also the problem of managing obsolete files and duplicates. If a user visits the site once, how long should the data be maintained before it is deleted? For some sites, the data may need to be maintained. For others, it may only be needed during the current session. You could delete any user files who's timestamp is older than a predetermined period; however, if users re-visit the site within that period, you would need to update the file timestamp by writing data to the file so it does not get deleted. As you can see, it would be nice to have a more robust way to associate data with a pointer stored on the client. And you do have access to that method—with a database.

f. Maintaining State in a Database

Databases are probably the most scalable way to maintain state. Storing state in a database is definitely slower when the application has only a few users, but as the number of users and the size of the data increases, using a database to store state quickly outstrips using Session variables. It is also easier to script than using QueryString or Form variables, especially when you have a large application where you must store many more state variables.

Another major advantage of storing state in a database is that state can be maintained not only during a session, but also between sessions. The only other way to maintain state between sessions is to use persistent cookies or files. Cookies maintain state on the client, whereas state stored in a database resides on the server. Maintaining cross-session state on the server is better, as people frequently change computers, delete cookie files, or sign on to their computers using other sign-ons, so client-side state is less certain.

To store state in a database, the appropriate table must be properly set up. Meaning that the application must be able to identify the data row or rows that belong to an individual or session. This is accomplished by setting a single identifying cookie that acts as a primary key to the table, then retrieving the data using that cookie value when each request starts. For secured sites, the application can create and set its own cookie value, such as a unique user's ID or sign-on. For cross-session state maintenance, each user will need to sign on with a login and password. For single-session state maintenance, the application can use the SessionID cookie as long as it is not necessary to store state for an individual user, just an individual session.

To avoid a database growing unchecked, a scheme to delete or archive obsolete state data (data you've stored for individuals that never return to your site) must be implemented. For many public sites, this is the bulk of the data collected; for internal sites, it may occur only when an employee leaves the company. If persistent cookies are used to connect people to their data, a means for a user whose cookie has been lost to reconnect to his or her data would also be desirable. That's why many sites ask you to

provide special personal information so they can ask you about it if you lose or delete the site cookie.

g. Summing Up State Maintenance Options

If you only need to store small amounts of data (less than 4 KB), and especially if you do not need cross-session data, use cookies. Cookies work well and you will be able to develop the application quickly. If you do not expect to ever have to scale the site beyond a single server, use Session variables. If you need to keep data across sessions, or if the data you need to maintain is large, you can use a file or database to persist the data. If your client does not have cookies enabled, use hidden form or QueryString variables to store a pointer value on the client. Use the pointer value to retrieve associated information from files or database tables on the server. You can also use these methods if your server does not have sessions enabled. If you need high scalability, use database tables in combination with any other method that can store a pointer to the data. Remember that the less data you must retrieve for any request, the faster and more scalable your application becomes. Finally, you can use more than one method at the same time-database tables, files, and cookies to store information between sessions, as well as Session variables, QueryString variables, hidden form variables and cookies to store information during a session. Perhaps future versions of the HTTP protocol will solve the problems caused by statelessness, but in the current state of Web technology, it is up to the programmer to maintain the user's state.

5. Connectivity Requirement

Another problem with three-tier development that follows from the stateless nature of the Web (more specifically the HTTP Server protocol) is that HTTP applications are also connectionless. In a two-tier application, the user connects to the database and remains connected throughout the session. Web applications, however, must establish a new database connection with each new page.

6. User Environment

Yet another difficulty with three-tier interfaces is that the programmer has less control over the user's environment. While a two-tier application developer has a great deal of control over the appearance and execution of the application, the appearance of a Web application is determined by the particular browser and platform that the end user is running. The same characteristics that provide greater flexibility and portability in Web applications also lead to a loss in control in the way the application looks and behaves. A Web page that looks nice on the developer's screen may have a very different appearance to an end user. This is a tradeoff, which must be considered in deciding whether to choose the Web approach over two-tier technology. Any Web application should be viewed on a variety of browsers and platforms before being deployed.

In addition to the lack of control over appearance, the Web developer also has less control over the flow of execution than in a client/server environment. The navigation capabilities of browsers allow users to go back to the previous screen at any time, start in the middle of the application, or go halfway through the application and then quit. The programmer must take all of these possibilities into consideration.

7. Security

There are also significant drawbacks to the Web approach in the area of security. Due to the Internet's open nature, security is a topic that must be given careful consideration when planning website and database design, implementation, and functionality. The measures put into place also have significant ramifications with regard to site performance and scalability. An attempt to cover the broad range of security issues related to the Internet is beyond the scope of this thesis but a discussion of issues relating to the implementation of the HFACS–ME Web prototype will be covered in Chapter 3.

8. Functionality

One final obstacle to three-tier development, and a substantial one, is that although the Web has some powerful capabilities, the current state of Web technology cannot do everything that a two-tier application can do. Some of the more complex functionality of two-tier applications can simply not be easily replicated in a Web

application without the use of client-side scripting and/or Dynamic HTML (DHTML). (Note that dynamic HTML refers to Web content that changes each time it is viewed depending on parameters, whereas, when capitalized, *Dynamic HTML* refers to new HTML extensions that will enable a Web page to react to user input without sending requests to the Web server. Microsoft and Netscape have submitted competing Dynamic HTML proposals to the World Wide Web Consortium (W3C), which is producing the final specification.). Therefore, until Web technology and the standards implemented by browsers include such capabilities, a three-tier application may not provide all the features of a two-tier application. However, for relatively simple applications, a three-tier interface is a viable option.

III. THREE TIER CLIENT/SERVER COMPONENTS

A. OVERVIEW

Before the Internet became the ubiquitous communication medium that it is today it was a project designed to connect scientists and researchers around the world. However, it had some problems. There were far too many access tools, and none of them were very user-friendly back in the days of text screens and command keys. The Internet did not stand out having a snappy graphical interface. As access tools gained more functionality and new standards were put into place, HTTP and HTML quickly took over and contributed to the mass appeal of the Internet today.

The following sections discuss the components that make up the three-tier client/server architecture and address design and implementations considerations and challenges that were addressed when developing the HFACS–ME Web application.

B. WEB COMMUNICATIONS PROTOCOLS

There are two main protocols for using the Web: Transmission Control Protocol/Internet Protocol (TCP/IP) and HTTP.

TCP/IP is a result of research funded by the U.S. government's Advanced Research Project Agency (ARPA). The original intent was to allow researchers from around the world to communicate ideas and files so they could advance research projects more easily. This network of research computers, which used TCP/IP as its protocol, eventually became the Internet.

The Transmission Control Protocol (TCP) determines how to divide the information into packets, and the Internet Protocol (IP) transports these packets. The Internet Protocol does not guarantee that the packets will be received in the order they were sent; it is up to the TCP to reassemble the packets in the correct order. TCP/IP addresses are 16-digit numbers separated into four sections (known as *octets*), such as 255.14.130.12; the number in each section cannot be higher than 255. Each section identifies, to some degree, the location of the recipient of the packets. The first two sections are generally considered the network address, the third section is considered the subnet mask, and the fourth section is the address of the physical machine.

HTTP is a protocol that sits on top of TCP/IP. The protocol translates requests from Web pages into requests over the network; it then takes browser requests in the format of a method. The HTTP methods are: (1) GET, (2) PUT, (3) POST, and (4) DELETE. The GET method requests a file from the Web server. It is simply a method for linking from one page to another, not for handling any kind of form that you fill out. The PUT method is rarely used because it allows a request to create a new file or append to the file if it already exists. Forms to pass parameters to the Web server use the POST method. The DELETE method can be used to delete a file from the Web server. For security reasons, Web servers generally do not permit PUT and DELETE methods.

The HTTP model of Web browsing fits neatly into the client-server model. The Web browser, acting as the client, makes requests for pages to the Web server. The Web server fulfils these requests by responding with a Web page. Generally, each request is sent as a separate and unique connection

All Web pages are sent as text files. Along with the request, the browser sends a header of information about itself, including what types of files it can handle. The server then uses this information to determine whether it should send the page. For the server to respond to the browser, the server sends a status code along with the page. If the page is sent, the status code is usually a success code; if the page is not sent, the status code is some type of error code.

HTTP is the standard of the Internet Engineering Task Force (IETF). The current release version is HTTP/1.1 (IETF draft standard RFC 2616). HTTP/1.1 is designed to bring about significant performance gains through support for persistent connections and pipelining for much more efficient use of TCP networks, continued extension of the caching model and support for multi-homing servers (allowing a single web server to serve multiple web sites each with their own unique address). Improvements in HTTP/1.1 are limited due to the requirement for backwards compatibility with HTTP/1.0. (ITEF, 1999)

C. WEB BROWSER, AKA THE HTML CLIENT

The Web browser provides the GUI, which displays information and incorporates user interface components. These components may be part of the browser application

itself (i.e., address bar, Back button, etc.) or those controls displayed as part of the Web application (i.e., drop-down lists, hyperlinks, Submit buttons, etc.). To display information returned from a Web server the browser needs to interpret the text data received and render it appropriately.

1. HTML Standards and Compatibility

The language used to accomplish this is HyperText Markup Language or HTML, which is a generic markup language for representing documents. HTML describes the relationship between a document's content and its structure and allows document-based information to be shared and re-used across applications and computer platforms in an open, vendor-neutral format. HTML has two essential features—hypertext and universality. Hypertext means that you can create a link in a Web page that leads the visitor to any other Web page or to practically anything else on the Internet. It means that the information on the Web can be accessed from many different directions. Universality means that because HTML documents are saved as ASCII or Text Only files, virtually any computer can read the Web page. It does not matter if the page is displayed on a Mac or a Windows machine, a UNIX box, or a hand-held device like a Palm. The Web is open to all.

However, while HTML is available to all, that does not mean that everyone experiences it the same way. A standards committee called the World Wide Web Consortium (W3C) (<http://www.w3.org/>) governs HTML. The current HTML Specification, Version 4.01 details the recommendations for compliance endorsed by the W3C (W3C, 1999). As a *recommendation* to consortium members (of which Microsoft and Netscape, among others, are members) the HTML standard is non-binding, and as such there is not universal compliance with or standardized implementation of HTML among browsers. Each browser developer has their own vision of the future direction of HTML and they routinely implement proprietary HTML extensions to their own browser. Even the implementation of the HTML Standard is not consistent across browsers, which cause Web pages displayed on competing browsers to render differently. This creates a dilemma for Web developers: (1) they can embrace the featureset of a particular browser, potentially alienating the users of browsers which do not support the chosen features;

(2) they can develop multiple (parallel) websites, each optimized for a particular platform and/or browser, or (3) they can forgo the latest features available in the interest of "universal" compatibility. In fact, according to The Web Standards Project (www.webstandards.org) founded by a coalition of designers disgusted with the increasing fragmentation of the Web, Web designers waste an incredible 25 percent of their time devising workarounds for proprietary tags, writing multiple versions of pages to satisfy each browser, and educating their clients about the impossibility of creating certain effects for all browsers (Castro, 2000).

The HTML standard has different levels of support. If an application developer wants all or most browsers to render pages consistently, they should use HTML language structure of the lower levels. The lower level features do not give as many choices for delivering data as the higher level features do. A choice must be made between features and the quantity of browsers that can view the data. Table 1 shows the different levels of support and the general features that were added with each level. Level 0 support is necessary, regardless of the kind of browser type, including text-only browsers.

HTML Level	Features Supported
Level 0	Mandatory: Headings, lists, anchors, etc.
Level 1	Images, emphasis, text highlighting
Level 2	Forms, character definitions
Level 3	Tables, figures, etc.
Level 4	Mathematical formulas

Table 1. The levels of HTML support. (From SYBEX, 2001)

The HFACS–ME Web prototype attempted to adhere to the HTML 4.01 standard to minimize incompatibilities and display differences among browsers. This means that emerging or proprietary technologies, such as DHTML and ActiveX have been avoided. This policy is summarized in the following excerpt from the Internet Technology Standards Guidance, issued by the Department of the Navy's Chief Information Officer:

With the aggressive addition of new web browser features by both Netscape and Microsoft, adherence to the HTML standard provides the only guarantee of compatibility with all web browsers. Features beyond the current HTML standard should be used only when the user is certain that both vendors support the new feature. Even adherence to the HTML standard is not a guarantee that all users will be able to access all the functionality of a web site. A site developer must ensure that the entire target audience has, or will, upgrade their browser to the version that supports the current HTML standard or added feature before fielding new HTML features. (DON CIO, 1999)

That said, in a situation where an incompatibility might exist without a universal solution, this research opted to use Microsoft Internet Explorer 5.0 as the standard browser platform. This decision is based on the Navy's selection of Microsoft products for both desktop operating systems and office application suites (both of which with MS Internet Explorer are bundled). (CINCPACFLT, 1997)

2. Client-Side Scripting

Another feature that is dependent on the browser is client-side scripting. Client-side scripts are programs that are passed to and executed on the client machine. There are a number of benefits of transferring some of the workload from the server to the client and of course several drawbacks. Probably one of the most useful tasks for client-side script is for form validation. By validating the data before it is submitted to the server, network traffic and server-side errors are reduced. It should be noted that the HFACS–ME Web interface takes advantage of standard HTML drop-down boxes for data input. This ensures that data being submitted to queries is valid and correctly formatted, and minimizes the need for client-side data validation.

Still other uses for client-side scripts are to dynamically change the appearance of HTML page elements, perform specific actions when trigger events occur (i.e., when the user clicks on a button, or the cursor passes over a hyperlink), to modify design time control properties, and to handle client-side data processing, just to name a few. As with HTML, browser implementation and client security settings determine whether the scripts will run. Most client-side scripts are written in either Visual Basic Script

(VBScript) or JavaScript. VBScript is only compatible with Internet Explorer and due to potential security issues is not as widely used as JavaScript. Java's security model is commonly called the "sandbox model." The idea is that the applets, like toddlers, get to play in a sandbox, but the walls of the sandbox are high enough that they cannot easily get out and do any damage. Currently, VBScript does not afford the same protection. The HFACS–ME Web application uses very limited client-side scripting for navigation display, event triggers, and the dynamic rendering of the Graph display.

D. WEB SERVER

In order to serve files to Web clients (browsers), a network application that can intelligently interpret and fulfill client requests were needed. The application that performs this service is called the Web server. Web servers are background processes that run transparently to system users by taking advantage of a capability called *multitasking*. This ability to run multiple processes simultaneously on single computer enable a Web server to service requests from multiple clients at the same time. By breaking the application into multiple parts that can execute in parallel, the operating system can rapidly switch between threads of execution and divide processor time between them. In this manner, Web servers become more responsive, run smoother, and client's requests are not left sitting in a long queue for processing.

For this project Microsoft's Internet Information Server, version 5.0 was selected as our Web server for the following reasons: (1) IIS comes as part of the Windows Server operating system so no additional cost and minimal integration effort were required, (2) Microsoft Web development tools (MS FrontPage and Visual InterDev) were available for Web site development, and (3) the author was familiar with MS Visual Basic and VBScript. With the Web server chosen, the next step was to investigate the means create dynamic Web pages

1. Active Server Pages (ASP)

ASP is a technology introduced with the release of Microsoft's Internet Information Server 3.0. ASP is a proprietary technology that only runs on Microsoft's line of Web server applications, the most recent being Internet Information Server 5.0.

Alternatives to ASP include Sun's Java Script Pages (JSP), Chili!Soft ASP and Hypertext Preprocessor (PHP).

ASP adds the capability of customizing Web pages on the server before they are transmitted over the network to the browser requesting them. With ASP, program scripts can be added directly to the Web page and inserted precisely between the HTML instructions that need to be customized. Instead of writing lengthy, hard-to-read C programs or Perl scripts that disguise the HTML instructions, ASP allow developers to put code right where it is needed. A developer can start with a complete, viewable HTML document and incrementally customize it with small fragments of script code that automatically direct the changes to the document when the code is executed on the server.

As mentioned previously, it is possible to use client-side scripting to customize a Web page after it has been downloaded. This code could actually alter the page's appearance by modifying the HTML instructions before the browser has examined it. Web pages with extensive client-side customization code are much larger than pages that are designed once and always have the same appearance. However, smaller Web pages are more popular than larger ones because they transfer faster and are more likely to be revisited. With tools such as ASP, however, these drawbacks can be overcome by server-side customization.

Customization on the server allows the workload to be placed on the server, not the client. More important, the code executed by the server is not transferred to the client. Only the resulting Web page is transferred, and it is often as small and streamlined as a non-customized page. Customization is worthwhile only if it results in Web pages that are compelling to either the company developing the Web site or the user accessing it. There may be reasons to add tricks such as randomizing ads so that different images are displayed each time a Web page is hit, but these reasons are rather trivial compared to Web sites that can automatically customize themselves based on user preferences, history of previous visits or purchases, current events, or news. Information is the missing piece, and more often than not, information is tucked away inside overwhelmingly arcane databases.

One advantage to customizing with ASP is that the Web server has access to databases that are hidden from client computers. Web servers on the Internet are often placed behind corporate firewalls, which act as barriers that protect computers in a company from the wild, unguarded chaos of the global Internet. Requests, such as those for Web pages, are often allowed to travel through the firewall. These requests are considered harmless and are usually the main reason a corporate network is connected to the Internet in the first place.

Database servers, however, are often considered too critical to a company's existence to risk opening them up for direct contact with the outside world. Web servers are privileged because normally they exist logically on the same side of the firewall as the database server, so they can access the databases directly. This restriction is compounded by most databases being accessible only to machines on a local area network. Local area networks have the advantage of being faster and more reliable than distributed networks, such as the Internet, and databases are often designed with this advantage as a requirement for operation. The more reliable a network, the less likely data will be lost during transmission or a transaction will be broken in mid-step. Proprietary protocols are often devised to reduce these risks even further. Unfortunately, these protocols are often bound to the same local area networks the databases were designed for, so even if the database servers could be connected directly to the global Internet, it is unlikely they could be communicated with using Internet protocols.

Active Server considers everything that appears between the bracketing symbols, `<%` and `%>`, to be scripting code. All VBScript statements executed by the server before the page is transmitted must appear between these two brackets. Almost any valid script can be used as complex as many statement lines and complete functions or sub-routines, or as simple as partial fragments of statements or expressions. As far as Active Server is concerned, all scripting code placed between these brackets is seen as one long sequential script. When processing the Web page, Active Server ignores the HTML instructions, executes the combined program, and replaces the bracketed sections with new HTML instructions generated by the code in the corresponding locations. None of the code that appears between brackets, or even the brackets themselves, ends up in the HTML document that is eventually sent to the client. This final statement is particularly

important. It means that the browser (and inquisitive users) are not able to see the business rules, programming logic, or database connection details that are used to generate the returned page because that data is never sent to the browser, only the resulting HTML code.

2. ActiveX Data Object (ADO)

ADO offers a wide range of functions that can be used as a uniform interface to access any database—including SQL Server, Oracle, and Microsoft Access. To accomplish this ADO relies on a relatively recent Microsoft technology known as OLE DB, which was designed to provide database access through COM-based interfaces. OLE DB, in turn, superseded ODBC. OLE DB has a number of advantages over ODBC, particularly thanks to certain Win32®-friendly design decisions. For instance, OLE DB provides mandatory support for multithreading (a firm requirement for COM objects). Figure 4 depicts the relationship among ADO, OLE DB, and ODBC.

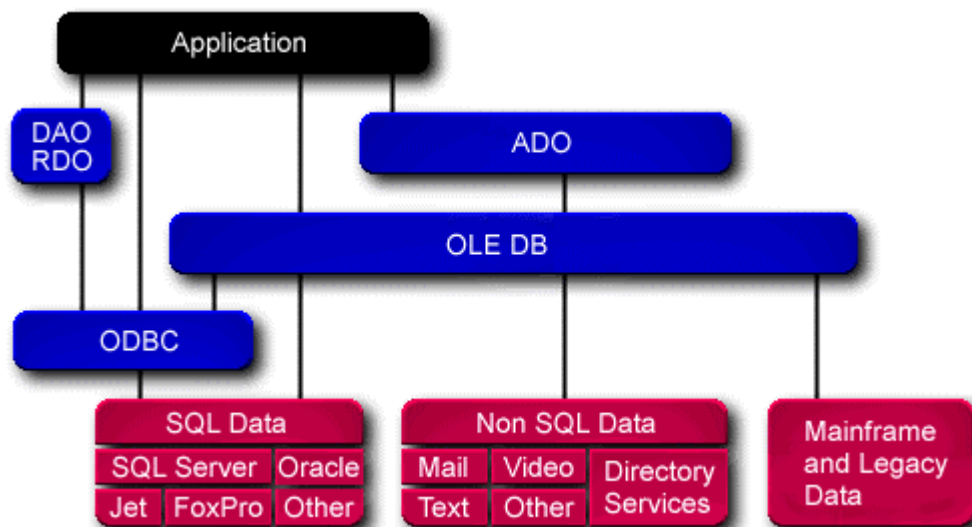


Figure 4. Comparison of ADO, OLE DB, and ODBC (From <http://www.microsoft.com/data/archb.htm>)

Where ODBC relies on the functionality provided by specific drivers to access the various DBMSs, OLE DB works by delegating this task to a provider. A provider is

simply a piece of software that has been developed to be compatible with OLE DB and supply it with data. Because of the vast quantity of ODBC legacy code, ODBC is an OLE DB provider. ODBC is not always a safe choice, particularly because it does not guarantee its drivers to be thread-safe, but should be okay if the latest SQL Server, Microsoft Access, or Oracle drivers are used. There are quite a few things going on behind the scenes when ADO is used to access a database. Requests go through up to four different pieces of middleware before they actually end up on the database engine.

One component of ADO that deserves special attention is the Connection Object. This represents the link between the Web server and the database server. All communications between the ADO Recordset Objects (used to return data from SQL calls) and Command Object (used to call stored procedures), and the database server are negotiated through the Connection Object. By controlling various Connection parameters a web developer can control when and how the connection to the database is made and when it is shut down. As far as the database is concerned, users do not interact with the database, Connections do. To the database, Connections are the points of contact to the outside world, and even though multiple Recordsets may be open at a time, to the database they are one multifaceted interaction. To use an analogy, say you and a group of friends hop in the car and pull up to a drive-up window at a fast food restaurant. The driver acts as the Connection Object for the requests passed by the other passengers, and the person taking the order views it as one order.

To make use of the Connection Object, several things have to happen. First the Connection Object has to be created. This is normally accomplished by issuing a statement like:

```
Set conn = Server.CreateObject("ADODB.Connection")
```

Once the Connection Object is created, parameters can be passed to it to define the connection provider (i.e., OLE DB, ODBC, DNS, etc.) the data source, and connection authentication parameters such as a data source UserID and Password, among others. An example is listed below:

```
conn.ConnectionString = "Provider=SQLOLEDB;  
    Source=DATASERVER1; Database=HFACS; UID=sa;PWD=;"  
conn.open
```

In the previous example, an OLE DB data provider, specifically designed for use with SQL Server (Provider = SQLOLEDB), is identified, followed by the name of the database server to which we are trying to connect (Source = DATASERVER1) (Note: in the example, DATASERVER1 is the NetBIOS name of the computer the database server is running on. This source could have taken the form of an IP address, had the database server not been within the same domain as the Web server, or could have been a specific database file (i.e., c:\inetpub\safety\access\hfacs.mdb, etc.) if the data source were Microsoft Access. The Database parameter identifies the specific database to which we are connecting (Database = HFACS) and finally, UID and PWD parameters provide a means of authenticating the connection to the database. Finally, the last statement (conn.open) opens the connection for use by Recordset and Command Objects needing to retrieve data from the database.

One caution about Connection Object security. Because the connection parameters are written out in plain text, some means must be employed to prevent users from viewing the connection data and gaining unauthorized access to the database. One way to accomplish this is to use the .asp extension on the files containing connection information. In doing so, whenever the page is requested it will be sent to the ASP interpreter and all non-HTML code (including the Connection data) will be removed before it is displayed to the client.

E. DATABASE SERVER

At the heart of most client/server systems, database servers provide secure access to shared data for client applications that request database services. These database services not only include access to the raw data, but also data integrity management, business rule enforcement, database maintenance and recovery and other functions. These functions are normally packaged under an overarching application called a Database Management System (DBMS).

1. DBMS Selection

One of the most significant design decisions made during this project was upon which database management system to base the server. Earlier versions of the MEIMS stand-alone system were based on Microsoft Access 97 and the initial prototype implementation undertaken continued using Access (albeit Access 2000). As a prototype, HFACS–ME Web was viewed as a proof-of-concept demonstration and the necessary Web functionality was believed to be achievable without the additional cost of purchasing an enterprise-level DBMS. Unfortunately, this assumption was quickly proven incorrect and the limitations of Access in a distributed environment became apparent. This required that all code written in support of the Access 2000 back-end had to be rewritten to connect with a new DBMS.

The capabilities of three different DBMSs available from Microsoft were examined in the course of this study. They are, in order of increasing capability: Microsoft (MS) Access, Microsoft Data Engine (MSDE), and Microsoft SQL Server.

a. MS Access

MS Access is a small, inexpensive database application with a strong emphasis on presenting data and making it easy to use. It has wizards built into the program that can do almost everything, including reports. Its programming language is easy to learn and the query tools use drag-and-drop technology. Much of the SQL code that would be used to manipulate data is hidden from the user. The user simply drags and drops the columns of interest from the tables. The learning curve for this program's database is small compared to MS SQL Server. It also has the ability to run on Windows NT or Windows 9X¹. It works well for small databases, but once the database size grows beyond the recommendations given in the product manual, performance (in terms of speed) will suffer. There are also limitations on the complexity of database queries that can be handled (this fact, which surfaced in an early phase of development, was probably

¹ Windows 9X notation indicates compatibility with Windows 95, Windows 98, Windows 98 Second Edition, and Windows ME operating systems. Windows NT indicates compatibility with Windows NT 4.0, Windows 2000 Professional and the Windows Server family of operating systems.

the most influential in moving the author to press for a more robust and capable DBMS). Finally, the extremely limited security model is not appropriate for a Web database.

b. MSDE

MSDE is a free (for owners of MS Access or Office 2000 Professional), less fully-featured version of SQL Server 2000 and SQL Server 7 that can be used to build true client/server databases with Access 2000's new ADP project file type. This innovation provides a GUI to a SQL Server data source through a single OLE DB connection. Access projects can be used with both MSDE and the desktop, full, and enterprise versions of SQL Server. MSDE allows users to gain many of the benefits of SQL Server including improved robustness and better security, but at the cost of added administrative complexity. In addition, MSDE's scalability is limited in isolation (for example, it can accommodate up to two processors on a Windows NT box, and it has a 2 gigabyte (GB) database size limit), but is limited to five concurrent connections (a killer for a Web server). However, MSDE has the same database format as SQL Server, so prototyping (or a starter) solution using MSDE can be undertaken. Then, when the application grows in popularity or it is time to roll it out from its prototype test environment, it can be migrated easily to any version of SQL Server.

c. SQL Server 2000

MS SQL Server is designed as a true client-server database application in which most data crunching happens on the server. The client makes requests and accepts results, which are simple operations compared to manipulating data. MS SQL Server is meant to be a full-blown true database application. Therefore, there is no windowed representation of the data and no built-in reports (although it does come with Enterprise Manager and Query Analyzer applications for database management). All the functionality lies in its ability to handle and secure large amounts of data with many users, which is a huge amount of processing. Because of the performance emphasis, SQL Server 2000 is designed to run Windows Server operating systems only (although Server Administration tools are available to administer the SQL Server installation from a

Windows 9X machine). It handles all the major database issues but requires the administrator and users to be very knowledgeable about databases.

Another strength of SQL Server over Access is the ability to perform transaction processing and error recovery. MS Access does not handle true transactions. Even when it says a transaction is committed, its caching schema might not allow other users to see the changes. Also, there is no way to roll back committed data, so if a delete erases important data, the only recovery option you have is to restore the *.MDB file from a backup. MS SQL Server, however, can handle these issues by including a transaction processor. When transaction processing is turned on (by using SQL commands), the data can be either committed or rolled back. When transaction processing is not turned on, the data is automatically committed to the database (no caching involved).

One of the most crucial elements of a Web database is security. MS SQL Server uses a strong security model that can restrict a user's access rights on a very precise scale, such as allowing read-only access to Table A, but full access to Table B. MS Access offers no such capability.

2. Database Design

A relational database is a collection of related data. A database usually represents some collection of real data, such as accounting information or a record of aviation mishaps. The data usually has some logical relationship; in other words, a random collection of data would not be called a database. A database usually has a specific purpose, such as finding all the mishaps in a collection that fit the category of "Class A" or an accounting database that finds all the entries for a particular client.

A relational database has some benefits; the main one being that duplicate data, such as Mishap Factors from the same Mishap, can be reduced. The information can be stored just once and then referenced by all the duplicates. This benefit is twofold. First, each time a change has to be made to the information, it has to be made only once, instead of being changed for each reference to the information. Second, the amount of disk space needed is greatly reduced. This concept of reducing redundant data is known as *normalizing a database*

Normalization consists of the standard rules of predicate calculus applied to relationships to prevent a design that can cause repeated and inconsistent data. Poorly designed relationships gives rise to complex SQL statements, with multiple joins, necessary to re-mold the structure. To achieve each of the respective normal forms, the table definitions must ensure the following:

- First Normal Form – Removes all repeating groups of data by giving each logical group a separate table and providing a primary key in each.
- Second Normal Form – Key fields are chosen so that non-key fields depend on all fields in the primary key.
- Third Normal Form – No fields depend on other non-key fields.

The goal was to adopt third normal form for all tables, however a conscious decision was made to stop short of 3rd Normal Form in a lookup table that details the relationship between 1st, 2nd and 3rd Level Factors (tblFactors). Given that this table is only 34 records long and unlikely to grow significantly, the additional query complexity that would have resulted if the table had been split into three separate tables (as required by 3rd Normal Form) would have had a detrimental affect on server performance. The process of compromising strict adherence to normalization rules in favor of performance is called *denormalization*. The final HFACS–ME database structure is illustrated in Appendix A, Figure A-1.

a. Database Language

A key component of any relational database consists of methods for entering data into the database and getting data out of the database. Most databases support some sort of programming language for performing these tasks. SQL (pronounced *sequel*) is often cited as being the *lingua franca*² of relational database management systems. Certainly no other database language has found such wide acceptance among such a broad range of products. Since it was first standardized in 1986, later revised in 1992 (SQL92), and again in 1996 (SQL96), SQL has become universally adopted. Even non-relational database systems support a SQL interface. But, unlike other

² **lingua franca** - A medium of communication between peoples of different languages.

computer languages such as C or COBOL, which are the exclusive domain of programmers, SQL is employed by a variety of professionals. Programmers, database administrators, and business analysts alike use SQL to access information. SQL allows users to access data in relational database management systems, such as Oracle, Sybase, Informix, Microsoft SQL Server, Access, and others, by allowing users to describe the data the user wishes to see. SQL also allows users to define the data in a database, and manipulate that data. The language supports creating database structures, such as tables and indexes (called *Data Definition Language* or *DDL*), but also allows for data manipulation, such as entering and updating data (called *Data Manipulation Language* or *DML*). It should be pointed out that, like browsers that are governed by the HTML standard, Relational Database Management Systems (RDBMS) normally are designed to comply with the current SQL standard and then proprietary extensions are added to compensate for specific limitations or to add functionality. Microsoft's SQL Server 2000 uses Transact SQL (T-SQL) and Oracle 8 uses Procedural Language extension to SQL (PL/SQL).

3. Stored Procedures

Stored procedures are collections of SQL statements stored in a SQL Server database. They can be simple SELECT statements or complex queries with embedded logic and conditional execution statements and they can be invoked directly from a front-end application (including an ASP page). Stored procedures offer a number of benefits in terms of scalability, performance and security.

Stored procedures assist in achieving a consistent implementation of logic across applications. The SQL statements and logic needed to perform a commonly performed task can be designed, coded, and tested once in a stored procedure. Each application needing to perform that task can then simply execute the stored procedure. Coding business logic into a single stored procedure also offers a single point of control for ensuring that business rules are correctly enforced across a range of applications.

Stored procedures can also shield users from needing to know the details of the tables in the database. If a set of stored procedures supports all of the business functions users need to perform, users never need to access the tables directly; they can just execute

the stored procedures that model the business processes with which they are familiar. In addition to simplifying the application for the end user, this is also a security feature in that users can be restricted from performing operations for which they are not authorized.

Stored procedures can also dramatically improve performance. Many tasks are implemented as a series of SQL statements. Conditional logic applied to the results of the first SQL statements determines which subsequent SQL statements are executed. If these SQL statements and conditional logic are written into a stored procedure, they become part of a single execution plan on the server. The results do not have to be returned to the client to have the conditional logic applied; all of the work is done on the server. The IF statement in this example shows embedding conditional logic in a procedure to keep from sending an empty result set to the application if insufficient stock is not on hand:

```
IF (@QuantityOrdered < (SELECT QuantityOnHand FROM Inventory
WHERE PartID = @PartOrdered))
BEGIN
    -- Stock on hand. SQL to update tables and process order.
END
ELSE
BEGIN
    -- Stock on hand insufficient, find alternative items
    -- SELECT statement to retrieve the IDs of alternate items
END
```

Similarly, applications do not need to transmit all of the SQL statements in the procedure: they have to transmit only an EXECUTE or CALL statement containing the name of the procedure and the values of the parameters. The corresponding stored procedure is then retrieved from the database and processed on the database server. The following code is all that is needed to call the stored procedure that provides the data to populated the Factor Analysis table:

```
cmd.CommandText = "spMishapCount_Filtered_with_Factors"
Set rsCodes=cmd.Execute
```

The first line assigns the stored procedure name to the CommandText parameter of the Command Object (cmd). Then the Command Object (with its associated

parameters) is executed, and the resulting Recordset is stored in a local variable called rsCodes for later use. As an example of how significant the performance impact can be, consider the code necessary to produce a report using HFACS–ME Web. To produce a report without using stored procedures would require the Web server to pass approximately 145,000 bytes of SQL code to the database server. Using stored procedures (the method currently implemented) the only data that need be sent from the Web server to the database server is the name of the stored procedure (approximately 22 bytes of data). To put this into perspective, if it took one second to transfer the stored procedure name (22 bytes), it would take 1 hour and 50 minutes to transfer the SQL statements before processing could begin. Although this example exaggerates the transfer time dramatically, it clearly illustrates the performance gain achievable with stored procedures.

SQL Server 2000 and SQL Server 7.0 incorporate improvements to earlier versions of statement processing that extend many of the performance benefits of stored procedures to all SQL statements (not just stored procedures). SQL Server 2000 and SQL Server 7.0 do not save a partially compiled plan for stored procedures when they are created. A stored procedure is compiled at execution time, like any other T–SQL statement. SQL Server retains execution plans for all SQL statements in the procedure cache, not just stored procedure execution plans. The database engine uses an efficient algorithm for comparing new T–SQL statements with the T–SQL statements of existing execution plans. If the database engine determines that a new T–SQL statement matches the T–SQL statement of an existing execution plan, it reuses the plan. This reduces the relative performance benefit of precompiling stored procedures by extending execution plan reuse to all SQL statements. Without the benefit of stored procedures, every time a query is passed to a server, the following actions are required:

- The server receives the query's text.
- The server interprets the query, making sure that the syntax is correct.
- The server creates a "plan" to access the database in the optimal way according to what is requested by the query. This includes finding the most appropriate indexes and sorting algorithms to use.

- The server finally starts accessing the data in accordance with the plan that it just calculated, returning the result set to you.

From the preceding discussion it should be clear that stored procedures have many benefits. Clearly, in an Web-based application where multiple simultaneous connections to the database server are likely, performance and scalability are critical. One other benefit, briefly mentioned above, relating to restricting access to unauthorized users leads into a discussion of security.

F. SECURITY ISSUES

One of the most critical tasks in establishing a Web site is assessing the need for site security, access control and protection of sensitive data. Although the full technical details of Web site security implementation are beyond the scope of this thesis, there are a number of software security issues that need to be addressed. With a Web application like HFACS–ME Web, each software component, the Internet Information Server, the SQL Server database and the Windows NT operating systems upon which they are installed, each brings with it certain security capabilities. Given that Microsoft produces all the products they are designed to complement each other and provide overlapping security features. However, this overlapping of "protection" occasionally creates situations where a change in one area creates unintended consequences in others. Careful analysis and documentation of changes is critical to prevent an overly restrictive or ineffective security policy from resulting.

1. Access Control

Each application used in the HFACS Web offers the ability to restrict access to objects under their control. The Windows NT operating system is built upon a robust security architecture that uses the concept of Users and Groups to determine permissions. When a user attempts to logon to a computer running Windows NT, the Security Account Manager (SAM) checks their User Name and Password against its database and grants access if it finds a matching entry. (If the computer is part of a computer domain then the login is checked against the Domain Controller's database). As the user requests access to resources on the computer (i.e., files, folders, printers, etc) the user's credentials are

compared to the resource's Access Control List (ACL) to determine the level of access permitted. Note, the ability to enforce permissions on file and directory resources is dependent on whether the New Technology File System (NTFS) is being used. With permissions the administrator can implicitly or explicitly, allow or disallow access to the local computer and/or domain resources.

Internet Information Server also has the ability to restrict access based upon User accounts (or can allow Anonymous access), but it can also restrict connections based on the IP address of the client, the domain (e.g., .mil, .com, .org, etc.), and the TCP port (the default TCP port for HTTP is 80). It can also restrict what can be done once a connection has been made. For instance, a user may only be authorized Read permissions that would allow them to view .htm or .html pages, but not .asp pages (which require Read & Execute permissions because of the embedded script).

Finally, SQL Server has its own permissions, which as mentioned before, can grant or deny access to any database object based on the user or their role.

2. Secure Communications

What about the untamed world of the Internet at large? Given the potentially sensitive nature of the mishap data that will be accessed using the HFACS Web, a means to ensure that the data cannot be casually viewed in transit is necessary.

The most common method for encrypting data traversing the Internet is through the use of Secure Sockets Layer or SSL. Take using a credit card to pay for goods and services across the Internet, for example. Without encryption of the request from the browser to the Web server, anyone who has a tool to read TCP /IP packets would be able to see the credit card information, as long as they are along the route of the packets. However, if the HTTP request is encrypted, the credit-card information could still be seen, but it would make no sense. This type of encryption in HTTP is known as *HTTPS*, also referred to as *secure server*. IIS has the ability to enforce SSL encryption with the use of a key or digital certificate Using IIS' Certificate server we could generate a public/private key pair for the HFACS–ME Web service, then we could distribute our self-signed certificate to authorized users. With the appropriate certificate installed, when a HTTP request arrived requesting a secured page, the server and the requesting browser

would negotiate a session key to be used during the transmission such that only the holder of that session key could decrypt the message.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. METHODS

A. RESEARCH APPROACH

A Web-based analysis and reporting tool for maintenance error in aviation would greatly facilitate Naval Aviation's effort to capture the human factors in mishaps and develop appropriate interventions. The Human Factors Analysis and Classification System–Maintenance Extension (HFACS–ME) Web application is a computer-based prototype based upon Web technology. The underlying Web pages were designed using FrontPage 2000, and are hosted on an Internet-connected Web server running Microsoft Internet Information Server (IIS) 5.0. The HFACS–ME data and business logic, originally contained in a Microsoft Access 97 database, was converted to Microsoft SQL Server 2000 in the form of stored procedures and views (see Appendix D). The prototype, which contains information on 595 maintenance error related mishaps that occurred between FYs 1990 and 1999, utilizes data derived from the NSC's Safety Information Management System (SIMS) database. The GUI for the system is provided by a Web browser and allows an end-user with basic computer skills to operate the system.

The HFACS–ME Web prototype was evaluated by a representative sample of potential end-users. The participants were provided a prepared task list that required them to navigate through and utilize features of the tool. At the completion of the task list, the participants viewed and used all portions of the prototype, and completed an exit survey composed of questions pertaining to demographic background information and both objective and open-ended items to elicit the participants' views of the usability of the system and value of both the system and the data. The objective data was transcribed into a Microsoft Excel spreadsheet for analysis while a content analysis was conducted on the open-ended survey questions. Note, the exit survey used only five Likert style questions because the major focus of the effort was the creation of the prototype vice the usability study. The questions were shaped intuitively and are considered to be simply the first stage of developing a formalized post-prototype tool.

B. HUMAN FACTORS AND ANALYSIS CLASSIFICATION SYSTEM–MAINTENANCE EXTENSION (HFACS–ME) WEB

1. Overview

The HFACS–ME Web prototype was designed to allow the user to access the database via four functional areas: (1) mishap data presentation and filtering, (2) HFACS–ME factor analysis, (3) graphical presentation of multi-dimensional data, and (4) Factor Distribution Report generation. Each function is displayed on separate pages with interactive controls providing the user interface. The following paragraphs provide a functional description of the prototype. It should be noted here that HFACS–ME Web was designed to be displayed in a browser window measuring 800 pixels by 600 pixels, or greater. One of the challenges to web developers, is to convey all the necessary information on a page without requiring the user to scroll (particularly horizontally). This situation has improved somewhat in recent years as the price of larger computer monitors has fallen, however nearly 50 percent of all users still use the 800 by 600 resolution. This has important implications for Web site designers who build sites for the lowest common denominator. As the majority of target users migrate to higher resolutions developers can design sites with additional content and graphics and be confident that a majority of Web users will be able to view them. Refer to Appendix B for screenshots corresponding to the HFACS–ME Web pages described in the following paragraphs. A full listing of all HTML and ASP code developed for the prototype is included in Appendix C.

2. HFACS–ME Web Homepage

The Homepage of the HFACS–ME Web prototype (see Figure B-1) is the initial page the users see after entering the appropriate URL in the address bar of their browser. Menu selections (hyperlinks) allow the user to select one of four main options:

(1) Mishap Data, (2) Factor Analysis, (3) Graph Data, and (4) Reports. Each item can be accessed by clicking on the hyperlink in the menu bar, located below the HFACS–ME logo. Help is provided to the user on this and all pages in the form of “tool tips” (i.e., brief description) when the mouse arrow is placed over a control (e.g., hyperlinks,

images, etc.). There are also links to support pages for an Online Tutorial, Help page, Access Policy page, and a Contact Us page.

3. Data Selection Page

The Data Selection page (see Figure B-2) is common to the Mishap Data, Factor Analysis and Graph Data sequences. On this page the user can use nine drop-down boxes to select what data is to be included in the dataset returned from the database. Each drop-down box has a selection of (All). When (All) is the selected item, no criteria is passed for that category to the SQL stored procedure.

The first six drop-down boxes are related to the mishap event itself, and they are: Aircraft Type (e.g., A4, F14, H53, P3, etc.), Mishap Type (i.e., Flight Mishap (FM), Flight-Related Mishap (FRM), and Aircraft-Ground Mishap (AGM)), Mishap Class (i.e., A, B, and C), Location (e.g., Ashore, Embarked, Detached, etc.), Service (e.g., USN, USMC, etc.), and Fiscal Year (i.e., 1990–1999). The final three drop-down boxes are related to HFACS–ME taxonomy factors that were present in a given mishap, and they are: 1st Level Factors (e.g., Management Conditions, Maintainer Acts, etc.), 2nd Level Factors (e.g., Crew Coordination, Medical, etc.), and 3rd Level Factors (e.g., Adverse Mental State, Attention/Memory, etc.).

Instructions are included to guide the user in selecting multiple items within a particular category, and an explanation of how the criteria are used within the query is given, for example:

Use the drop down boxes below to select the desired Mishap Criteria and Factors Criteria to include in the dataset. Multiple items within a particular criteria group (i.e. Aircraft Type = F14, F18) may be selected by holding down the Ctrl button and left clicking on the desired items. This will result in a query that will return data matching ANY of the selected items.

If criteria are selected in several criteria boxes (i.e. Aircraft type and Mishap Class), the resulting dataset will be only those records that match ALL criteria. For example, if the user selects *Aircraft Type=FA18, F14* and *Mishap Class=A*, then the database will return all records involving either a *F14* or *FA18* and resulted in a *Class A* mishap.

When the user has completed making all criteria selections, they click on the Submit button at the bottom of the page. This *on_click* event will fire the Submit event in the ASP code and submit all data from the nine drop-down boxes (and any hidden form values) to the next page. The Reset button allows the user to return all values to their original state when the page was first loaded. The Back button is equivalent to the user clicking on the browser Back button, in that it loads the previous page from the browser's History.

4. Mishap Details Page

The Mishap Details Page (see Figure B-3) displays the list of Mishaps matching the user-selected criteria. The total number of mishap matching the user's criteria is displayed at the top of the table. The mishap data is displayed in a table view that contains the following columns: Mishap ID, Aircraft, Type, Class, Location, Service, and Date. Even though the HFACS–ME Factors criteria (i.e., 1st Level, 2nd Level, and 3rd Level Factors) are used to filter the dataset, the factors themselves are not included in the table display. The Mishap ID is a hyperlink (blue text and underlined are standard to indicate the presence of a hyperlink), which when clicked on will load the Expanded Mishap View to display greater mishap and factor details.

5. Expanded Details Page

The Expanded Details Page (see Figure B-4) displays information specific to a single mishap. In addition to the basic mishap data displayed on the Mishap Details page, the Expanded Details page provides a mishap summary and a list of maintenance related factors identified in the mishap. For each factor identified, a factor summary and the 1st, 2nd and 3rd level HFACS–ME taxonomy factors associated with each are displayed. There is also a checkbox that allows the user to select verbose descriptions vice summaries to be displayed on this page. To change from summaries to long descriptions or back again, the user clicks on the checkbox and then on the Refresh button below it. The checkbox label changes based on the current display setting (i.e., when displaying summaries the label reads “Display Long Descriptions” to indicate the change that would

take place if the checkbox is selected. When long descriptions are being displayed the label reads “Display Summaries”).

6. Factor Analysis Page

The Factors Analysis page displays a composite view of the HFACS–ME factors represented in the database. When the page initially loads, the table displays the factor distribution across all mishaps contained in the database (see Figure B-5). The user can filter the data represented in the Factor Analysis table by using the drop-down boxes below the table (as previously described in the Data Selection page) The table is divided such that the leftmost column displays the 1st Level Factors, the second column displays the 2nd Level Factors, and the remaining columns display the 3rd Level Factors.

1st Level Factor	2nd Level Factor	3rd Level Factor			
Management Conditions 449 75%	Organizational 215 36%	Inadequate Processes 94 16%	Inadequate Documentation 91 15%	Inadequate Design 77 13%	Inadequate Resource 4 1%
	Supervisory 335 56%	Inadequate Supervision 259 44%	Inappropriate Operation 78 13%	Uncorrected Problem 33 6%	Supervisory Misconduct 91 15%

Figure 5. Factor Analysis Layout

Figure 5 illustrates that the 1st Level Factor-Management Conditions has two 2nd Level Factors-Organizational and Supervisory. The 2nd Level Factor Organizational has four 3rd Level Factors: Inadequate Processes, Inadequate Documentation, Inadequate Design and Inadequate Resource. Each cell in the Mishap Analysis table displays the factor name, the number of mishaps containing that factor and the percentage of the total mishaps that it represents.

One important fact to note is that the number of mishaps listed for each of 1st Level Factor cannot be determined merely by adding up the number of mishaps listed in the 2nd Level Factors column. This is because multiple factors often are identified for each mishap. When two or more factors fall under a single higher-level factor, the higher-level factor is incremented once for each mishap and not for each factor, to avoid duplicate counts from skewing the higher-level factors.

7. Graph Page

When a user selects the Graph Data menu hyperlink, they are presented with a Graph Data Selection page, much like the Mishap Data Selection page (see Figure B-1). Once the user selects the criteria to be included in the graph and clicks on the Submit button, the Grouping Select page loads (see Figure B-6) and allows the user to select two of the three-axes (the third being the Factor Count, which is pre-selected) to be displayed on the subsequent Graph page. The possible selections for each axis are: Aircraft Type, Mishap Type, Mishap Class, Location, Service, Fiscal Year, 1st Level Factor, 2nd Level Factor or 3rd Level Factor. Once the axes criteria have been selected the user has the option to select Display Graph or Display Graph and Data Table. Both options open the resulting Graph page and the latter option opens a second window that displays a color-coded table of the graph data (see Figure B-7).

The number of mishap factors matching each of the criteria selected by the user is presented in a three-dimensional, multi-colored view (see Figure B-8). To aid in identifying specific data, each vertical column displays the values of each of its three axes when the user moves the cursor over any portion of the column. Due to the display size limitation, all graphs where HFACS–ME Factors (i.e., 1st, 2nd or 3rd Level Factors) are selected to display on an axis, display factor codes vice the long factor name. To assist the user in understanding the codes, a hyperlink, which opens an HFACS–ME taxonomy legend (see Figure B-9), is displayed next to the graph title.

8. Report Page

Clicking on the Reports hyperlink on the menu bar loads the Report Menu page (see Figure B-8). On this page the user is presented with six report options. Each report details the number and percentages of mishaps by HFACS–ME 1st, 2nd, and 3rd Level factors. The user may select from the following distribution presentations: Aircraft Type, Mishap Class, Mishap Type, Location, Organization and Fiscal Year (see Figure B-9). The resulting Report displays individual reports for each subgroup within the presentation category. Hyperlinks are displayed throughout the report to allow the user to move easily from one report to another.

9. Support Pages

Additional pages are provided to assist the user during their use of the website or to provide information about the site. Presently the Help and Online Tutorial pages are not operational. The Help and Online Tutorial menu items were added to the website menu bar as "placeholders" in anticipation of future development in the prototype.

C. DATA COLLECTION

1. Subjects/Participants

Students ($n = 12$) attending the Aviation Safety Officer (ASO) course at the School of Aviation Safety, and students at the Naval Postgraduate School in Monterey, California participated in the study. Participants were selected to represent a wide cross section of Naval and Marine Corps Aviators and Flight Officers from all aircraft communities. ASO course graduates are responsible for the management and implementation of squadron safety programs to include mishaps and include investigations and reporting. They are likely to be one of the primary end-users of the HFACS–ME Web application. Participant demographics were characterized by aviation background, computer experience, and availability of software and hardware systems used in the Navy and Marine Corps.

2. Apparatus

The completed HFACS–ME Web prototype consisted of four main sections: database queries, factors analysis, graphic presentations, and report generation. These sections allowed the participant to achieve the four functional requirements for the software tool: data collection, organization, analysis, and reporting (see Chapter 4 for a more complete description of the prototype). The HFACS–ME Web prototype was hosted on a publicly accessible web server at the Naval Postgraduate School, Monterey, California. The Web server used was Microsoft Internet Information Server, version 5.0 and the database server used was Microsoft SQL Server 2000. The prototype website was developed using VBScript, Javascript and HTML (see Appendix C). The use of Microsoft FrontPage 2000 was limited to basic website design and website management. No FrontPage generated ASP code was used in the website, in an effort to avoid

proprietary code that might limit the usability of HFACS–ME Web, if viewed with browsers other than Internet Explorer. FrontPage Server Extensions were installed on the IIS server to facilitate site management. The Usability Study participants were encouraged to utilize any Internet capable computer/browser to access the HFACS–ME Website. Each was provided a UserID and Password to allow site access.

3. Instrument

A participant usability survey was constructed, consisting of three parts: (1) Participant demographics, (2) Likert type assessment questions, and (3) Open-ended items. Collection of demographic information was accomplished through the participant selecting from a list of descriptors. Survey questions were designed to determine if the prototype software tool met participant query, reporting, and analysis requirements. The Likert questions used a five point rating scale with verbal anchors: Strongly Agree, Agree, Neutral, Disagree, and Strongly Disagree. Open-ended questions were included to gain subjective responses on the overall impression of the prototype Web tool, recommendations for improvement, and comments on areas not adequately covered by parts one and two of the survey.

4. Procedure

Prototype testing occurred over a span of two weeks. Participants were given an information packet consisting of an orientation sheet that described the purpose of the HFACS–ME Web research and the evaluation goals. In addition, they were given an eight-page guide to walk them through prototype testing (see Appendix E). The guide consisted of: (1) instructions for accessing the HFACS–ME Web Prototype--including website address and User ID/password, (2) prototype tool task list--a series of planned navigation routes within the prototype whereby the participant would be able to view the entire system, and (3) a participant's impression Exit Survey (see Appendix F).

It was expected that each participant would need 15-20 minutes to complete the process. Though information on time to navigate for each individual was not taken, informal feedback to the author indicated a range of 20-45 minutes with the longer times being needed for those with less computer experience. At the completion of the task list,

participants viewed all portions of the prototype system, and formed an opinion on its effectiveness. Participants then completed an exit survey composed of demographic background questions and perusal of the prototype system. Surveys were all submitted through a drop box provided in a common area.

D. DATA TABULATION

The data was transcribed from the survey onto a Microsoft Excel 2000 spreadsheet. The Likert questions, based on a five-point scale, were coded into Excel, using the values one (1) through five (5) corresponding with the anchors Strongly Disagree to Strongly Agree. Descriptive statistics were generated using Excel functions including the mean, standard deviation, range, and frequency distribution of the collected data. Content analysis was conducted on the responses provided from the open-ended survey questions. The categorization of participants by participant aircraft maintenance organization type and computer/software application experience level were noted.

E. DATA ANALYSIS

Basic and general information about the demographic and question results were depicted using descriptive analysis. Analysis of the results, including demographic information and satisfaction levels with the prototype was performed, using the functions of Microsoft Excel. The results are summarized as descriptive statistics (e.g., mean, standard deviation, mode, etc.), charts and summary tables in the following chapter. After compiling the analysis of the participant demographics, and Likert type assessment questions, a review of the responses to the open-ended items was conducted to identify common perceptions (both positive and negative) relating to the HFACS–ME Web prototype's design, functionality, and usability.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

A. SAMPLE

The 13-item exit survey was administered to participants from a School of Aviation Safety “Aviation Safety Officer” course and students attending graduate education at the Naval Postgraduate School. The participants were designated Naval Aviators, Naval Flight Officers, and Marine Corps Aviators. The group represented a cross-section of the aviation commands that make up the squadrons in the Navy and Marine Corps. Participants were asked to provide computer configuration information prior to commencing the Usability Study to assess how variations in browser type, video resolution and bandwidth might affect user satisfaction with the prototype (see Table 2).

Test Configurations	# of Participants	% of Total
Computer Type		
Government	8	66.7%
Personal	4	33.3%
Public	0	0.0%
Connection Type		
LAN	8	66.7%
Dial-Up	4	33.3%
Dial-Up Speed (bps)		
28.8k or less	0	0.0%
33.6k	1	8.3%
56k	3	25.0%
Not Applicable	8	66.7%
Screen Resolution		
Greater than 1024x768	3	25.0%
1024x768	6	50.0%
800x600	1	8.3%
640x480	0	0.0%
Don't Know	2	16.7%
Browser Used		
Internet Explorer	11	91.7%
Netscape	1	8.3%
Other	0	0.0%

Table 2. Participant Computer Configuration for Usability Study

B. DEMOGRAPHIC INFORMATION

The material collected in Part I of the exit survey consisted of demographic information and established the aviation and computer experience levels of each participant had both with computers and in aviation. The information is later used to determine if experience level in either category affected a participant's level of satisfaction and/or impacted the usability of the HFACS–ME Web prototype. The following paragraphs characterize the survey results for part I.

Question one revealed that ten of the participants were members of commands that performed maintenance at the squadron level (n = 10, 76.9%). One participant indicated that their command performed both Organizational and Intermediate level maintenance (n = 1, 7.7%). The remaining two participants were current students at the Naval Postgraduate School (n = 2, 15.4%). One NPS students indicated previous experience with organizational level maintenance at the squadron level.

Maintenance Affiliation	# of Participants	% of Total
Organizational Level	10	76.9%
Intermediate Level	1	7.7%
Depot Level		0.0%
No Maintenance Performed	2	15.4%

Table 3. Participant Maintenance Affiliation (n=13)

Question two indicated that the participant spends, on average, 4.95 hours per day using Web browser, email, word processor, spreadsheet and database applications. Of significance is that the average participant spends nearly two and a half hours a day using a Web browser while only half an hour is spent using a database. This suggests that the average user is likely to be more comfortable using a browser interface to retrieve safety data, rather than a stand-alone database application. This interface familiarity may also speed adoption of the HFACS–ME Web concept and reduce training requirements, as compared to a stand-alone application. The computer application usage distribution is depicted in Table 4.

	Web Browser	Email	Word Processor	Spreadsheet	Database
Average Usage (Hrs/day)	2.38	1.54	1.63	0.54	0.52
Std Deviation	0.86	0.72	0.68	0.33	0.63

Table 4. Number of Hours Per Day Participants Use Specific Software Applications (n=12)

Question three established participant's usage of different Web browsers. Not surprisingly (given DON adoption of Microsoft operating systems and Office suite), 100 percent of the participants (n = 12) stated they used Internet Explorer as their primary Web browser at work, and at home. One participant indicated that he used Netscape, at work and at home, in addition to Internet Explorer. Another participant indicated use of both Netscape and IE at home only. All participants indicate usage of browsers that have at minimum HTML Level 3 support and at least basic DHTML capabilities (see Table 5).

Browser Usage	Work		Home	
	Number of Respondents	% of Total	Number of Respondents	% of Total
Internet Explorer 6.x	3	25.0%	3	25.0%
Internet Explorer 5.x	10	75.0%	9	75.0%
Internet Explorer 4.x		0.0%		0.0%
Netscape 6.x		0.0%	1	0.0%
Netscape 4.x		0.0%	1	0.0%
Opera (any version)		0.0%		0.0%
Other		0.0%		0.0%
Unknown		0.0%		0.0%

Table 5. Participant Web Browser Usage Distribution (n=12)

Question four identified participants usage of a variety of computer operating systems (OS). As expected all users (n = 12) indicated the use of Windows NT as the OS for work computers. A majority (n = 7, 58.3%) use operating systems from the Windows 9X family at home, followed closely by Windows NT (n = 4, 33.3%). One participant used a Macintosh (n=1, 8.3%). Participants could indicate more than one OS but were not required to identify the specific OS used within an OS family (see Table 6).

	Windows 9X	Windows NT	Macintosh	UNIX	Linux	Other
Work		12				
Home	6	5	1			
Total	6	17	1	0	0	0
% of Total	25.0%	70.8%	4.2%	0.0%	0.0%	0.0%

Table 6. Normal Operating System of the Participants (n=12)

C. PARTICIPANT SATISFACTION WITH THE HFACS–ME WEB PROTOTYPE

1. Responses to Impressions and Usability Question

Part II of the exit survey examined a participant's impressions of the usability of the HFACS–ME Web prototype and its value to Naval Aviation. Participants responded to five statements selecting from one of the following responses: strongly disagree, disagree, neutral, agree, and strongly agree. Values of one (1) through five (5) respectively were assigned to the statements. The participants were also given the chance to make subjective comments on any of the five statements.

(a) Statement one asked whether or not a participant found the prototype to be presented in a logical form. The histogram of the frequency distribution for statement one is presented in Figure 6. The mean was 4.66, standard deviation = 0.492, range = 2. All participants (n = 12; 100%) agreed that the prototype was designed and presented in a logical fashion.

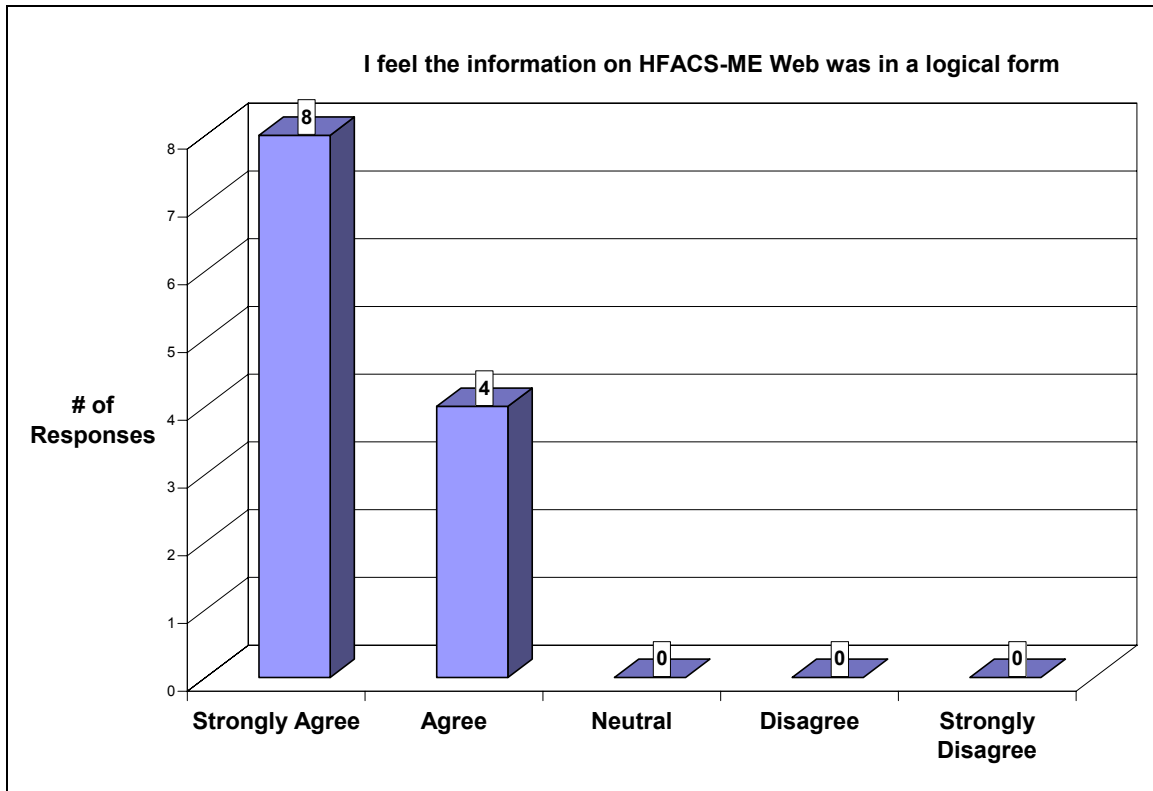


Figure 6. Exit Survey, Part II, Statement One, Response Distribution

(b) Statement two asked about the ease of navigation of the prototype. The histogram of the frequency distribution for statement two is presented in Figure 7. The mean was 4.67, standard deviation = 0.492, range = 2. All of the participants ($n = 12$; 100%) agreed that the prototype was easy to navigate.

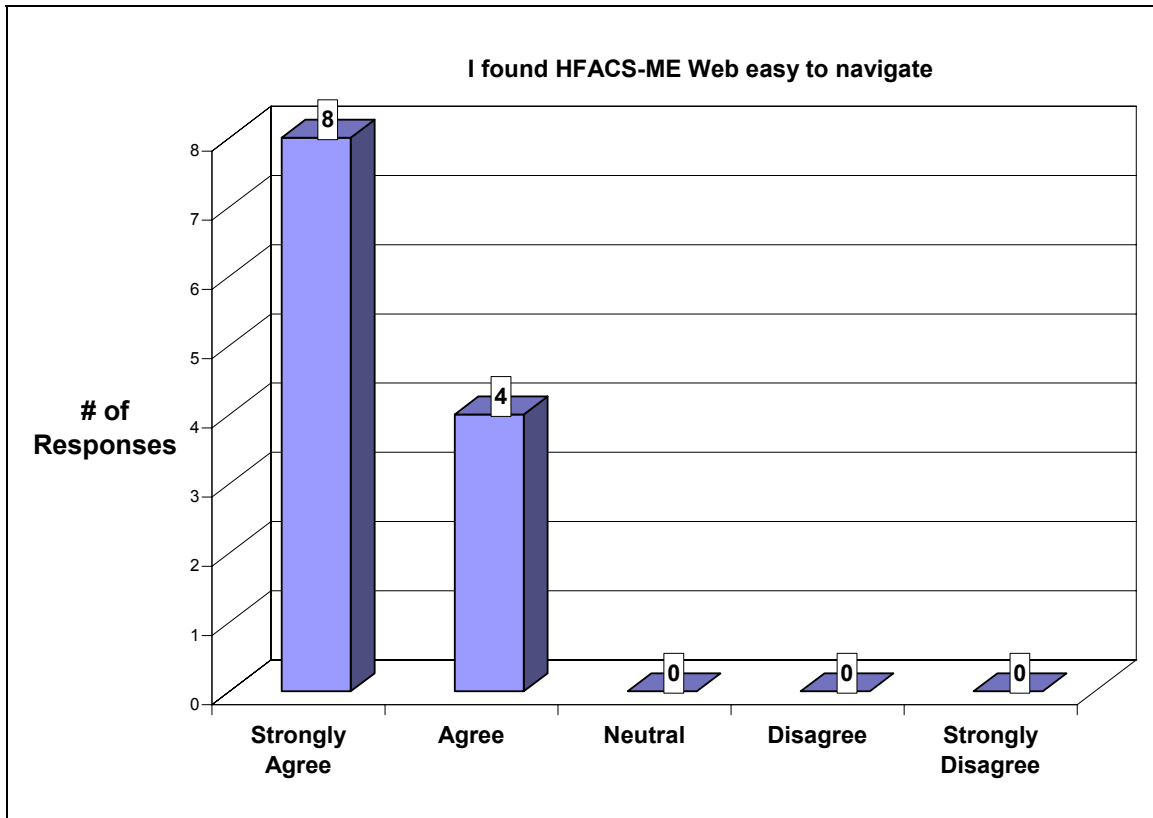


Figure 7. Exit Survey, Part II, Statement Two, Response Distribution

(c) Statement three. The participants were asked whether they felt HFACS–ME Web was “interesting.” The histogram of the frequency distribution for statement three is presented in Figure 8. The mean was 4.58, standard deviation = 0.669, range = 3. Most of the participants ($n = 11$; 91.7%) indicated the prototype was of great interest to them.

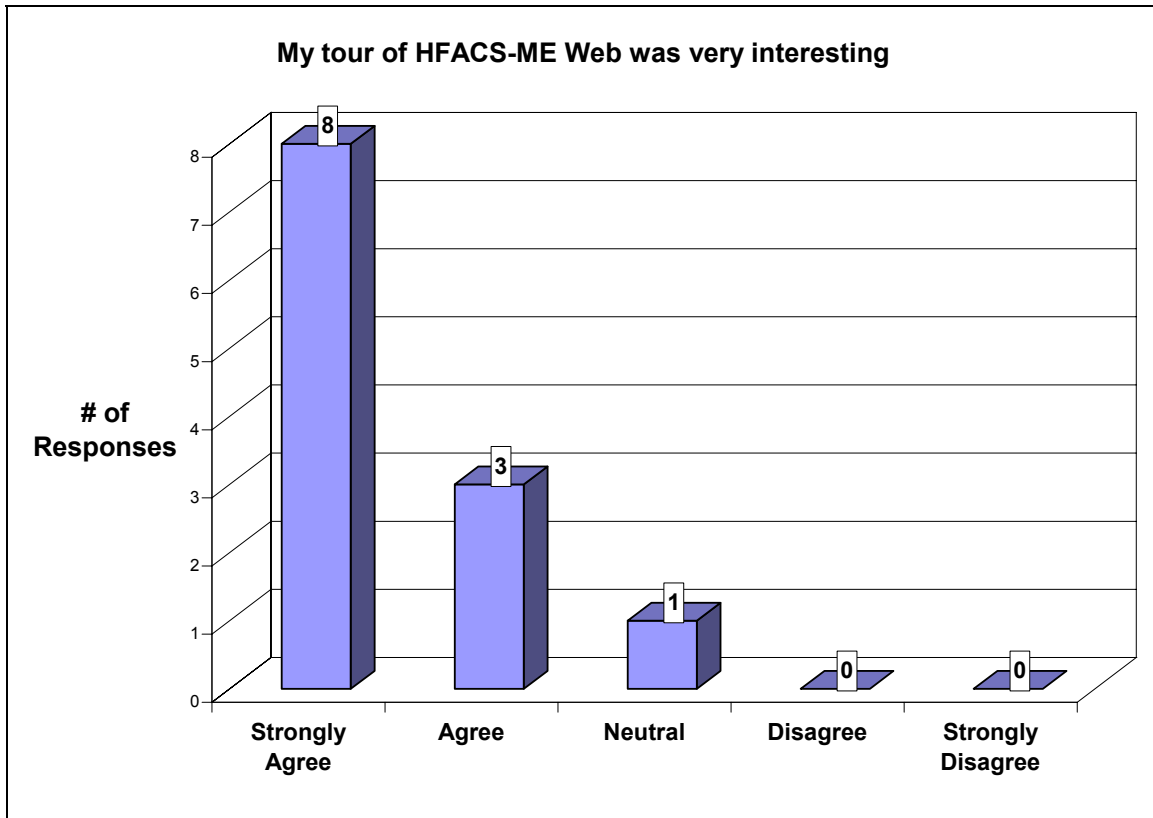


Figure 8. Exit Survey, Part II, Statement Three, Response Distribution

(d) Statement four asked about the relevance of the prototype to aviation maintenance operations. The histogram of the frequency distribution for statement four is presented in Figure 9. The mean was 4.54, standard deviation = 0.688, range = 3. Most participants ($n = 10$; 90.9%) indicated the prototype was highly relevant to maintenance operations. The one participant who did not respond to statement four had previously indicated that the command to which he/she is attached does not perform aircraft maintenance.

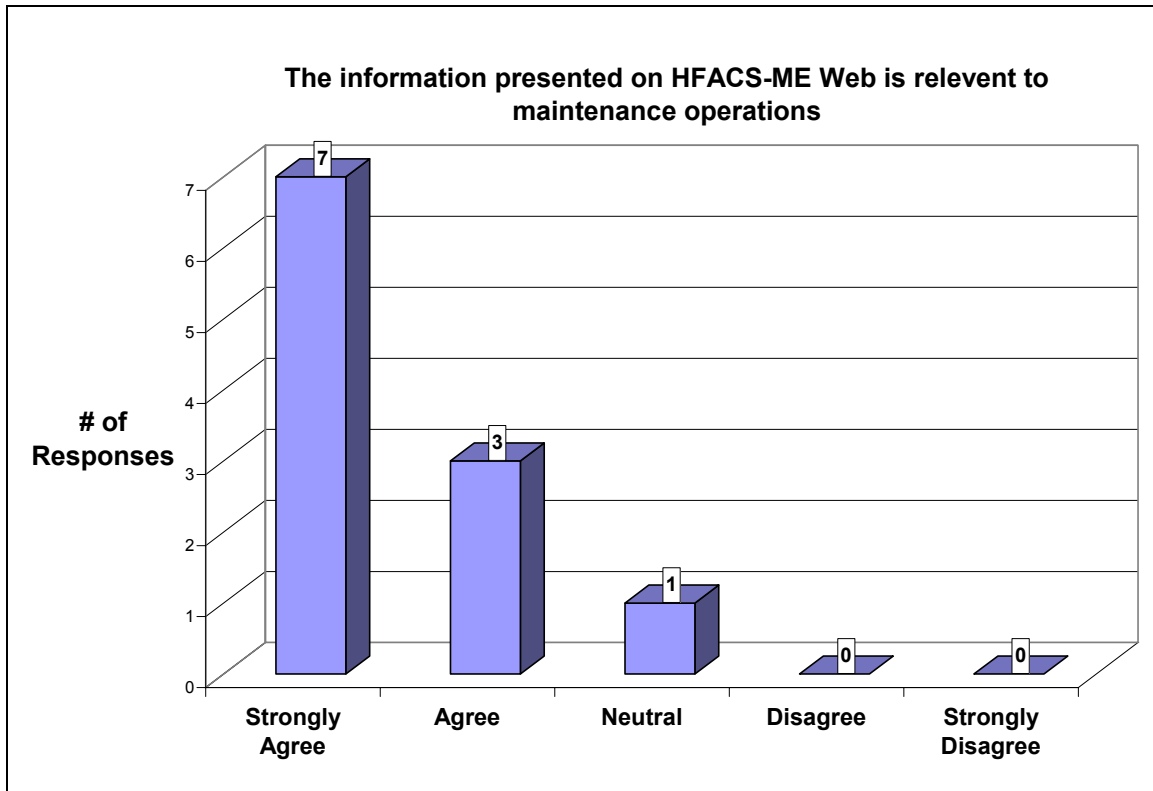


Figure 9. Exit Survey, Part II, Statement Four, Response Distribution

(e) Statement five asked whether prototype concept was a good one. The histogram of the frequency distribution for statement five is presented in Figure 10. The mean was 4.91, standard deviation = 0.289, range = 2. All participants (n = 12; 100%) indicated the concept of the prototype was a good one.

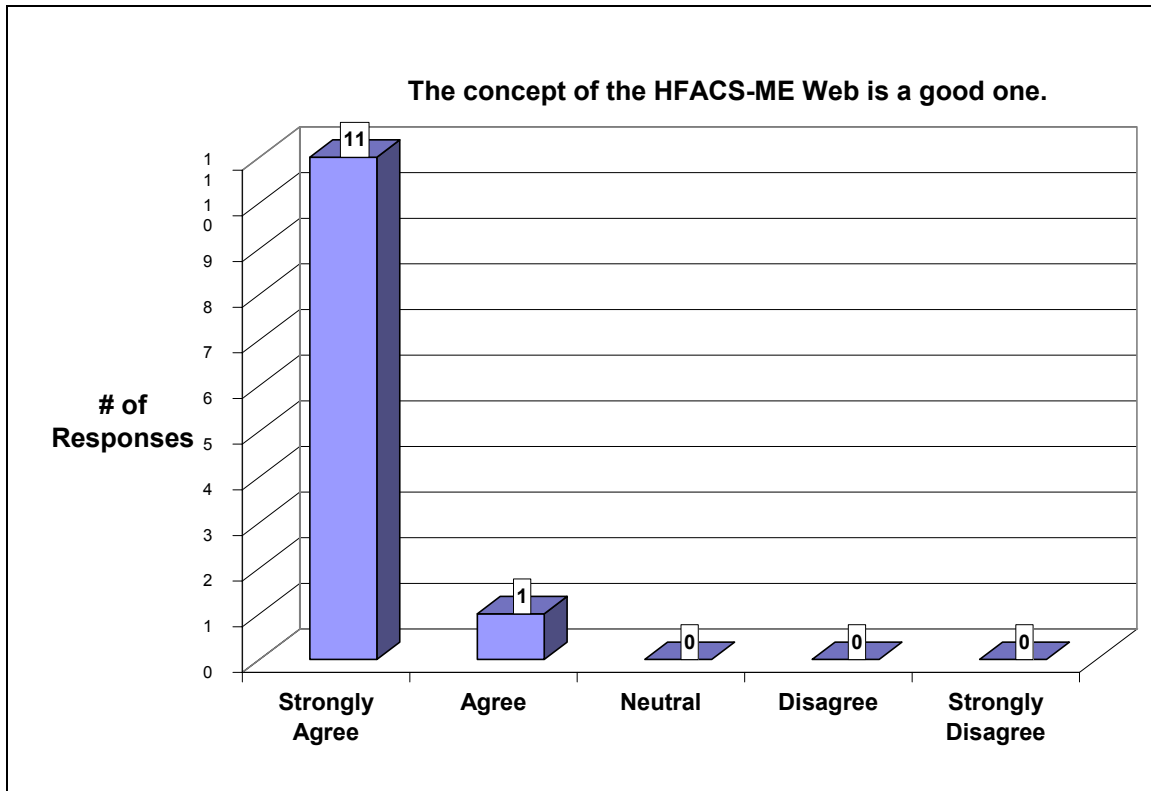


Figure 10. Exit Survey, Part II, Statement Five, Response Distribution

The final section of Part II asked participants to comment on five areas specific to the visual appeal and functionality of the HFACS–ME Web interface.

Visual appeal. Most comments were extremely positive concerning the "look" of the HFACS–ME Web interface. Some comments include:

"Great graphics, but some O-level organizations, especially dets may only have dial-in capability thus taking a long time to load pages."

"Very professional look and feel. Consistent navigation menu minimizes need to return to homepage every time."

"Everything looks great. Awesome tool."

Layout. Comments were generally positive concerning the layout of the HFACS–ME Web interface. Some comments include:

"Clean, uncluttered layout. Use of "tool tips" allows user to see more detailed menu description without overcrowding display."

"Easy to navigate"

"Very logical layout, hyperlinks quickly move one."

"Would like layout condensed to allow entire page to be visible with Favorites window open on the left side." (Authors note: This individual selected "Don't Know" to the demographic question concerning screen resolution, so it is not possible to determine an optimal size based on the information available)

"Clean, concise."

Appropriate use of colors and graphics. Comments were positive concerning the appropriate use of colors and graphics in HFACS–ME Web. Some comments include:

"Very nice colors, but you need a helo picture somewhere".

"Some of the colors in the graphs could possibly be changed. Don't really like the black and white bars."

Design consistency. Comments were extremely positive concerning the overall design and consistency of the HFACS–ME Web interface. Some comments include:

"Intuitive design make navigation easy. Consistent menu bar allows user to move between functional areas with ease."

"Great. Makes it easier to understand regardless of what you are in."

Functionality of menu items and hyperlinks. Most comments were positive regarding navigation elements incorporated in HFACS–ME Web. Some comments include:

"Good use of hyperlinks allows user to move between mishaps and reports quickly."

"Website was easy to navigate and menus are appropriately titled. Easy to use."

General comments. One participant recommended that the companion Online Tutorial be written so that someone with limited maintenance and safety knowledge can use this tool. Saying that even though the tool may be intended for safety and maintenance supervisors and managers, it could be a valuable tool in maintenance workcenter training evolutions.

2. Responses to Open-ended Questions

Part III of the exit survey contained three open-ended questions for the participants to respond to their overall satisfaction with the prototype. Every participant availed himself of this opportunity to provide constructive criticism. The responses from all 12 participants were overwhelmingly positive. Every participant indicated there was great merit in a tool such as the prototype and all of the “criticisms” were presented in a professional/positive manner. The desire of the participants was to take this prototype, in its current form, and improve it for their use in the fleet.

(a) Question one asked the participant to list the most positive aspects of the HFACS–ME Web prototype. Nine participants indicated the prototype was an excellent source of data that could be used for training, trend analysis, and decision-making. Others thought the prototype was useful to provide comparisons between variables (aircraft, mishap type, location, etc.). Some sample inputs include:

“A ton of information presented in a very logical, easy to use and understand format.”

“The HFACS–ME Web prototype allows a user to determine common mishap causal factors and prevent future ones of the same type.”

"Easy access that anyone who has a need should be able to get the info"

"The presence of hyperlinks allows the user to move quickly to the information desired."

"The ability to toggle between summaries and long descriptions allows the user to take a high level or detailed view, as appropriate"

"Easy to figure out capabilities just by looking at menus and drop boxes."

"Easy to access information. Would be ideal for research or organizing a training program."

"Current data available anywhere in the world using a Web browser."

"Quick/easy to use compared to other options...contacting the NSC for a data request."

(b) Question two asked for the most negative aspects of the HFACS–ME Web prototype. A number of problem areas of the prototype application were focused in one of three areas: HFACS–ME terminology, data consistency, and interface issues.

HFACS–ME. Four participants noted the HFACS–ME taxonomy is not a common terminology and thus found it difficult to understand. The ability to access the HFACS–ME Code descriptions from various parts of the prototype helped, but additional explanation of the each (and examples) vice a mere translation of the three-letter code would have been more beneficial to the participant. The participants felt that any eventual end-user of the prototype would need a good working knowledge of HFACS–ME in order to be able to get the most use out of the prototype; three users indicated the presence of a functional Help or Online Tutorial with an HFACS–ME breakdown would have been helpful.

Data Consistency. Two participants noted that the mishap factors data displayed were inconsistent with the aircraft type or mishap description. (Author's note: the data contained in the database was used in several iterations of MEIMS research and it is surmised that at some point during the evolution of the MEIMS tool, the factor data became "disconnected" from the mishap data. This problem has also been identified in the current MEIMS research. The HFACS–ME Web display data was verified against the current version of the database to ensure functional accuracy, but this highlights a crucial point – user confidence in the value of the application is easily shaken if the underlying data is not accurate.)

Interface. Three participants commented that the need to repetitively enter the same query data on several pages was frustrating. One user questioned why direct data entry was not possible as opposed to drop-down boxes. His frustration stems from the fact that his aircraft model was near the bottom of the alphabetical list of aircraft types, and scrolling down each time was tedious (Author's note: the reason for using drop-down boxes is two-fold. First, the items listed in the drop-down box are retrieved from the mishap database when the page loads, thereby ensuring that only items represented in the database are listed and that every item listed will have at least one corresponding entry in the database. This prevents the user from querying the database for items which have no matches. The second reason has to do with error control. By

requiring the user to select from a list of validated and properly formatted entries, the developer is not required to design complex error trapping routines. Without the protection afforded by the drop-down boxes a user might input any one of the following in an effort to retrieve mishap data for the P3 Orion: P-3, P3C, P-3C UIII, Orion, etc., none of which would return any data.)

Other “negatives”.

Navigation issues were minor, limited to suggestions for improved access between pages (being able to go directly from one page to another without having to back out of previously selected pages (two participant inputs) and the need to avoid scrolling to see all data (Two users, both who viewed the prototype with a screen resolution of 800 by 600 pixels).

In some instances the three-dimensional graphs in the front, hide data in the back, requiring the user to back out and select "Display Graph and Data Table". One participant noted difficulty reading the x-axis labels (fiscal years) on the Graph display. Two users commented on the color selection for the graph column. One mentioned that the black and white colored columns were distracting. A second participant also identified the black column as being too dark.

(c) Question three asked for suggested changes to the prototype. The participants brought out several key points critical for inclusion in future versions of HFACS–ME Web. Most of the suggestions related directly to one or more of the previously mentioned “negatives.” Three comments were made about improving the ability for the end-user to understand HFACS–ME through either improved HFACS definitions within the prototype, additional Help/Online Tutorial, and formal training for all end-users. One participant also made a suggestion to improve the interface and navigation of the prototype to increase usability (e.g., adding additional methods to view HFACS–ME definitions and better descriptions of Levels 1, 2, and 3).

Although not a part of the current HFACS–ME Web design, two participants envisioned the use of HFACS–ME Web as a means of submitting mishap data in the future. Both participants made statements to the effect that the tool could be a resource to allow safety personnel to enter data that would form the core of a mishap report. Both comments also acknowledged the need to incorporate the "chop chain" before publishing

the data (Author's note: this issue was partially addressed in the current version of the MEIMs tool (see Nelson, 2001 and Flanders & Tufts, 2001)).

One participant suggested that the HFACS–ME Web interface be modified so that the same website would allow the user to access both maintenance and aircrew error data and analysis

Other inputs:

Increasing the size of the database by using mishaps prior to 1989 and adding hazard reports was felt to be a means of improving the depth of the data (three participants).

Investigation of applicable data-mining techniques for future incorporation into interface.

Add Distance Learning module and an in-depth online tutorial.

Some specific changes to the actual interface were also suggested (e.g., increasing drop-down box size in order to view all of the available options, a better method to show aircraft model to prevent confusion by adding the nickname to the model number: F14 Tomcat, P3 Orion; being able to filter database data, using the same method as in the Data Selection page, before producing a report.

VI. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

A. SUMMARY

Naval Aviation has determined to reduce its mishap rate. The reduction of human error involved in maintenance related mishaps would be one step in achieving that goal; now it has to find appropriate tools to accomplish this. The Human Factors Analysis and Classification System–Maintenance Extension (HFACS–ME) is a taxonomy which covers maintenance operations and falls in line with the Naval Aviation Safety Program’s notion of multiple causal factors, the idea of sequential events leading to an event, and several established human factors theories. HFACS–ME was successfully used to examine human error in mishaps and incidents. The prototype HFACS–ME Web prototype is a safety information management system based on the HFACS–ME taxonomy used to facilitate the characterization and analysis of human error in Naval Aviation maintenance mishaps. Tools such as a refined version of HFACS–ME Web will provide assistance in identifying human error patterns and facilitate intervention development. The significant strength of HFACS–ME Web is that it capitalizes on existing Web browser technology and the Internet as a communications medium to link geographically distributed users to a centrally managed safety database. This allows users to view and analyze current mishap data anytime, from anywhere, as long as they have a means to connect to the Internet. With the knowledge gained from this tool, safety and maintenance personnel can examine their own practices and devise training and interventions that may break the mishap chain.

B. CONCLUSIONS

The participants’ overall satisfaction of the HFACS–ME Web prototype indicated there is a need to provide access to mishap data information for use in training, analysis, and investigations. Participant feedback demonstrated the concept of HFACS–ME Web to be sound and its tie-in with maintenance operations readily apparent. However, the prototype requires some adjustment before it can be deployed for Fleet use. For HFACS–ME Web to be able bridge the knowledge gap between the Naval Safety Center and Fleet safety and maintenance personnel, the following issues must be resolved:

User's lack of familiarity with HFACS–ME taxonomy and associated terminology. Help pages, Frequently Asked Question pages and the planned Online Tutorial need to provide sufficient understanding of taxonomy to enable the user to operate the HFACS–ME Web.

Database accuracy. As discussed previously, the current HFACS–ME prototype database contains errors in the tblMishapFactors table. These errors cause incorrect mishap factors to be displayed with each mishap.

Security. Given the privileged nature of mishap data, the security aspects of the HFACS–ME Web need to be addressed. Specifically, a more scalable access control mechanism needs to be implemented and enforced, and consideration should be given to implementing secure socket layer for encryption of in-transit data.

Also several minor shortfalls need to be refined:

Modify (or replace) graph code to address issues related to improper x-axis label depiction and graph column color selection (delete black). Also improve graph scaling function to account for browser window size vice screen resolution.

Providing solutions to these identified shortcomings will improve the usability of future versions of HFACS–ME Web; and subsequently the opportunity for it to be a factor in reducing the aviation mishap rate is enhanced.

C. RECOMMENDATIONS

The following is a list of recommended improvements, that would improve functionality, ease-of-use, security and performance of the HFACS–ME Web prototype:

1. Interface Improvements

- A Web development expert should participate in the fine-tuning of HFACS–ME Web interface options to ensure HTML and ASP code is efficient.
- Consider adding client-side script to dynamically populate drop-down boxes such that the selection in one box dynamically changes the available options in other boxes (e.g., selecting Aircraft Type = P3 changes options displayed in Location drop-down box to disable or

remove Embarked from the list of available locations since there are no Embarked P3 mishaps in the database, etc.)

- Investigate performance and scalability implications of incorporating Session variables to store Data Selection criteria in Web server memory. If feasible, this change would allow users to enter selection criteria one time and move from one display format to another without having to re-enter the selection criteria (e.g., the user makes criteria selections after clicking on Mishap Data menu item, once the data is viewed in tabular form, they can select Graph to display the same group of mishaps in a three-dimensional graph, and finally select Report to generate a breakdown of the HFACS–ME distribution for the data, all without re-entering the criteria, etc.)
- Add aircraft NATO name in addition to type/model to drop-down lists to prevent selection of incorrect aircraft type (e.g., F14 Tomcat, P3 Orion, etc.).
- Modify the Factors Analysis page so that when the user double-clicks on a table cell, the mishaps represented by that factor are displayed (i.e., if the cell "Inadequate Documentation" indicates six mishaps within the dataset contain that third level factor, then by double-clicking on that cell, the user should see the Mishap Data for those six records, etc.)
- Investigate Hierarchical Recordset capabilities introduced in ADO 2.0 to display records with drill-down capabilities.
- Modify Mishap Data page to allow users to sort returned mishap data by clicking on column label (i.e., clicking on Mishap Date will re-sort data by date, clicking Type will re-sort data by mishap type, etc.). Clicking column headings would alternate between ascending and descending sort order.
- Add a chronological report type that formats the data returned by the user's criteria in a Chronological Mishap report, displaying mishap

details and related factor data. Allow the user to toggle between summary and long descriptions in report body.

- Consider adding a "Top Ten List" display or report option to create a list of the most prevalent 3rd Level Factors in a given dataset. (i.e., for all H60 mishaps the most common factor is Maintainer Infraction with 12 mishaps, next is Inadequate Documentation with 8 mishaps, etc.) This will allow the user to quickly identify areas for potential intervention without having to manually search and order the analysis data.
- Arrange data in three-dimensional graphs so that the fields with the largest numbers are put in the rear rows and scaled down to the front so that no data is hidden to the end-user.
- Modify HFACS–Web design to incorporate Extensible Markup Language (XML). XML is a text-based meta-language that uses tags, elements, and attributes to add structure and definition to documents. It is similar to HTML in syntax and implementation, but different with regard to functionality. Where HTML allows users to control how documents are displayed, XML allows them to describe the actual contents of the documents. It is a markup language because it uses tags to mark-up documents and it is a meta-language because it uses these tags to give structure to documents that it in turn uses as a means of communication. XML is extensible because it enables users to create their own collection of tags (unlike HTML).
- Investigate use of third party software application to generate Web reports and graphs. Current implementation is hand-coded and provides limited functionality. Commercially produced software applications (e.g., Crystal Reports, Cognos, etc.) offer significant flexibility and may enable users to customize graph and report outputs as well as provide dynamic drill-down and advanced data analysis capabilities.

- Pursue development of an Investigation module similar to that incorporated in current MEIMS tool, to allow safety personnel to generate preliminary safety reports. Module design would walk user through HFACS–ME taxonomy to ensure all relevant causal factors are identified and documented.
- Add cautionary note to Home page indicating the use of HFACS–ME Web is not intended to take the place of the rigorous data analysis techniques used by the Naval safety Center. Also note that inferences from a small sample of data may not be consistent across the whole population., and therefore should not be the sole basis for committing significant resources to a particular perceived problem.
- Investigate potential to generate, staff and distribute safety reports. Review available collaborative technologies to determine suitability for digital "chop" and endorsement of safety reports.

2. HFACS–ME Taxonomy

- Incorporate improved HFACS–ME definitions and examples within HFACS–ME Web by ensuring access to the definition page are available on every page (incorporating an additional hyperlink on the menu bar would offer the simplest solution). Better descriptions of the HFACS–ME acronyms would also improve usability and understanding.
- Incorporate planned Online Tutorial and Help pages to improve the end-users knowledge and understanding of the HFACS–ME taxonomy. Because the HFACS–ME taxonomy is part of the Navy's aviation safety instruction and is not (yet) included in any maintenance instruction, the planned Online Tutorial and Help pages are likely to be crucial to taxonomy understanding for non-safety personnel. Both should be designed with the assumption that the user has little or no previous experience with the HFACS–ME taxonomy.

3. Performance

- Investigate the potential performance enhancement achieved by caching static drop-down box data in Web server memory using Application variables vice querying database every time a page is loaded.
- Conduct database log file analysis to identify performance bottlenecks. Investigate the use of table indexes as a means to improve server performance. To achieve significant performance additional hardware resources may be required or performance in other areas may be adversely affected.

4. Security Improvements

- Incorporate User ID/Password lookup from secured database table vice assigning each authorized user an NT account on the Web server. This should greatly simplify management of the Web server and improve application scalability.
- Incorporate multilevel permissions to limit data access and to prevent casual users from performing administrative functions on the Web and database servers.
- Incorporate Secure Socket Layer encryption in data transfer between Web server and client browser. This will require creation of an HFACS–ME Web certificate (preferably signed by the Navy Root Certificate Authority) that can be distributed to authorized users along with their login credentials.
- Examine authentication and encryption options that become available once the Navy Common Access Card and Public Key Infrastructure (PKI) are implemented.

5. Database

- Correct problems relating to database data mismatch and add mishap data from 1999 to present. Once database is up-to-date, devise

automatic or manual method to incorporate new mishap data in near real time.

- Investigate use of data mining techniques on HFACS–ME data to identify factor correlations not readily apparent by visual inspection.
- The current HFACS–ME Web development stage dictates that HFACS–ME Web/MEIMS researchers at the Naval Postgraduate School and the School of Aviation Safety administer the database. However, moving the database under the cognizance of the Naval Safety Center once HFACS–ME Web and MEIMS become "operational" would seem appropriate. To accomplish this migration, a thorough review of database compatibility with existing and planned NSC databases is required and a data migration/transformation plan is necessary to ensure the business logic will transfer
- Keep "business logic" in stored procedures and views separate from those used by MEIMS application. This precaution will ensure that a change in one application will not have a detrimental effect on the other. Changes to system procedures should be tested and verified off-line. Once validated, the new or revised procedures should be added to the database via scripts after archiving the old procedures.
- Ensure modifications to the database schema are documented and tested by both HFACS–ME Web and MEIMS developers to resolve potential conflicts before the changes are made to the operational database.
- Include mishaps prior to 1989 and all hazard reports to improve the depth of the database.
- Incorporate annual flight hour data as a way to normalize data output. By generating appropriate mishap *rates* (i.e., F18 mishaps per 100,000 flight hours, etc.) a meaningful comparison can be made between categories. This will show relative weight, vice always being more heavily weighted for aircraft types with a larger inventory (FA-18, H-46, etc.).

6. The Future of HFACS–ME Web

By capitalizing on the familiarity and convenience of the Internet and the scalability, performance and security of the three-tier client/server architecture, the HFACS–ME Web prototype has the potential to allow authorized users unprecedented access to safety data and analysis. This "self-service" application will allow users to query and analyze maintenance mishap data, 24 hours a day, 7 days a week, regardless of location. Not only will this reduce the burden on analysts and technicians at the Naval Safety Center, it will enable safety and maintenance personnel at all levels to look for areas to focus training and mishap intervention techniques. It will heighten awareness of the most prevalent mishap factors affecting a particular aviation maintenance community or across the aviation industry. By expanding HFACS–ME Web to encompass all military services or civilian aviation organizations, managers might gain insight into systemic problems or they may identify model organizations to be emulated based on their low maintenance mishap rates. Although human error in aviation maintenance may never be completely eliminated, HFACS–ME Web may enable organizations to move closer to that elusive goal.

APPENDIX A. HFACS–ME DATABASE SCHEMA

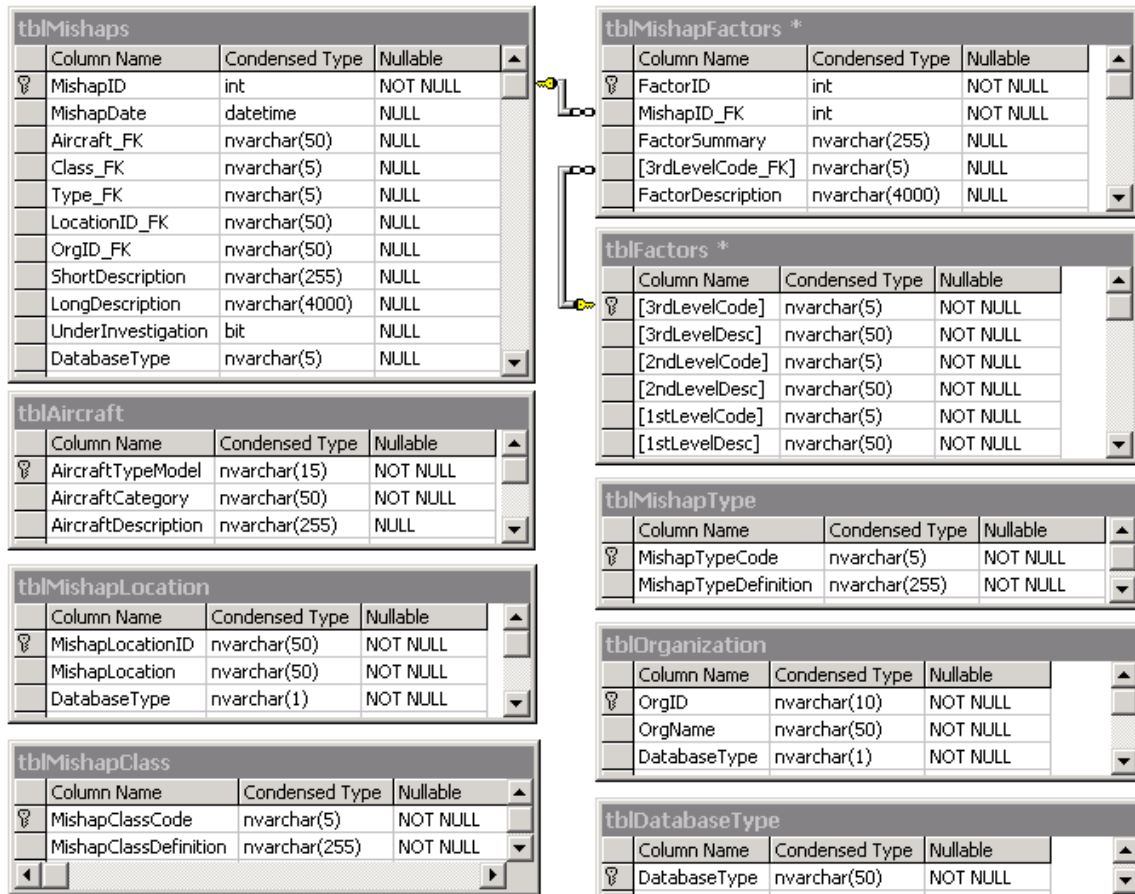


Figure A-1. HFACS–ME Database Schema

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. HFACS–ME WEBSITE SCREEN SHOTS

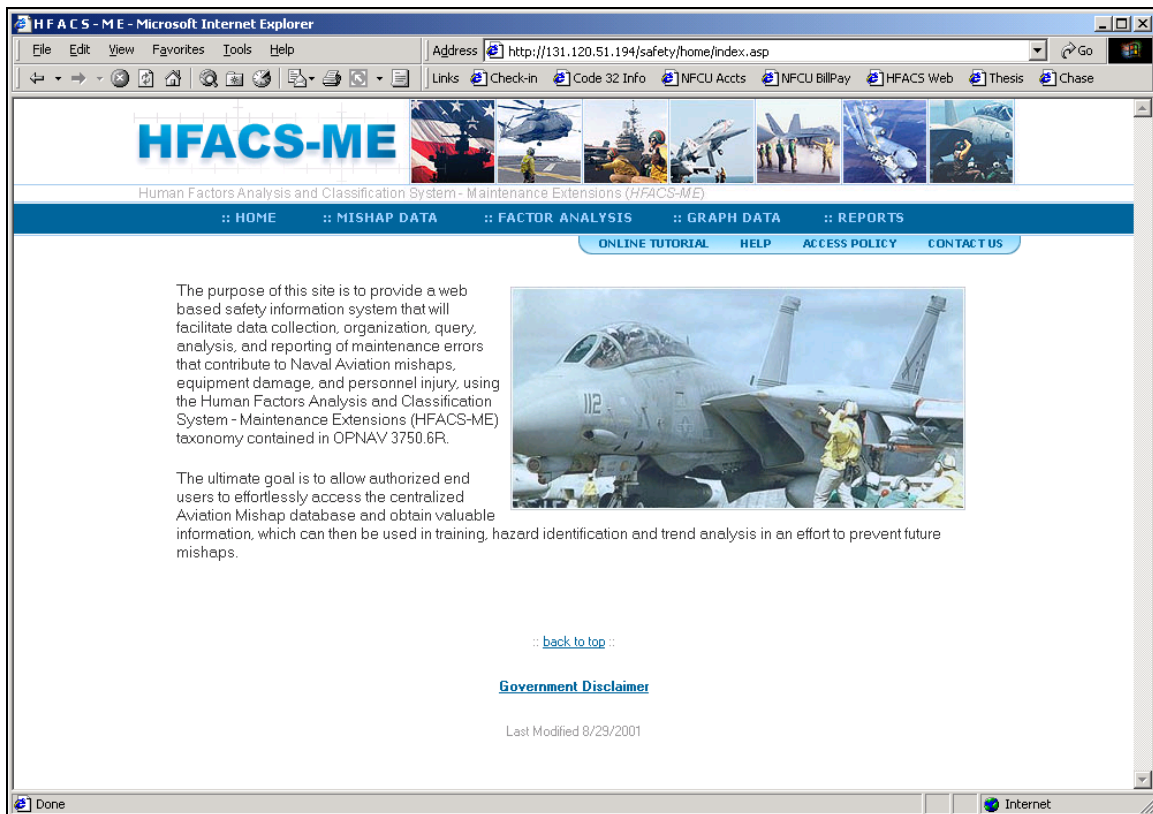


Figure B-1. HFACS–ME Web Homepage

Table view - Microsoft Internet Explorer

Address: http://131.120.51.194/safety/mishap/query.asp

Links: Check-in Code 32 Info NFCU Accts NFCU BillPay HFACS Web Thesis Chase

HFACS-ME

Human Factors Analysis and Classification System- Maintenance Extensions (HFACS-ME)

[HOME](#)
[MISHAP DATA](#)
[FACTOR ANALYSIS](#)
[GRAPH DATA](#)
[REPORTS](#)

[ONLINE TUTORIAL](#)
[HELP](#)
[ACCESS POLICY](#)
[CONTACT US](#)

Mishap Data Selection Page

Use the drop down boxes below to select the desired Mishap Criteria and Factors Criteria to include in the dataset. Multiple items within a particular criteria (ie. Aircraft Type = F14, F18) may be selected by holding down the Ctrl button and left clicking on the desired items. This will result in a query that will return data matching ANY of the selected items.

If criteria are selected in several criteria boxes (ie. Aircraft type and Mishap Class), the resulting dataset will be only those records that match ALL criteria. For example, if the user selects **Aircraft Type=F18, F14** and **Mishap Class=A**, then the database will return all records involving either a F14 or F18 and resulted in a Class A mishap.

Aircraft Type	Mishap Type	Mishap Class	Location	Service	Fiscal Year
(All)	(All)	(All)	(All)	(All)	(All)
A4	FM	A	Ashore	USN	1990
AV8	FRM	B	Embarked	USMC	1991
C12	AGM	C	Detached	UNK	1992

1st Level Factor	2nd Level Factor	3rd Level Factor
(All)	(All)	(All)
Maintainer Acts	Crew Coordination	Adverse Mental State
Maintainer Conditions	Environment	Adverse Physical State
Management Conditions	Equipment	Attention/Memory

[back to top](#)

Done Internet

Figure B-2. Data selection Page

Table view - Microsoft Internet Explorer

Address <http://131.120.51.194/safety/mishap/query.asp>

Links [Check-in](#) [Code 32 Info](#) [NFCU Accts](#) [NFCU BillPay](#) [HFACS Web](#) [Thesis](#) [Chase](#)

595 Mishaps matched Computer and Information Programs Curricular Office
<http://www.nps.navy.mil/code32/>

[Click on Mishap ID to view detailed Mishap Factor data](#)

Mishap ID	Aircraft	Type	Class	Location	Service	Date
1	T34	FM	C	ASH	USN	1/10/1994
2	F14	FM	C	ASH	USN	1/28/1994
3	F14	AGM	C	ASH	USN	3/23/1994
4	F14	FM	C	ASH	USN	6/15/1994
5	F14	FM	A	ASH	USN	8/26/1994
6	F14	AGM	C	ASH	USN	10/14/1994
7	F14	AGM	C	ASH	USN	10/26/1994
8	F14	AGM	B	ASH	USN	10/6/1995
9	F14	AGM	A	ASH	USN	10/11/1995
10	E2	AGM	C	EMB	USN	10/27/1999
11	E2	AGM	C	EMB	USN	11/25/1990
12	E2	FM	A	EMB	USN	7/31/1992
13	E2	AGM	C	EMB	USN	6/9/1994
14	E2	FM	C	ASH	USN	1/14/1990
15	E2	FM	C	ASH	USN	4/23/1990
16	E2	AGM	C	ASH	USN	6/29/1990
17	E2	AGM	C	ASH	USN	3/9/1993
18	E2	FM	C	ASH	USN	12/30/1993
19	E2	FM	C	ASH	USN	5/15/1995
20	E2	AGM	C	ASH	USN	9/11/1995
21	E2	AGM	C	ASH	USN	10/15/1995
22	E2	AGM	B	ASH	USN	1/6/1996
23	AV8	FM	C	ASH	USMC	8/3/1990
24	AV8	FM	A	ASH	USMC	8/8/1990
25	AV8	FM	C	ASH	USMC	1/6/1991
26	AV8	AGM	B	ASH	USMC	5/9/1991
27	AV8	FM	C	ASH	USMC	6/10/1991
28	AV8	FM	B	ASH	USMC	8/31/1991
29	AV8	FM	A	ASH	USMC	3/6/1992
30	AV8	FRM	C	ASH	USMC	7/16/1992
31	AV8	FRM	C	ASH	USMC	12/8/1992
32	AV8	AGM	C	ASH	USMC	2/28/1993
33	AV8	AGM	C	ASH	USMC	4/1/1993
34	AV8	FM	B	ASH	USMC	4/6/1993

Done Internet

Figure B-3. Mishap Query Results Page

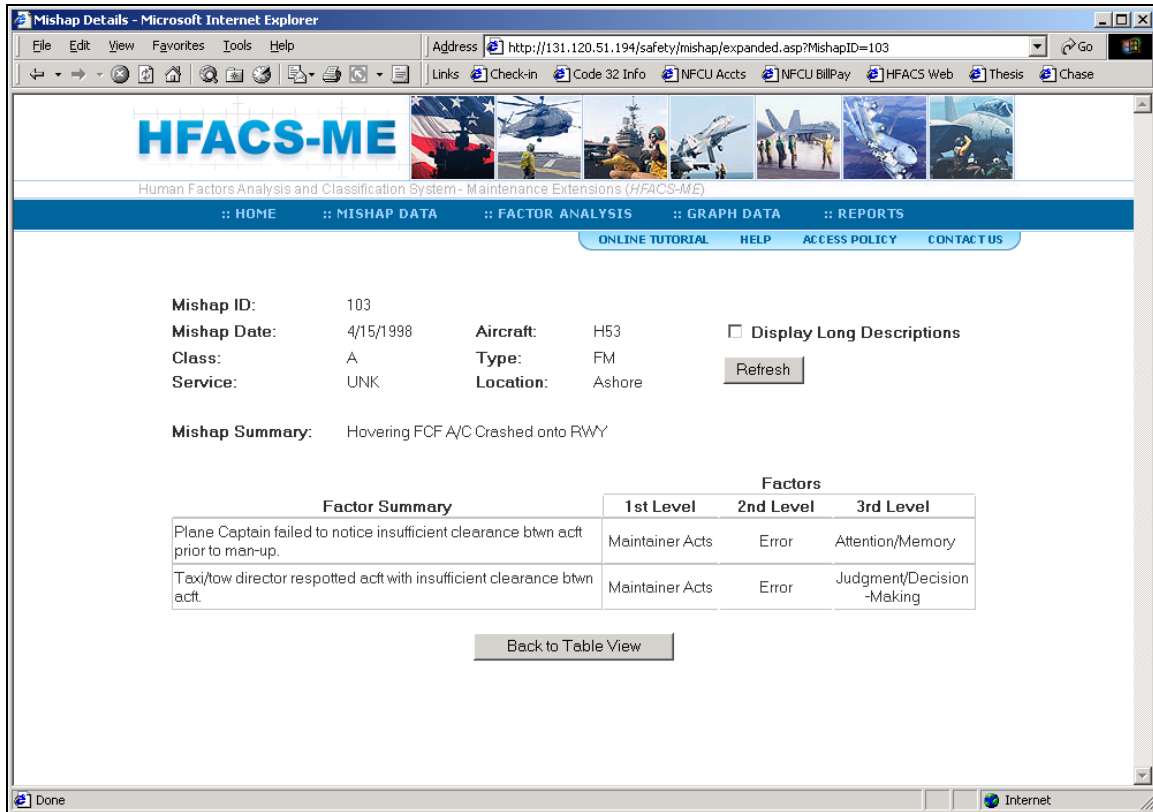


Figure B-4. Mishap Expanded Details Page

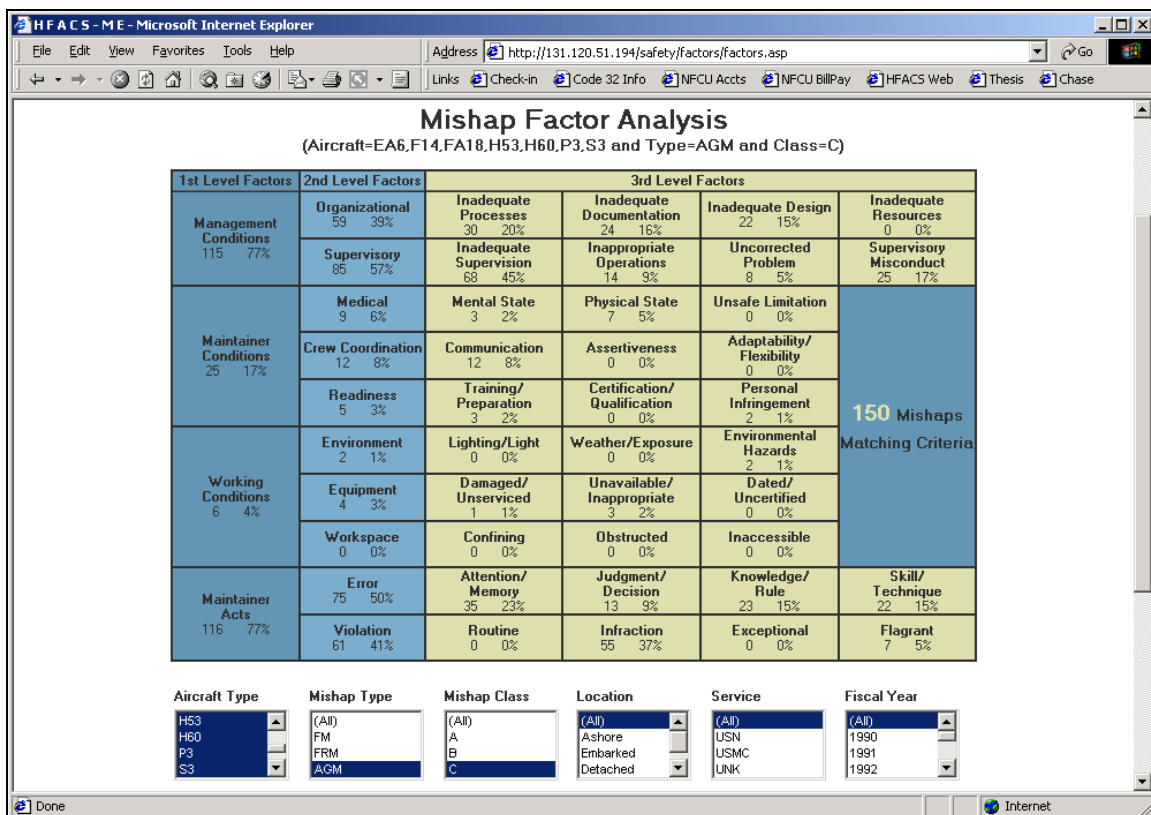


Figure B-5. Mishap Factor Analysis Results Page

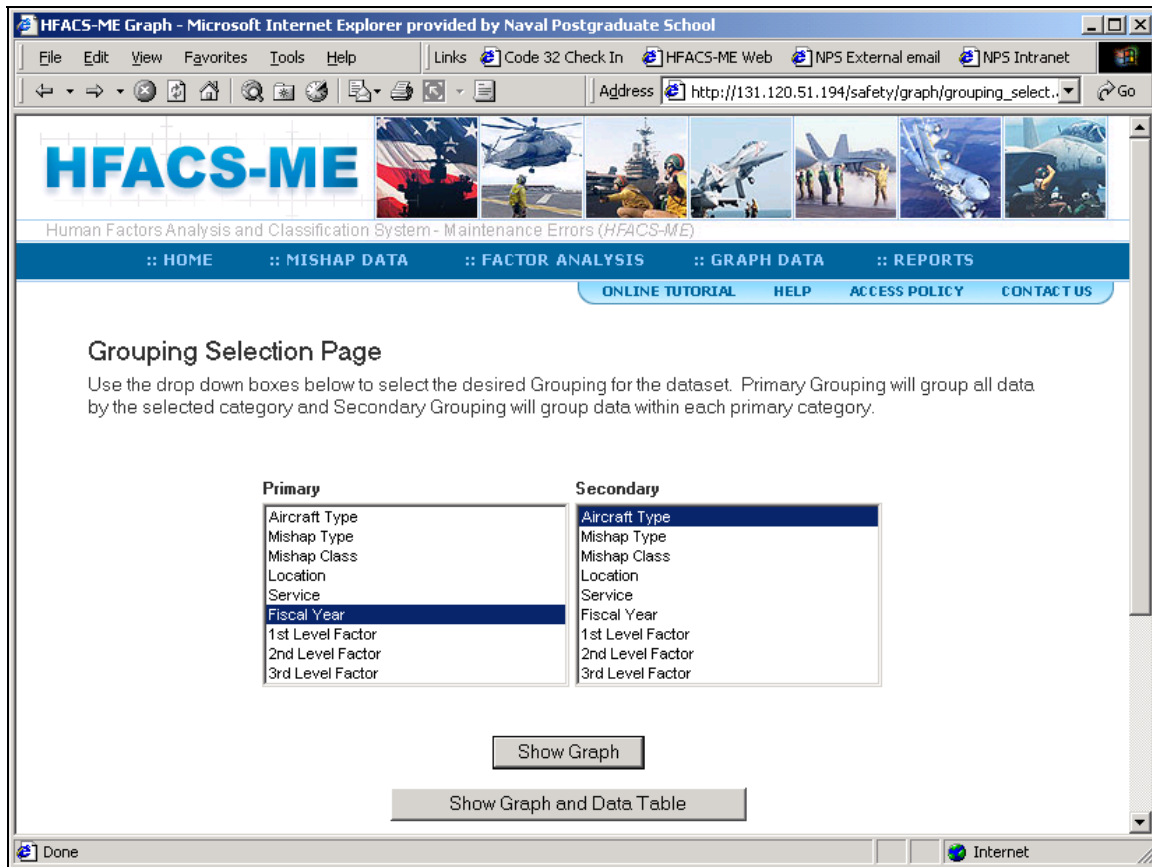


Figure B-6. Graph Axis Selection Page

Mishap Table - Microsoft Internet E...					
<u>Mishap Counts by Aircraft and Fiscal Year</u>					
LEGEND	Aircraft				
Fiscal Year	EA6	F14	H60	P3	S3
■ 1990	3	15	16	3	2
■ 1991	11	28	14	16	5
■ 1992	6	32	8	18	6
□ 1993	16	12	12	23	12
■ 1994	14	20	0	17	4
■ 1995	16	7	3	9	6
■ 1996	8	11	2	7	0
■ 1997	6	18	5	2	7
■ 1998	3	6	10	6	0
■ 1999	6	12	6	12	9

Figure B-7. Graph Data Table

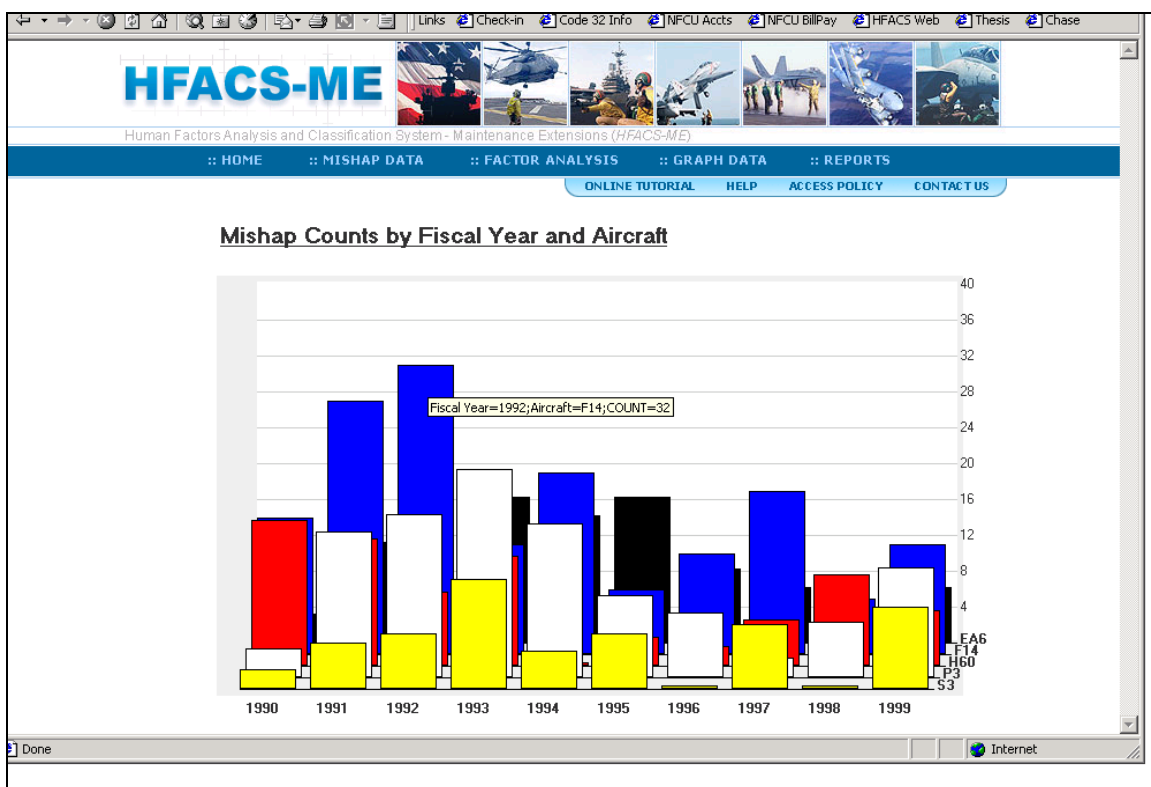


Figure B-8. Graph Results Page

HFACS-ME Taxonomy

1 st Order	1st Code	2 nd Order	2nd Code	3 rd Order	3rd Code
Management Conditions	MG	Organizational	ORG	Inadequate Processes	PRO
				Inadequate Documentation	DOC
				Inadequate Design	DES
				Inadequate Resources	RES
		Supervisory	SUP	Inadequate Supervision	IDQ
				Inappropriate Operations	OPS
				Uncorrected Problem	PRB
				Supervisory Misconduct	MIS
Maintainer Conditions	MC	Medical	MED	Adverse Mental State	MNT
				Adverse Physical State	PHY
				Unsafe Limitation	LIM
		Crew Coordination	CRW	Inadequate Communication	COM
				Inadequate Assertiveness	ASS
				Inadequate Adaptability/Flexibility	ADA
		Readiness	RDY	Inadequate Training/Preparation	TRG
				Certification/Qualification	CRT
Personnel Readiness Infringement	INF				
Working Conditions	WC	Environment	ENV	Inadequate Lighting/Light	LGT
				Unsafe Weather/Exposure	WXE
				Unsafe Environmental Hazards	EHZ
		Equipment	EQP	Damaged/Unserviced	DMG
				Unavailable/Inappropriate	UNA
				Dated/Uncertified	DUC
		Workspace	WRK	Confining	CON
				Obstructed	OBS
Inaccessible	INA				
Maintainer Acts	MA	Error	ERR	Attention/Memory	ATT
				Judgment/Decision	JDG
				Knowledge/Rule Based	KNW
				Skill/Technique	SKL
		Violation	VIO	Routine	ROU
				Infraction	IFC
				Exceptional	EXC
				Flagrant	FLG

Figure B-9. HFACS–ME Taxonomy Legend

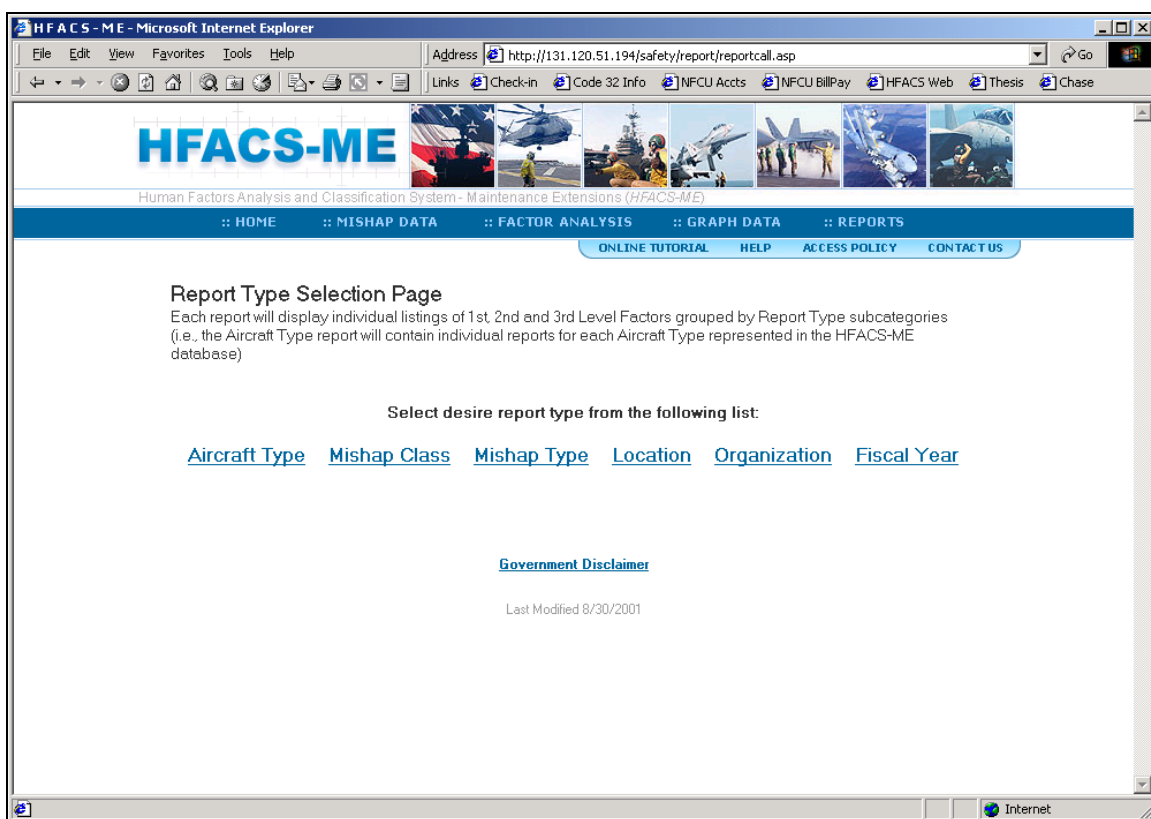


Figure B-10. Report Selection Page

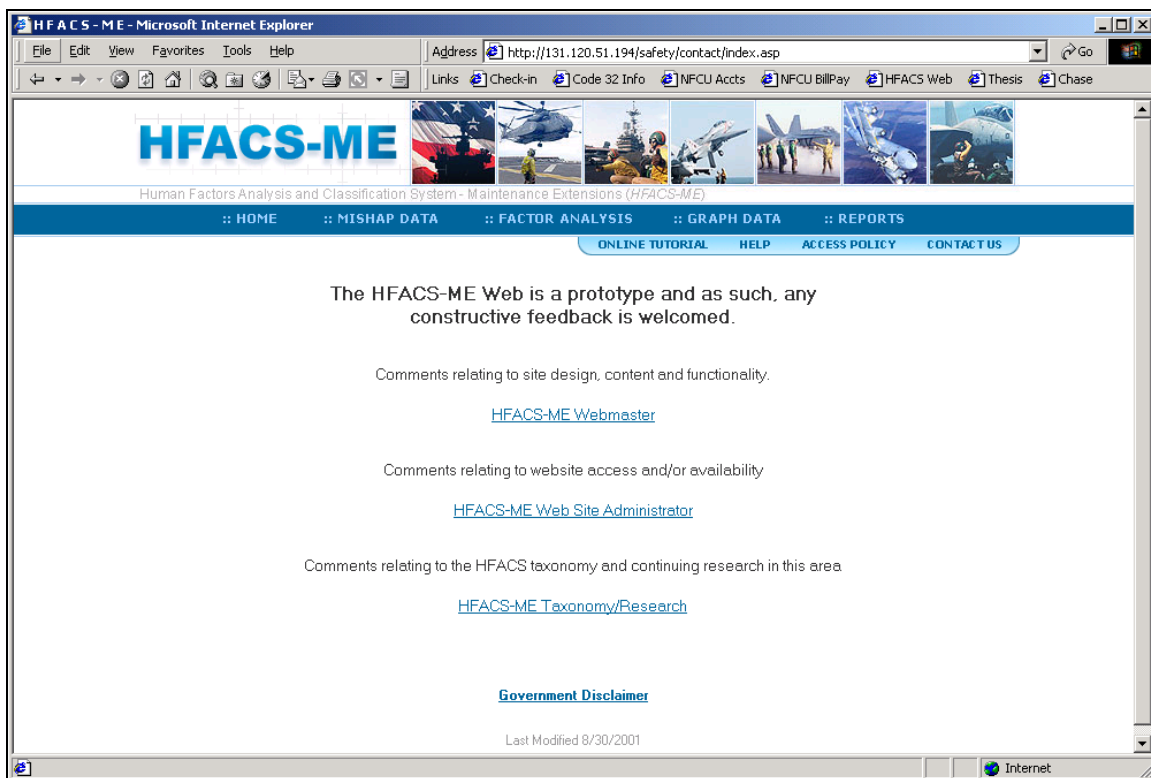


Figure B-12. HFACS–ME Contact Page

APPENDIX C. HTML AND ASP SOURCE CODE

This Appendix contains code found in the HFACS–ME website. All code is contained in HTML and ASP pages, written using HTML, Microsoft Visual Basic Script, and Javascript.

COMMON SUBDIRECTORY

connection.asp

Connection.asp is used as a server-side Include file throughout the HFACS–ME Web site. It provides a single connection definition for all database access required. Note: Password obscured in connection string for security reasons.

```
<%  
Dim conn  
Dim aConnectionString  
  
aConnectionString = "Provider=SQLOLEDB; Source=SEVEN;  
Database=HFACS;UID=sa;PWD=*****;"  
  
Set conn = Server.CreateObject("ADODB.Connection")  
conn.Mode = adModeReadWrite  
conn.ConnectionString = aConnectionString  
conn.open  
  
%>
```

criteria_dropdown_INC.asp

Criteria_dropdown_INC.asp is used as a server-side Include file throughout the HFACS–ME Web site. The nine drop-down boxes are used throughout the website to allow the user to selectively filter the data used for various display formats. The code below queries database for valid drop-down box values for six of nine drop-down boxes, then creates drop-down boxes. Data for the following drop-down boxes are dynamically generated: Aircraft Type, Organization, Mishap Year , 1st Level factor, 2nd Level factor, and 3rd Level factor. The remaining three boxes are hard coded given they are less likely to have additional items added, they are: Mishap Type, Mishap Class, and Location.

```
<%
Dim rsAC      'recordset for Aircraft Type drop-down
Dim rsOrg     'recordset for Organization drop-down
Dim rsYear    'recordset for Mishap Year drop-down
Dim rs1st     'recordset for 1st Level Factor drop-down
Dim rs2nd     'recordset for 2nd Level Factor drop-down
Dim rs3rd     'recordset for 3rd Level Factor drop-down

'Get data to populate Aircraft drop-down box
cmd.CommandText = "spAircraft_List"
Set rsAC = cmd.Execute

'Get data to populate Service drop-down box
cmd.CommandText = "spOrg_List"
Set rsOrg = cmd.Execute

'Get data to populate Mishap Year drop-down box
cmd.CommandText = "spMishap_Year"
Set rsYear = cmd.Execute

'Get data to populate 1st Level Factor drop-down box
cmd.CommandText = "sp1st_Level_Factors_List"
Set rs1st = cmd.Execute

'Get data to populate 2nd Level Factor drop-down box
cmd.CommandText = "sp2nd_Level_Factors_List"
Set rs2nd = cmd.Execute

'Get data to populate 3rd Level Factor drop-down box
cmd.CommandText = "sp3rd_Level_Factors_List"
Set rs3rd = cmd.Execute
%>

<div align="center">
  <center>

<table width="700">

  <tr>
    <td nowrap align="left" width="109" height="17"><b><font size="1">Aircraft
Type</font></b></td>
    <td nowrap align="left" width="109" height="17"><b><font size="1">Mishap
Type</font></b></td>
    <td nowrap align="left" width="108" height="17"><b><font size="1">Mishap
Class</font></b></td>
```

```

        <td nowrap align="left" width="110" height="17"><b><font size="1">
Location</font></b></td>
        <td nowrap align="left" width="109" height="17"><b><font size="1">
Service</font></b></td>
        <td nowrap align="left" width="109" height="17"><b><font size="1">Fiscal
Year</font></b></td>
    </tr>
    <tr>
        <td nowrap width="109" valign="top"><font size="1">
            <select size="4" name="cboAircraft" class=select font face= "Arial" multiple
style="width:100">
                <option <% If (Request.Form("cboAircraft")="") or
(Request.Form("cboAircraft")=", ") Then Response.Write("selected")%>
value="">(All)</option>
                <%
                'continue until we get to end of recordset
                Do While Not rsAC.EOF
                'for each option create an option tag with a corresponding value
                %>
                <option font size="1"><% If instr(Request.Form("cboAircraft"),
rsAC.Fields("ACType"))<0 Then Response.Write("selected")%>
value='<%=rsAC.Fields("ACType")%>'><%=rsAC.Fields("ACType")%></option>
                <%
                'get next record
                rsAC.Movenext
                Loop
                rsAC.close
                %>
            </select></font>
        </td>
        <td nowrap width="109" valign="top"><font size="1">
            <select size="4" name="cboType" class=select multiple style="width:100">
                <option <% If (Request.Form("cboType")="") or (Request.Form("cboType")=", ")
Then Response.Write("selected")%> value="">(All)</option>
                <option <% If instr(Request.Form("cboType"), "FM")<0 Then
Response.Write("selected")%> value="FM">FM</option>
                <option <% If instr(Request.Form("cboType"), "FRM")<0 Then
Response.Write("selected")%> value="FRM">FRM</option>
                <option <% If instr(Request.Form("cboType"), "AGM")<0 Then
Response.Write("selected")%> value="AGM" >AGM</option>
            </select></font>
        </td>
        <td nowrap width="108" valign="top"><font size="1">
            <select size="4" name="cboClass" class=select multiple style="width:100">
                <option <% If (Request.Form("cboClass")="") or (Request.Form("cboClass")=", ")
Then Response.Write("selected")%> value="">(All)</option>
                <option <% If instr(Request.Form("cboClass"), "A")<0 Then
Response.Write("selected")%> value="A">A</option>
                <option <% If instr(Request.Form("cboClass"), "B")<0 Then
Response.Write("selected")%> value="B">B</option>
                <option <% If instr(Request.Form("cboClass"), "C")<0 Then
Response.Write("selected")%> value="C">C</option>
            </select></font>
        </td>
        <td nowrap width="110" valign="top"><font size="1">
            <select size="4" name="cboLocation" class=select multiple style="width:100">
                <option <% If (Request.Form("cboLocation")="") or
(Request.Form("cboLocation")=", ") Then Response.Write("selected")%>
value="">(All)</option>

```

```

        <option <% If instr(Request.Form("cboLocation"), "ASH")<0 Then
Response.Write("selected")%> value="ASH">Ashore</option>
        <option <% If instr(Request.Form("cboLocation"), "EMB")<0 Then
Response.Write("selected")%> value="EMB">Embarked</option>
        <option <% If instr(Request.Form("cboLocation"), "DET")<0 Then
Response.Write("selected")%> value="DET">Detached</option>
        <option <% If instr(Request.Form("cboLocation"), "UNK")<0 Then
Response.Write("selected")%> value="UNK">Unknown</option>
        </select></font>
    </td>
    <td nowrap width="109" valign="top"><font size="1">
        <select size="4" name="cboService" class="select multiple style="width:100">
        <option <% If (Request.Form("cboService")="") or
(Request.Form("cboService")=" ", " ) Then Response.Write("selected")%>
value="">(All)</option>
        <%
        'continue until we get to end of recordset
        Do While Not rsOrg.EOF
        'for each option create an option tag with a corresponding value
        %>
        <option <% If instr(Request.Form("cboService"), rsOrg.Fields("OrgID"))<0 Then
Response.Write("selected")%>
value='<%=rsOrg.Fields("OrgID")%>'><%=rsOrg.Fields("OrgID")%> </option>
        <%
        'get next record
        rsOrg.Movenext
        Loop
        rsOrg.close
        %>
        </select></font>
    </td>
    <td nowrap width="109" valign="top"><font size="1">
        <select size="4" name="cboYear" class="select" multiple style="width:100">
        <option value=""<% If (Request.Form("cboYear")="") or
(Request.Form("cboYear")=" ", " ) Then Response.Write("selected")%>(All)</option>
        <%
        'continue until we get to end of recordset
        Do While Not rsYear.EOF
        'for each option create an option tag with a corresponding value
        %>
        <option <% If instr(Request.Form("cboYear"),
Cstr(rsYear.Fields("MishapYear")))<0 Then Response.Write("selected")%>
value=<%=rsYear.Fields("MishapYear")%>><%=rsYear.Fields("MishapYear")%></option>
        <%
        'get next record
        rsYear.Movenext
        Loop
        rsYear.close
        %>
        </select></font>
    </td>
</tr>
<!-------Factor Criteria----->
<tr>
    <td nowrap colspan="2" align="left" width="220"></td>
    <td nowrap colspan="2" align="left" width="220"></td>
    <td nowrap colspan="2" align="left" width="220"></td>
</tr>
<tr>

```

```

        <td nowrap colspan="2" align="left" width="220" height="21"><b><font
size="1">1st Level Factor</font></b></td>
        <td nowrap colspan="2" align="left" width="220" height="21"><b><font
size="1">2nd Level Factor</font></b></td>
        <td nowrap colspan="2" align="left" width="220" height="21"><b><font
size="1">3rd Level Factor</font></b></td>
    </tr>
    <tr>
        <td nowrap colspan="2" width="220" valign="top"><font size="1">
            <select size="4" name="cbo1stLevelFactors" class=select multiple
style="width:220">
                <option <% If (Request.Form("cbo1stLevelFactors")="") or
(Request.Form("cbo1stLevelFactors")=" ", " ) Then Response.Write("selected")%>
value="">(All)</option>
                <%
                'continue until we get to end of recordset
                Do While Not rs1st.EOF
                %>
                <option <% If instr(Request.Form("cbo1stLevelFactors"),
rs1st.Fields("1stLevelCode"))<>0 Then Response.Write("selected")%>
value='<%=rs1st.Fields("1stLevelCode")%>'> <%=rs1st.Fields("1stLevelDesc")%></option>
                <%
                'get next record
                rs1st.Movenext
                Loop
                rs1st.close
                %>
            </select></font>
        </td>
        <td nowrap colspan="2" width="220" valign="top"><font size="1">
            <select size="4" name="cbo2ndLevelFactors" class=select multiple
style="width:220">
                <option <% If (Request.Form("cbo2ndLevelFactors")="") or
(Request.Form("cbo2ndLevelFactors")=" ", " ) Then Response.Write("selected")%>
value="">(All)</option>
                <%
                'continue until we get to end of recordset
                Do While Not rs2nd.EOF
                %>
                <option <% If instr(Request.Form("cbo2ndLevelFactors"),
rs2nd.Fields("2ndLevelCode"))<>0 Then Response.Write("selected")%>
value='<%=rs2nd.Fields("2ndLevelCode")%>'> <%=rs2nd.Fields("2ndLevelDesc")%></option>
                <%
                'get next record
                rs2nd.Movenext
                Loop
                rs2nd.close
                %>
            </select></font>
        </td>
        <td nowrap colspan="2" width="220" valign="top"><font size="1">
            <select size="4" name="cbo3rdLevelFactors" class=select multiple
style="width:220">
                <option <% If (Request.Form("cbo3rdLevelFactors")="") or
(Request.Form("cbo3rdLevelFactors")=" ", " ) Then Response.Write("selected")%>
value="">(All)</option>
                <%
                'continue until we get to end of recordset
                Do While Not rs3rd.EOF
                %>

```



```

        <option <% If instr(Request.Form("cbo3rdLevelFactors"),
rs3rd.Fields("3rdLevelCode"))<>0 Then Response.Write("selected")%>
value='<%=rs3rd.Fields("3rdLevelCode")%>'><%=rs3rd.Fields("3rdLevelDesc")%></option>
        <%
        'get next record
        rs3rd.Movenext
        Loop
        rs3rd.close
        %>
    </select></font>
</td>
</tr>

<%
'Release connections
set rsAC=nothing
set rsOrg=nothing
set rsYear=nothing
set rs1st=nothing
set rs2nd=nothing
set rs3rd=nothing

%>
</table>
</center>
</div>

```

disclaimer.asp

This page generates a DoD required disclaimer. This disclaimer is patterned after the current NPS disclaimer given that HFACS–ME Web is currently hosted on a Web server within the NPS network.

```
<body Link="#000080" vlink="#000080" aLink="#0000FF">
<div align="center">
<center>
<!--#include FILE = "../common/header.htm" -->
<table border="0" width="723" height="1275">
  <tr>
    <td valign="top" height="32">
      <p align="center"><font size="2">
        <a href="#privacy">Privacy Act Notice</a> |&nbsp;
        <a href="#advisory">Privacy Advisory</a> |&nbsp;
        <a href="#security">Security Notice<br></a>
        <a href="#disclaimer">Disclaimer for External Links</a> |&nbsp;
        <a href="#cookie">Cookie Disclaimer</a></font></p>
      <p align="center">&nbsp;</p>
    </td>
  </tr>
  <tr>
    <td valign="top" height="1210" width="715">
      <h3 align="center">&nbsp;<font face="Arial"><a name="privacy"></a>Privacy Act
      Notice</font></h3>
      <p><font size="2">The Human Factors Analysis Classification System -
      Maintenance Extensions (HFACS–ME) homepage is provided as a service by the Aviation
      Safety School</font></p>
      <p><font size="2">Information presented on the Naval Safety Center homepage is
      considered public information and may be distributed or copied for non-commercial
      purposes. Use of appropriate byline, photo and image credits is requested.</font></p>
      <p><font size="2">To help manage this site, we collect information. We use
      software programs to create summary statistics, which are used for website planning
      and maintenance, determining technical design specifications, and analyzing system
      performance. For security purposes, and to ensure that this service remains available
      to all users, we use software programs to monitor network traffic and to identify
      unauthorized users.</font></p>
      <p><font size="2">Except for authorized law enforcement investigations, no
      other attempts are made to identify individual users or their usage habits. Server
      logs are scheduled for regular destruction in accordance with National Archives and
      Records Administration General Schedule 20.</font></p>
      <p><font size="2">Unauthorized attempts to upload information or change
      information on this service are strictly prohibited and may be punishable under the
      Computer Fraud and Abuse Act of 1986 and the National Information Infrastructure
      Protection Act.</font></p>
      <h3 ALIGN="CENTER"><font face="Arial"><a name="advisory"></a>Privacy
      Advisory</font></h3>
      <p><font size="2">If you identify yourself by sending an e-mail, you may also
      decide to identify yourself by sending personal information such as name, address,
      phone number and e-mail address. Information is collected for the purpose of
      responding to requests or comments, and to improve our services. Your e-mail may be
      forwarded to other government agencies who are better able to respond to your request.
      The information you send may be retained for documentation purposes or follow up
      contacts. In other limited circumstances, including requests from Congress or limited
      other parties, we may be required by law to disclose information that you
      submit.</font></p>
    </td>
  </tr>
</table>
</div>
</body>
```

Be aware that despite anything we do to protect electronic information, e-mail is not necessarily secure against interception. If your communication is sensitive, or includes personal information, you may prefer to send your comments by regular mail instead.

Security Notice

Warning: USE OF THIS OR ANY OTHER DEPT. OF DEFENSE INTEREST COMPUTER SYSTEM (DODICS) CONSTITUTES AND EXPRESS CONSENT TO MONITORING AT ALL TIMES.

This DODICS and all related equipment are to be used for the communication, transmission, processing, and storage of official U.S. Government or other authorized information only. All DODICS are subject to monitoring at all times. If monitoring of any DODICS reveals possible violation of criminal statutes, all relevant information may be provided to law enforcement officials. This World-Wide Web (www) Server is provided as a service to the Department of Defense for distribution of publicly available information. Naval Safety Center is a command in the United States Navy. Use of this www Server is intended only for the retrieval of information available through Naval Safety Center www documents. Only authorized administrators may make changes to the information or software content of this www Server. After reading and understanding the foregoing statement, you may continue with this www Server or exit from this server (or document).
USE OF THIS SYSTEM CONSTITUTES A CONSENT TO MONITORING AT ALL TIMES.

Disclaimer for External Links

The appearance of hyperlinks does not constitute endorsement by the Department of Defense of the hyperlinked Web site or information, products or services contained therein. For other than authorized activities such as military exchanges and Morale, Welfare and recreation sites, the Department of Defense does not exercise any editorial control over the information you may find at these locations. Such links are provided consistent with the stated purpose of this DoD Web site.

Cookie Disclaimer

The Naval Safety Center does not use persistent cookies, i.e., tokens that pass information back and forth from your machine to the server and remain after you close your browser. The Naval Safety Center does use session cookies, i.e., tokens that remain active only until you close your browser, in order to provide accurate web statistics ie. unique users, visits and page hits. This information cannot be obtained without a session cookie that 'follows the user' through the Website. No database of information obtained from these cookies is kept and when you close your browser, the cookie is deleted from your computer.

The Naval Safety Center uses session cookies for the sole purpose of quantifying user statistics in an effort to improve the readability and informational quality of our Website.

You may choose not to accept these cookies and still use the site. The help information in your browser software should provide you with instructions on how to disable cookies.

<!--#include FILE = "footer.htm" -->

footer.htm

Footer.htm is a server-side Include file which is added to all HFACS–ME Web pages (except reports.asp) to provide a link to the top of the page, the Disclaimer, and the last revision date of the page.

```
<div align="center">
<center>
<table border="0" cellpadding="0" cellspacing="0" bgcolor="#ffffff">
  <tr>
    <td valign="top" align="center">
      <br><br>
      <span class="lighttext">
        :: <a href="#top">back to top</a> ::
      <br><br><br>
    <b><a href=http://www.nps.navy.mil/disclaimer/ target="_blank"> Government
    Disclaimer</a></b>
      <br><br>
      <!--#include FILE = "../common/mod_date.inc" -->
      </span><br><br></td>
  </tr>
</table>

</center>
</div>
```

footer2.htm

Footer2.htm is identical to footer.htm except that it deletes the ::back to top:: link. It is used only on the report.asp page where links to the top of the page are part of the existing page code. This prevents a duplicate ::back to top:: link at the bottom of the page.

```
<div align="center">
  <center> <table border="0" cellpadding="0" cellspacing="0" bgcolor="#ffffff">
    <tr>
      <td valign="top" align="center">
        <br><br>
        <span class="lighttext">
          <b><a href=http://www.nps.navy.mil/disclaimer/ target="_blank"> Government
          Disclaimer</a></b>
            <br><br>
            <!--#include FILE = "../common/mod_date.inc" -->
            </span><br><br></td>
        </tr>
      </table>

    </center>
  </div>
```

header.htm

Header.htm is a server-side Include file which is the basis for formatting all HFACS–ME Web pages. It incorporates all graphics and menu bars and calls the site style sheet (style.css) and two javascript files (img.js and button_functions.js) which control Dynamic HTML (i.e., mouse_over events, etc.)

```
<html>
<head>
  <title>H F A C S - M E</title>

  <link rel="stylesheet" type="text/css" href="../scripts/style.css">
  <script language="javascript" src="../scripts/img.js"></script>
  <script language="javascript" src="../scripts/button_functions.js"></script>

</head>
<body background="../images/background.gif" bgcolor="#ffffff" marginheight="0"
marginwidth="0" topmargin="0" leftmargin="0">
<a name="top"></a>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr align="center">
    <td align="center">
      <table border="0" cellpadding="0" cellspacing="0">
        <tr valign="top">
          <td valign="top"></td>
          <td valign="top"></td>
        </tr>
      </table>
    </td>
  </tr>
</table>

  <table border="0" cellpadding="0" cellspacing="0">
    <tr valign="top">
<!--HOME hyperLink-->
      <td valign="top" width="87"><a href="../home/index.asp"
onmouseover="imgOver('link0'); window.status=''; return true;"
onmouseout="imgOut('link0');">
        </a></td>

<!--MISHAP DATA hyperLink-->
      <td valign="top" width="140">
        <a href="../mishap/query.asp" onmouseover="imgOver('link2');
window.status=''; return true;" onmouseout="imgOut('link2');">
          </a></td>

<!--FACTOR ANALYSIS hyperLink-->
      <td valign="top" width="165">
        <a href="../factors/factors.asp" onmouseover="imgOver('link3');
window.status=''; return true;" onmouseout="imgOut('link3');">
          </a></td>

<!--GRAPH DATA hyperLink-->
```

```

        <td valign="top" width="131">
<a href=" ../graph/criteria_select.asp" onmouseover="imgOver('Link4');
window.status=''; return true;" onmouseout="imgOut('Link4');">
        </a></td>

<!--REPORTS hyperLink-->
        <td valign="top" width="108">
        <a href=" ../report/reportcall.asp" onmouseover="imgOver('Link1');
window.status=''; return true;" onmouseout="imgOut('Link1');">
        </a></td>
</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="700">
<tr valign="top">
        <td valign="top"></td>

<!--ONLINE TUTORIAL sublink-->
        <td valign="top" width="130" height="21">
        <a href=" ../tutorial/index.asp" onmouseover="imgOver('Link5');
window.status=''; return true;" onmouseout="imgOut('Link5');">
        </a></td>

<!--HELP sublink-->
        <td valign="top" width="56" height="21">
        <a href=" ../help/index.asp" onmouseover="imgOver('Link6');
window.status=''; return true;" onmouseout="imgOut('Link6');">
        </a></td>

<!--ACCESS POLICY sublink-->
        <td valign="top" width="108" height="21">
        <a href=" ../access/index.asp" onmouseover="imgOver('Link7');
window.status=''; return true;" onmouseout="imgOut('Link7');">
        </a></td>

<!--CONTACT US sublink-->
        <td valign="top" width="96" height="21">
        <a href=" ../contact/index.asp" onmouseover="imgOver('Link8');
window.status=''; return true;" onmouseout="imgOut('Link8');">
        </a></td>
</tr>
</table><br>
<div align="center">
        <td>
</td>
</tr>
</table>

```

hfacsme_taxonomy.htm

Hfacsme_taxonomy.htm is a page which pops-up in a new window to display a mapping of the HFACS–ME taxonomy and the 1st, 2nd, and 3rd Level codes used in the Graph display (graph.asp). The hyperlink to display the hfacsme_taxonomy.htm page is only displayed if the user selects either 1st, 2nd, or 3rd Level Factors as a grouping option.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>HFACS–ME Taxonomy</title>
</head>

<body>

<p align="left"><b><font size="5">HFACS–ME Taxonomy</font></b></p>
<div align="left">

<table border="1" cellpadding="0" cellspacing="0" style="border-collapse:
collapse;mso-table-layout-alt:fixed;border:none;mso-border-alt:solid windowtext
.75pt;
mso-table-lspace:9.0pt;mso-table-rspace:9.0pt;mso-table-anchor-vertical:paragraph;
mso-table-anchor-horizontal:margin;mso-table-left:center;mso-table-top:20.5pt;
mso-padding-alt:0in 0in 0in 0in">
  <tr style="height:12.75pt">
    <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
      <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><b><span
style="font-size:10.0pt">1<sup>st</sup>
Order<o:p></o:p></span></b></p></td>
    <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
      <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><b><span
style="font-size:10.0pt">1st
Code<o:p></o:p></span></b></p></td>
    <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
      <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><b><span
style="font-size:10.0pt">2<sup>nd</sup>
```


Order	Order
2nd	3rd
Code	Code
Management	Management
Conditions	MG


```

mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">Organizational<o:p></span></p></td>
<td rowspan="4" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">ORG<o:p></span></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">Inadequate
Processes<o:p></span></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">PRO<o:p></span></p></td>
</tr>
<tr style="height:12.75pt">
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">Inadequate
Documentation<o:p>
</o:p></span></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">DOC<o:p>
</o:p></span></p></td>
</tr>
<tr style="height:12.75pt">
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-

```

```

color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
    <p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in;
    mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around;
    mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin;
    mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly"><span
style="font-size: 10.0pt">Inadequate
    Design<o:p>
    </o:p></span></p></td>
    <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
    <p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in;
    mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around;
    mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin;
    mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly"><span
style="font-size: 10.0pt">DES<o:p>
    </o:p></span></p></td>
</tr>
<tr style="height: 13.5pt">
    <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
    <p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in;
    mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around;
    mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin;
    mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly"><span
style="font-size: 10.0pt">Inadequate
    Resources<o:p>
    </o:p></span></p></td>
    <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
    <p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in;
    mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around;
    mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin;
    mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly"><span
style="font-size: 10.0pt">RES<o:p>
    </o:p></span></p></td>
</tr>
<tr style="height: 13.5pt">
    <td rowspan="4" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
    <p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in;
    mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around;
    mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin;
    mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly"><span
style="font-size: 10.0pt">Supervisory<o:p></span></p></td>
    <td rowspan="4" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
    <p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in;
    mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around;
    mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin;
    mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly">

```

mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">SUP<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Inadequate
Supervision<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">IDQ<o:p></p></td>

mso-element-anchor-internal:vertical:paragraph;mso-element-anchoring:horizontal:margin;mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"> PRB	
Supervisory Misconduct	
MIS	
Maintainer Conditions	
MC	
Medical	

mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">MED<o:p></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Adverse
Mental State<o:p></o:p></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">MNT<o:p></o:p></p></td>
</tr>
<tr style="height:12.75pt">
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Adverse
Physical State<o:p></o:p></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">PHY<o:p></o:p></p></td>
</tr>
<tr style="height:13.5pt">
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Unsafe
Limitation<o:p></o:p></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">LIM<o:p></o:p></p></td></tr>

<tr style="height:13.5pt">

<td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Crew
Coordination<o:p></o:p></p></td>

<td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">CRW<o:p></o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Inadequate Communication<o:p></o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">COM<o:p></o:p></p></td></tr>

<tr style="height:12.75pt">

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Inadequate Assertiveness<o:p></o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-

```

color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
    <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
    mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
    mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
    mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">ASS<o:p></o:p></span></p></td>
</tr>
<tr style="height:13.5pt">
    <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
        <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
        mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
        mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
        mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Inadequate
        Adaptability/Flexibility<o:p></o:p></span></p></td>
        <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
            <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
            mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
            mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
            mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">ADA<o:p></o:p></span></p></td>
</tr>
<tr style="height:13.5pt">
    <td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
        <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
        mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
        mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
        mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Readiness<o:p></o:p></span></p></td>
        <td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
            <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
            mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
            mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
            mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">RDY<o:p></o:p></span></p></td>
            <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
                <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
                mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
                mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
                mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Inadequate
                Training/Preparation<o:p></o:p></span></p></td>
            <td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-

```

color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">TRG<o:p></o:p></p></td>

</tr>

<tr style="height:12.75pt">

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Certification/Qualification<o:p></o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">CRT<o:p></o:p></p></td>

</tr>

<tr style="height:13.5pt">

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Personnel

Readiness Infringement<o:p></o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">INF<o:p></o:p></p></td>

</tr>

<tr style="height:13.5pt">

<td rowspan="9" valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Working
Conditions<o:p></o:p></p></td>

Environment	ENV
Inadequate Lighting/Light	
Unsafe Weather/Exposure	

color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">W&E<o:p></p></td>

</tr>

<tr style="height:13.5pt">

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Unsafe

Environmental Hazards<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">EHZ<o:p></p></td>

</tr>

<tr style="height:13.5pt">

<td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Equipment<o:p></p></td>

<td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">EQP<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">Damaged/Unserviced<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-

color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span style="font-size:10.0pt">DMG<o:p></p></td>

</tr>

<tr style="height:12.75pt">

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span style="font-size:10.0pt">Unavailable/Inappropriate<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span style="font-size:10.0pt">UNA<o:p></o:p></p></td>

</tr>

<tr style="height:13.5pt">

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span style="font-size:10.0pt">Dated/Uncertified<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span style="font-size:10.0pt">DUC<o:p></p></td>

</tr>

<tr style="height:13.5pt">

<td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in; mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around; mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin; mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span style="font-size:10.0pt">Workspace<o:p></p></td>

<td rowspan="3" valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid;

border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in; mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around; mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin; mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly">WRK<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in; mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around; mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin; mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly">Confining<o:p></p></td>

<td valign="top" style="text-align: Left; line-height: 100%; border-right-style: solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt; padding-bottom: 0in">

<p class="MsoNormal" align="left" style="text-align: left; text-indent: 0in; mso-element: frame; mso-element-frame-hspace: 9.0pt; mso-element-wrap: around; mso-element-anchor-vertical: paragraph; mso-element-anchor-horizontal: margin; mso-element-left: center; mso-element-top: 20.5pt; mso-height-rule: exactly">CON<o:p></p></td>

```

        <p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">INA<o:p></span></p></td>
</tr>
<tr style="height:13.5pt">
<td rowspan="8" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Maintainer<br>
Acts<o:p></span></p></td>
<td rowspan="8" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">MA<o:p></span></p></td>
<td rowspan="4" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Error<o:p></span></p></td>
<td rowspan="4" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">ERR<o:p></span></p></td>
<td rowspan="4" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Attention/Memory<o:p></span></p></td>
<td rowspan="4" valign="top" style="text-align: Left; line-height: 100%; border-
right-style: solid; border-right-color: windowtext; border-bottom-style: solid;
border-bottom-color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-
top: 1.4pt; padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly"><span
style="font-size:10.0pt">Attention/Memory<o:p></span></p></td>

```

ATT
Judgment/Decision
JDG
Knowledge/Rule Based
KNW
Skill/Technique


```

mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">IFC<o:p></span></p></td>
</tr>
<tr style="height:12.75pt">
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">Exceptional<o:p></span></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">EXC<o:p></span></p></td>
</tr>
<tr style="height:13.5pt">
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">Flagrant<o:p></span></p></td>
<td valign="top" style="text-align: Left; line-height: 100%; border-right-style:
solid; border-right-color: windowtext; border-bottom-style: solid; border-bottom-
color: windowtext; padding-left: 1.4pt; padding-right: 1.4pt; padding-top: 1.4pt;
padding-bottom: 0in">
<p class="MsoNormal" align="left" style="text-align:left;text-indent:0in;
mso-element:frame;mso-element-frame-hspace:9.0pt;mso-element-wrap:around;
mso-element-anchor-vertical:paragraph;mso-element-anchor-horizontal:margin;
mso-element-left:center;mso-element-top:20.5pt;mso-height-rule:exactly">span
style="font-size:10.0pt">FLG<o:p></o:p></span></p></td>
</tr>
</table>
</div>
</body>
</html>

```


mod_date.inc

Mod_date.inc is a server-side Include which is called from the footer.htm page. It inserts the Last Modified date at the bottom of every page.

```
<SCRIPT LANGUAGE=VBScript RUNAT=SERVER>
Function FileLastMod()
    ' Local variables
    Dim loFs, lsFile, lsPath, loFile, ldLast

    'Create an instance of FileSystemObject object
    Set loFs = CreateObject("Scripting.FileSystemObject")

    'Get the path of the current file (i.e. the file in which this code runs)
    lsFile = Request.ServerVariables("SCRIPT_NAME")

    'Get the physical path of the file
    lsPath = Server.MapPath(lsFile)

    'Get a handle/pointer to this file
    Set loFile = loFs.GetFile(lsPath)

    'Get the "Last Modified" property of this file
    ldLast = loFile.DateLastModified

    'Release the objects
    Set loFile = Nothing
    Set loFs = Nothing

    'Write out the date in the long date format e.g. "MM/DD/YY"
    FileLastMod = CStr(FormatDateTime(ldLast, 2))
End Function
</SCRIPT>

<% Response.Write ("<br>Last Modified " & FileLastMod()) %>
```

FACTORS SUBDIRECTORY

factors.asp

Factors.asp generates the Factors Analysis display. If no user criteria is passed to the page via Request.Form variables (including hidden variables) then the page produces the Factor Analysis display with all mishaps contained in the database. If criteria are provided by the user, the page dynamically builds the SQL string needed to "pre-filter" the mishap data before running the stored procedure to generate the table data.

```
<@ Language=VBScript %>
<%Option Explicit%>
<%Response.Buffer=true%>
<%
Dim cmd                'command object
Dim rsCodes            'recordset for Mishap Breakdown table counts
Dim rsTemp
Dim strParam          'string to hold stored procedure parameters
Dim blnMultiple
Dim blnFactors
Dim string
Dim tempstring
Dim m_year
Dim strSelect          '
Dim intTotal           '
Dim strCmd             '
Dim strTitle

'Declare all Mishap Count variables
Dim intMG
Dim intORG, intSUP, intPRO, intDOC, intDES
Dim intRES, intIDQ, intOPS, intPRB, intMIS
Dim intMC
Dim intMED, intMNT, intPHY, intLIM
Dim intCRW, intCOM, intASS, intADA
Dim intRDY, intTRG, intCRT, intINF
Dim intWC
Dim intENV, intLGT, intWXE, intEHZ
Dim intEQP, intDMG, intUNA, intDUC
Dim intWRK, intCON, intOBS, intINA
Dim intMA
Dim intERR, intATT, intJDG, intKNW, intSKL
Dim intVIO, intROU, intIFC, intFLG, intEXC
%>
<!-- #include FILE = "../common/adovbs.inc" -->
<!-- #include FILE = "../common/connection.asp" -->
<%
Function Prepstring(string)
    Do While Left(string,1)=", "
        string=Trim(Mid(string,2))
    Loop
    If string <> "" Then
        string = Replace( string, " ", "' ' " ) ' add apostrophes to CSV string
    End if
    Prepstring = string
End Function
```

```

Sub BuildTable(rsCodes, strTitle)

'Avoid divide by zero error
If rsCodes.Fields("TotalMishaps")=0 Then
    intTotal = 1
Else
    intTotal = rsCodes.Fields("TotalMishaps")
End If

'store MG factor counts in local variables
intMG = rsCodes.Fields("MG")

intORG = rsCodes.Fields("ORG")
intSUP = rsCodes.Fields("SUP")
intPRO = rsCodes.Fields("PRO")
intDOC = rsCodes.Fields("DOC")
intDES = rsCodes.Fields("DES")

intRES = rsCodes.Fields("RES")
intIDQ = rsCodes.Fields("IDQ")
intOPS = rsCodes.Fields("OPS")
intPRB = rsCodes.Fields("PRB")
intMIS = rsCodes.Fields("MIS")

'store MC factor counts in local variables
intMC = rsCodes.Fields("MC")

intMED = rsCodes.Fields("MED")
intMNT = rsCodes.Fields("MNT")
intPHY = rsCodes.Fields("PHY")
intLIM = rsCodes.Fields("LIM")

intCRW = rsCodes.Fields("CRW")
intCOM = rsCodes.Fields("COM")
intASS = rsCodes.Fields("ASS")
intADA = rsCodes.Fields("ADA")

intRDY = rsCodes.Fields("RDY")
intTRG = rsCodes.Fields("TRG")
intCRT = rsCodes.Fields("CRT")
intINF = rsCodes.Fields("INF")

'store WC factor counts in local variables
intWC = rsCodes.Fields("WC")

intENV = rsCodes.Fields("ENV")
intLGT = rsCodes.Fields("LGT")
intWXE = rsCodes.Fields("WXE")
intEHZ = rsCodes.Fields("EHZ")

intEQP = rsCodes.Fields("EQP")
intDMG = rsCodes.Fields("DMG")
intUNA = rsCodes.Fields("UNA")
intDUC = rsCodes.Fields("DUC")

intWRK = rsCodes.Fields("WRK")
intCON = rsCodes.Fields("CON")
intOBS = rsCodes.Fields("OBS")
intINA = rsCodes.Fields("INA")

```


[illegible]

	<%=IntCRT%>&nbsp;&~<%=FormatPercent((IntCRT/intTotal),0)%</td>
	<td width="118" valign="middle" align="center" bgcolor="#DBDCAD" height="40">Personal Infringement
</div>
	<%=IntINF%>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<%=FormatPercent((IntINF/intTotal),0)%</td>
	<td width="110" valign="middle" align="center" bgcolor="#6695B0" rowspan="3" height="8">Working
Conditions
</div>
	<%=IntWC%>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<%=FormatPercent((IntWC/intTotal),0)%</td>
	<td width="107" valign="middle" align="center" bgcolor="#79AECC" height="40">Environment
</div>
	<%=IntENV%>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<%=FormatPercent((IntENV/intTotal),0)%</td>
	<td width="117" valign="middle" align="center" bgcolor="#DBDCAD" height="40">Lighting/Light
</div>
	<%=IntLGT%>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<%=FormatPercent((IntLGT/intTotal),0)%</td>
	<td width="118" valign="middle" align="center" bgcolor="#DBDCAD" height="40">Weather/Exposure
</div>
	<%=IntWXE%>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~<%=FormatPercent((IntWXE/intTotal),0)%</td>
	<td width="118" valign="middle" align="center" bgcolor="#DBDCAD" height="40">Environmental Hazards
</div>
	<%=IntEHZ%>&nbsp;&nbsp;&nbsp;&~<%=FormatPercent((IntEHZ/intTotal),0)%</td>
	<td width="107" valign="middle" align="center" bgcolor="#79AECC" height="40">Equipment
</div>
	<%=IntEQP%>&nbsp;&nbsp;&~<%=FormatPercent((IntEQP/intTotal),0)%</td>
	<td width="117" valign="middle" align="center" bgcolor="#DBDCAD" height="40">Damaged/ Unserved
</div>
	<%=IntDMG%>&nbsp;&nbsp;&~<%=FormatPercent((IntDMG/intTotal),0)%</td>
	<td width="118" valign="middle" align="center" bgcolor="#DBDCAD" height="40">Unavailable/ Inappropriate
</div>
	<%=IntUNA%>&nbsp;&~<%=FormatPercent((IntUNA/intTotal),0)%</td>
	<td width="118" valign="middle" align="center" bgcolor="#DBDCAD" height="40">Dated/
Uncertified
</div>
	<%=IntDUC%>&nbsp;&~<%=FormatPercent((IntDUC/intTotal),0)%</td>

[illegible]


```

End If

tempstring = Prepstring(Request.Form("cboClass"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
        strTitle = strTitle & " and "
    End if
    strParam = strParam & "tblMishaps.Class_FK IN ('" & tempstring & "')"
    strTitle = strTitle & "Class=" & tempstring
    blnMultiple=True
End If

tempstring = Prepstring(Request.Form("cboLocation"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
        strTitle = strTitle & " and "
    End if
    strParam = strParam & "tblMishaps.LocationID_FK IN ('" & tempstring & "')"
    strTitle = strTitle & "Location=" & tempstring
    blnMultiple=True
End If

tempstring = Prepstring(Request.Form("cboService"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
        strTitle = strTitle & " and "
    End if
    strParam = strParam & "tblMishaps.OrgID_FK IN ('" & tempstring & "')"
    strTitle = strTitle & "Organization=" & tempstring
    blnMultiple=True
End If

'-----Call multi-select function-----
'multiple select of an integer
m_year = Request.Form("cboYear")
If m_year <> "" Then
    Do While Left(m_year,1)=", "
        m_year=Trim(Mid(m_year,2))
    Loop

    If blnMultiple Then
        strParam = strParam & " AND "
        strTitle = strTitle & " and "
    End if

    strParam = strParam & "Year(DateAdd(month,3,tblMishaps.MishapDate)) IN (" & m_year
& ") "
    strTitle = strTitle & "FY=" & m_year
    blnMultiple=True
End If

tempstring = Prepstring(Request.Form("cbo1stLevelFactors"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
        strTitle = strTitle & " and "
    End if

```

```

        strParam = strParam & "tblFactors.[1stLevelCode] IN ('" & tempstring & "')"
        strTitle = strTitle & "1st Lvl Factor=" & tempstring
        blnMultiple=True
        blnFactors=True
    End If

    tempstring = Prepstring(Request.Form("cbo2ndLevelFactors"))
    If tempstring <> "" Then
        If blnMultiple Then
            strParam = strParam & " AND "
            strTitle = strTitle & " and "
        End if
        strParam = strParam & "tblFactors.[2ndLevelCode] IN ('" & tempstring & "')"
        strTitle = strTitle & "2nd Lvl Factor=" & tempstring
        blnMultiple=True
        blnFactors=True
    End If

    tempstring = Prepstring(Request.Form("cbo3rdLevelFactors"))
    If tempstring <> "" Then
        If blnMultiple Then
            strParam = strParam & " AND "
            strTitle = strTitle & " and "
        End if
        strParam = strParam & "tblFactors.[3rdLevelCode] IN ('" & tempstring & "')"
        strTitle = strTitle & "3rd Lvl Factor=" & tempstring
        blnMultiple=True
        blnFactors=True
    End If

    If strTitle = "(" then
        strTitle = "(All Mishaps)"
    Else
        strTitle = strTitle & ")"
    End if
    strTitle = REPLACE(strTitle, "'", "")

    strSelect = "SELECT DISTINCT tblMishaps.MishapID INTO ##tblTemp_Filter_Table"
    strSelect = strSelect & " FROM tblMishaps"
    If blnFactors Then

        strSelect = strSelect & " INNER JOIN tblMishapFactors ON tblMishaps.MishapID"
    = tblMishapFactors.MishapID_FK"
        strSelect = strSelect & " INNER JOIN tblFactors ON"
tblMishapFactors.[3rdLevelCode_FK] = tblFactors.[3rdLevelCode]"
    End if
    strSelect = strSelect & " WHERE NOT(tblMishaps.DatabaseType='C')"
    If blnMultiple Then
        strSelect = strSelect & " AND " & Trim(strParam)
    End if

    strSelect = REPLACE(strSelect, "'", "''")

    strCmd="spMishapCount_Filtered_with_Factors_@query @query='" & strSelect & "'"

    cmd.CommandText = strCmd
    Set rsCodes=cmd.Execute

    If Err.Number <> 0 Then
        Response.Write "An error has occurred!<br>"
    End If

```


criteria select.asp

```
<%@ Language=VBScript %>
<%Option Explicit%>
<%Response.Buffer=false%>
<!-- #include FILE = "../common/adovbs.inc" -->
<!-- #include FILE = "../common/connection.asp" -->
<%
Dim cmd          'command object

Set cmd = Server.CreateObject("ADODB.Command")
cmd.ActiveConnection = conn
%>
<!-- #include FILE = "../common/header.htm" -->
<div align="center">
    <center>
<table border="0" cellspacing="0" cellpadding="0" width="700">
    <tr>
        <td><font size="4">Graph Data Selection Page</font></td>
    </tr>
    <tr>
        <td><font size="2">Use the drop down boxes below to select the desired Mishap
            Criteria and Factors Criteria to include in the dataset.&nbsp; Multiple items
            <b>within</b> a particular criteria (ie. Aircraft Type = F14, F18) may be
            selected by holding down the Ctrl button and left clicking on the desired
            items.&nbsp; This will result in a query that will return data matching ANY of
            the selected items.&nbsp;&nbsp;&nbsp;</font>
            <p><font size="2">If criteria are selected in several criteria boxes (ie. Aircraft
            type and Mishap Class), the resulting dataset will be only those records that
            match ALL criteria.&nbsp; For example, if the user selects <b>Aircraft
            Type=F18, F14</b> and <b>Mishap Class=A, </b>then the database will return all
            records involving either a F14 or F18 and resulted in a Class A
            mishap.&nbsp;<br></font>
            </td>
        </tr>
    </table>
</center>
</div>

<form method="post" action="grouping_select.asp" style="text-align: left">
<!-- #include FILE = "../common/criteria_dropdown_INC.asp" -->
<div align="center">
<table border="0" cellspacing="0" cellpadding="0" width="700">
    <tr><td>&nbsp;</td></tr>
    <tr>
        <td valign="bottom" colspan="6" width="664"><p align="center">
            <input type="submit" value="Submit Query">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
            <input type="reset" value="Reset" ~ "></p>
        </td>
    </tr>
</table>
```

```
</table>
<br><br>
</div>
</form>
```

```
<%
'Close connections
conn.close
set conn=nothing
%>
```

```
<!-- #include FILE = "../common/footer.htm" -->
```

graphpage.asp

Graphpage.asp generates a pseudo 3D graph of the data selected by the user on the criteria_select.asp and grouping_select.asp pages. Data from dynamic SQL query is stored in arrays and passed to Javascript graph function which outputs HTML graph and legend.

```
<%@ Language=VBScript %>
<%Option Explicit%>
<%Response.Buffer=false%>

<!-- #include FILE = "../common/adovbs.inc" -->
<!-- #include FILE = "../common/connection.asp" -->

<%
Dim cmd                'command object
Dim rsGraph            'recordset for Mishap Breakdown table counts
Dim strSelect          'string to hold SELECT statement
Dim strFrom            'string to hold FROM statement
Dim strWhere           'string to hold WHERE statement
Dim strParam           'string to hold WHERE parameters
Dim strGroupBy         'string to hold GROUP BY statement
Dim strOrderBy         'string to hold ORDER BY statement
Dim strColumn1         'string to hold user selected column name
Dim strColumn2         'string to hold user selected column name
Dim blnMultiple
Dim strSQL
Dim strPOST
dim tempString
dim m_year
Dim strCol1List, strCol2List, strColumnLabel_1, strColumnLabel_2
Dim rsCol1, rsCol2
Dim arLongerList,arrayRow,zValue
dim recCnt1,recCnt2
dim xaxis,zaxis,xLabel,zLabel
dim iscreenx,iscreeny,igraphx,igraphy,sShowtable
Dim blnTaxonomy,blnTaxonomy1,blnTaxonomy2,blnTaxonomyTemp

iscreenx = request("screenx")
if iscreenx = "" then
    iscreenx = session("screenx")
else
    session("screenx") = iscreenx
end if
iscreeny = request("screeny")
sShowtable = request("showtable")
igraphx = (iscreenx/2)-324 ' don't need screeny yet
if igraphx < 10 then igraphx = 10

Function ListColumn(rsCol)
<br>
<table border="1" cellpadding="0" cellspacing="0">
    <tr>
        <%
            rsCol.MoveFirst
            Do While NOT rsCol.EOF%>
                <td align="center" bgcolor="#FFFFFF"><%=rsCol.Fields("List")%></td>
        <%
            'get next record
```

```

                rsCol.MoveNext
            Loop
        </tr>
    </table>
    <br>
<% End Function

Function Prepstring(string)
Do While Left(string,1)=","
    string=Trim(Mid(string,2))
Loop
If string <> "" Then
    string = Replace( string, ", ", "','" ) ' add apostrophes to CSV string
End if
Prepstring = string
End Function

Function ColumnSelect(strColumn, blnTaxonomyTemp)
    blnTaxonomyTemp = "false"

    Select Case strColumn
        Case "Aircraft"
            ColumnSelect = "tblMishaps.Aircraft_FK"
        Case "Type"
            ColumnSelect = "tblMishaps.Type_FK"
        Case "Class"
            ColumnSelect = "tblMishaps.Class_FK"
        Case "Location"
            ColumnSelect = "tblMishaps.LocationID_FK"
        Case "Service"
            ColumnSelect = "tblMishaps.OrgID_FK"
        Case "Fiscal Year"
            ColumnSelect = "Year(DateAdd(month,3,tblMishaps.MishapDate))"
        Case "1st Level Factor"
            ColumnSelect = "tblFactors.[1stLevelCode]"
            blnTaxonomyTemp = "true"
        Case "2nd Level Factor"
            ColumnSelect = "tblFactors.[2ndLevelCode]"
            blnTaxonomyTemp = "true"
        Case "3rd Level Factor"
            ColumnSelect = "tblFactors.[3rdLevelCode]"
            blnTaxonomyTemp = "true"
        Case Else
            Response.write "Column Select Error. strColumn = " & strColumn
    End Select

End Function

Set cmd = Server.CreateObject("ADODB.Command")
cmd.ActiveConnection = conn
strParam = ""
blnMultiple = False
' If Request.ServerVariables("REQUEST_METHOD") = "POST" Then
    strPOST="True"
    'Build parameter list with Mishap Factors drop-down selections (NULL
    'values not stored)
    'multiple select of a string

tempstring = Prepstring(Request.Form("cboAircraft"))
If tempstring <> "" Then

```



```

        strParam = "tblMishaps.Aircraft_FK IN ('" & tempstring & "')"
        blnMultiple=True
    End if

    tempstring = Prepstring(Request.Form("cboType"))
    If tempstring <> "" Then
        If blnMultiple Then
            strParam = strParam & " AND "
        End if
        strParam = strParam & "tblMishaps.Type_FK IN ('" & tempstring & "')"
        blnMultiple=True
    End If

    tempstring = Prepstring(Request.Form("cboClass"))
    If tempstring <> "" Then
        If blnMultiple Then
            strParam = strParam & " AND "
        End if
        strParam = strParam & "tblMishaps.Class_FK IN ('" & tempstring & "')"
        blnMultiple=True
    End If

    tempstring = Prepstring(Request.Form("cboLocation"))
    If tempstring <> "" Then
        If blnMultiple Then
            strParam = strParam & " AND "
        End if
        strParam = strParam & "tblMishaps.LocationID_FK IN ('" & tempstring & "')"
        blnMultiple=True
    End If

    tempstring = Prepstring(Request.Form("cboService"))
    If tempstring <> "" Then
        If blnMultiple Then
            strParam = strParam & " AND "
        End if
        strParam = strParam & "tblMishaps.OrgID_FK IN ('" & tempstring & "')"
        blnMultiple=True
    End If

    '-----Call multi-select function-----
    'multiple select of an integer
    m_year = Request.Form("cboYear")
    If m_year <> "" Then
        Do While Left(m_year,1)=","
            m_year=Trim(Mid(m_year,2))
        Loop

        If blnMultiple Then
            strParam = strParam & " AND "
        End if

        strParam = strParam & "Year(DateAdd(month,3,tblMishaps.MishapDate)) IN (" & m_year
    & ")")
        blnMultiple=True
    End If

    tempstring = Prepstring(Request.Form("cbo1stLevelFactors"))
    If tempstring <> "" Then
        If blnMultiple Then

```

```

        strParam = strParam & " AND "
    End if
    strParam = strParam & "tblFactors.[1stLevelCode] IN ('" & tempstring & "')"
    blnMultiple=True
End If

tempstring = Prepstring(Request.Form("cbo2ndLevelFactors"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
    End if
    strParam=strParam & "tblFactors.[2ndLevelCode] IN ('" & tempstring & "')"
    blnMultiple=True
End If

tempstring=Prepstring(Request.Form("cbo3rdLevelFactors"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
    End if
    strParam=strParam & "tblFactors.[3rdLevelCode] IN ('" & tempstring & "')"
    blnMultiple=True
End If

    blnTaxonomy1="false"
    blnTaxonomy2="false"

    If Len(Request.form("cboColumn1"))>0 Then
        strColumnLabel_1 = Request.form("cboColumn1")
        strColumn1 = ColumnSelect(strColumnLabel_1,blnTaxonomy1)
    Else
        strColumnLabel_1 = "Aircraft"
        strColumn1 = "tblMishaps.Aircraft_FK"
    End if

    If Len(Request.form("cboColumn2"))>0 Then
        strColumnLabel_2 = Request.form("cboColumn2")
        strColumn2 = ColumnSelect(strColumnLabel_2, blnTaxonomy2)
    Else
        strColumnLabel_2 = "3rd Level Factors"
        strColumn2 = "tblMishapFactors.[3rdLevelCode_FK]"
        blnTaxonomy2 = "true"
    End if

    If (blnTaxonomy1="true" OR blnTaxonomy2="true") then
        blnTaxonomy = "true"
    Else
        blnTaxonomy = "false"
    End If

strSelect = "SELECT " & strColumn1 & " as Column1, " & strColumn2 & " as Column2,
COUNT(tblMishaps.MishapID) as Count "
strFrom = "FROM tblMishaps INNER JOIN tblMishapFactors ON tblMishaps.MishapID =
tblMishapFactors.MishapID_FK INNER JOIN tblFactors ON
tblMishapFactors.[3rdLevelCode_FK] = tblFactors.[3rdLevelCode] "
If Trim(strParam) <> "" Then
    strWhere = "WHERE NOT(tblMishaps.DatabaseType = 'C') AND " & Trim(strParam) & " "
Else
    strWhere = "WHERE NOT(tblMishaps.DatabaseType = 'C') "
End if

```

```

strGroupBy = "GROUP BY " & strColumn1 & ", " & strColumn2 & " "
strOrderBy = "ORDER BY " & strColumn1 & ", " & strColumn2
strSQL = strSelect & strFrom & strWhere & strGroupBy & strOrderBy
cmd.CommandText= strSQL
Set rsGraph = cmd.Execute
conn.CursorLocation = adUseClient

'Generate List of Items in Requested Columns 1&2
strCol1List = "SELECT DISTINCT " & strColumn1 & " as List " & strFrom & strWhere &
"ORDER BY " & strColumn1
cmd.CommandText= strCol1List
Set rsCol1 = cmd.Execute
strCol2List = "SELECT DISTINCT " & strColumn2 & " as List " & strFrom & strWhere &
"ORDER BY " & strColumn2
cmd.CommandText= strCol2List
Set rsCol2 = cmd.Execute

recCnt1 = rsCol1.recordcount
recCnt2 = rsCol2.recordcount
if recCnt1 > recCnt2 then
    arLongerList = rsCol2.GetRows
    xaxis = "Column2"
    xlabel = strColumnLabel_2
    zaxis = "Column1"
    zlabel = strColumnLabel_1
else
    arLongerList = rsCol1.GetRows
    xaxis = "Column1"
    xlabel = strColumnLabel_1
    zaxis = "Column2"
    zlabel = strColumnLabel_2
end if

'Response.write "<br><b>Column 1 List contains " & recCnt1 & " items</b><br>"
'ListColumn(rsCol1)

'Response.write "<br><b>Column 2 List contains " & recCnt2 & " items</b><br>"
'ListColumn(rsCol2)

If Err.Number <> 0 Then
    Response.Write "An error has occurred!<br>"
    Response.Write "Error number: " & Err.number & "<br>"
    Response.Write "Error description: " & Err.description & "<br>"

ElseIf rsGraph.EOF Then
    Response.Write "Criteria too restrictive. Empty recordset returned."
End If

%>

<html>
<head>
<title> Mishap Graph </title>
<meta name="Generator" content="Microsoft FrontPage 4.0">
<meta name="Author" content="">
<meta name="Keywords" content="">
<meta name="Description" content="">

```

```

<Link rel="stylesheet" type="text/css" href="../scripts/style.css">

<SCRIPT language="javascript">
<!-- //
    isDOM = (document.getElementById) ? true : false;
    isNS = (navigator.appName == "Netscape") ? true : false;
    isIE = (document.all) ? true : false;
    isIE4 = isIE && !isDOM;
    isMac = (navigator.appVersion.indexOf("Mac") != -1);
    isIE4Mac = isIE4 && isMac;
    isNS6 = isDOM && isNS;
    notNS6 = isNS && !isDOM;

    var counter=0;                // this will tell me how many z elements there are
    var graphMe = new Array();
    var yMax = new Array();
    var graphOutput;
    var legendOutput;
    var myColors = new Array
("black","blue","red","white","yellow","green","orange","dark_orange",
"dark_green","dark_red","dark_blue");

    function colorMe()
    {
        if(colorMe.arguments[0] != 'next')
        {
            colorIndex=colorMe.arguments[0];
            myColor=myColors[colorIndex];
            colorIndex++;
            if (colorIndex==myColors.length){colorIndex=0;}
            return myColor;
        }
        else
        {
            myColor=myColors[colorIndex];
            colorIndex++;
            if (colorIndex==myColors.length){colorIndex=0;}
            return myColor;
        }
    }
    function rowDefined(x,y,z)
    {
        this.x = x;
        this.y = y;
        this.z = z;
        this.xElements = x.length;
        this.yElements = y.length;
        this.yMax = 0;
        for (i=0;i<y.length;i++)
        {
            this.yMax = Math.max(this.yMax,y[i]);
        }
        yMax[counter]=this.yMax;
    }
    function createRow(x,y,z)
    {
        graphMe[counter] = new rowDefined(x,y,z);
        counter++;
    }
    function graphIt()

```

```

{
<%      dim xvalue,xcommastring,countcommastring,xarray,yarray
        xcommastring = ""
        countcommastring = ""
        for each xvalue in arLongerList
            if xcommastring = "" then
                xcommastring = "" & xvalue & ""
            else
                xcommastring = xcommastring & "," & "" & xvalue & ""
            end if
        next
        xarray = new Array(<%=xcommastring%>);
<%      Do While NOT rsGraph.EOF
        zValue = rsGraph.Fields(zaxis)
        countcommastring = ""
        for arrayRow = 0 to (recCnt2 - 1)
            if (rsGraph.EOF) then
                if countcommastring = "" then
                    countcommastring = "0"
                else
                    countcommastring = countcommastring & ",0"
                end if
            elseif (rsGraph.Fields(xaxis) <> arLongerList(0,arrayRow)) OR
(rsGraph.Fields(zaxis) <> zValue) then
                if countcommastring = "" then
                    countcommastring = "0"
                else
                    countcommastring = countcommastring & ",0"
                end if
            else
                if countcommastring = "" then
                    countcommastring = rsGraph.Fields("Count")
                else
                    countcommastring = countcommastring & "," &
rsGraph.Fields("Count")
                end if
                rsGraph.Movenext
            end if
        next
        yarray = new Array(<%=countcommastring%>);
        createRow(xarray,yarray,"<%=zValue%>");
<%      Loop
        //end while loop of ASP *****
        var colorIndex = 0;
        var myColor;
        var myYSCALE = new Array (1,2,3,4,5,6,7,8,9,10,15,20,25,30,40,50,75,
100,150,200,250,500); // increment choices for graph scale
        var whichColor = 0; // initialize first color choice, rotates
                                //in that order [index value]
        var barBorder = 1; // in pixels, adds a black border around
                                //the image or each bar
        var barSpace = 12; // in pixels, space between bars within a row
        var barDepth = 10; // in pixels, space between rows (y)
        var barOffset = 5; //in pixels, projection to give 3D effect (x)
        var barWidth = 25; // in pixels, width of color in bar...
                                // barBorder is extra pixels
        var graphTitle = 'Mishap Counts by <%=xLabel%> and <%=zLabel%>';
        var xLabel = "<%=xLabel%>"; // write with ASP *****
        var zLabel = "<%=zLabel%>"; // write with ASP *****
        var yLabel = 'COUNT';

```

```

// var graphHeight = z.length*(barDepth+barBorder)+330;
// in pixels,
var graphHeight = 3*(barDepth+barBorder)+330;           // in pixels,
var graphWidth = 650;
var zIndex = 0;           // starting z-index for graph
var graphBGcolor = "#eeeeee";           // background color in hex form
var graphYMIN=0;           // currently just 0
var graphYMAX=0;           // first set to 0, then found to be max of
                           //all Y values
var graphROWS=counter;     // depth of 3-d graph number of z-values
var graphCOLS=xarray.length; // width of 3-d graph number of x-values
for (i=0;i<Max.length;i++) // determines maximum height
{
    graphYMAX = Math.max(graphYMAX,yMax[i]);
}
for (i=0;i<myYSCALE.length;i++) // determines YSCALE increments
{
    if(myYSCALE[i]==Math.max((Math.ceil(graphYMAX/10)),myYSCALE[i]))
    {
        var YSCALE=myYSCALE[i];
        break;
    }
}
var verticalProjection = 10+(graphROWS-1)*barDepth;
//front edge + #rows * barDepth + (#rows-1)*barOffset
var verticalSpace = graphHeight - verticalProjection;
// amount of vertical space available to actually stretch the back bars
// get in divisible by 10 form
verticalSpace = 10*(Math.floor(verticalSpace/10));
var stretchFactor = verticalSpace/(YSCALE*10);
if (graphWidth==0)
{
    graphWidth = 5+graphCOLS*(barWidth+2*barBorder)+(graphCOLS-
1)*barSpace+10+(graphROWS-1)*barOffset;           // left edge + #cols * barWidth w/
border + (#cols-1)*barSpace + (#rows-1)*barOffset + right edge + very back right edge
}
else
{
    barWidth=Math.floor((((graphWidth-10-((graphCOLS-1)*barSpace))-10-
((graphROWS-1)*barOffset))-10)/graphCOLS)-(2*barBorder));           // if
graphWidth is set, reprogram barWidth to fit viewing area

    // because we rounding, let's recalculate the actual graphWidth
    graphWidth = 10+graphCOLS*(barWidth+2*barBorder)+(graphCOLS-
1)*barSpace+10+(graphROWS-1)*barOffset+10; // left edge + #cols * barWidth w/ border +
(#cols-1)*barSpace + (#rows-1)*barOffset + right edge + very back right edge
}

var backgroundWidth = 5+graphCOLS*(barWidth+2*barBorder)+(graphCOLS-
1)*barSpace+5;           // left edge + #cols * barWidth w/ border + right edge
// create GRAPH
graphOutput='';
graphOutput+="

145


```

```

        graphOutput+="!['+xLabel+''+graphMe[i].x[j]+'';'+zLabel+''+graphMe[i].z+'';'+yLabel+''+graphMe[i]
].y[j]+'\](\"images/white.gif\")

```

```

        myLeft=rememberMyLeft;
        rememberBaseTop+=barDepth;
        rememberBaseLeft=rememberBaseLeft-barOffset;
        rememberMyTop += barDepth;
    }
    zIndex++;
    myLeft +=5;
    for (i=0;i<xarray.length;i++)
    {
        graphOutput+='<span id="x'+i+'+_legend" style="position: absolute; left:
' +(myLeft+5)+'px; top: ' +(rememberBaseTop)+'px; z-index: ' +zIndex+'; font-size: 8pt;
font-weight: bold;">';
        graphOutput+=(xarray[i]);
        graphOutput+='</span>';
        myLeft += barWidth+barSpace+2*barBorder;
    }
    graphOutput+='</div>';

    //create LEGEND
    myColor=colorMe(whichColor);
    myTEMPtop=25+graphHeight;
    legendOutput+='<span id="legend" style="position: absolute; top: 0px; left:
0px; z-index: 1;">';
    legendOutput+='<table border=1 cellpadding=3 cellspacing=0>';

    //write Header of Legend
    legendOutput+='<tr>';
    legendOutput+='<td bgcolor="#ffffff"><b>';
    legendOutput+='LEGEND';
    legendOutput+='</b></td>';
    legendOutput+='<td colspan='+graphCOLS+' bgcolor="#cccccc"><b>';
    legendOutput+=xLabel; // this needs to be pulled from ASP
    legendOutput+='</b></td>';
    legendOutput+='</tr>';
    legendOutput+='<tr>';
    legendOutput+='<td bgcolor="#eeeeee"><b>';
    legendOutput+=zLabel;
    legendOutput+='</b></td>';
    for (j=0;j<graphCOLS;j++)
    {
        legendOutput+='<td bgcolor="#cccccc"><b>';
        legendOutput+=xarray[j];
        legendOutput+='</b></td>';
    }
    legendOutput+='</tr>';
    for (i=0;i<graphROWS;i++)
    {
        legendOutput+='<tr>';
        legendOutput+='<td align="left" bgcolor="#eeeeee">';
        legendOutput+='<b>';
        legendOutput+=graphMe[i].z;
        legendOutput+='</b></td>';
        for (j=0;j<graphCOLS;j++)
        {
            legendOutput+='<td align="center">';
            legendOutput+=graphMe[i].y[j];
            legendOutput+='</td>';
        }
        legendOutput+='</tr>';
    }

```



```
        conn.close
    set conn=nothing
' End if
%>
```

grouping_select.asp

Grouping_select.asp allows the user to select which categories of data to display on the x-axis and z-axis of the graph display.

```
<%@ Language=VBScript %>
<%Option Explicit%>
<%Response.Buffer=false%>

<!-- #include FILE = "../common/adovbs.inc" -->
<!-- #include FILE = "../common/connection.asp" -->

<%
Dim cmd                'command object
Set cmd = Server.CreateObject("ADODB.Command")
cmd.ActiveConnection = conn
%>

<html>
<head>
  <SCRIPT LANGUAGE="JavaScript">
    <!-- //
    function getSize()
    {
      document.groupForm.screenx.value = screen.availwidth;
      document.groupForm.screeny.value = screen.availheight;
    }
    function graphonly()
    {
      document.groupForm.showtable.value = "false";
      document.groupForm.submit()
    }
    function graphandtable()
    {
      document.groupForm.showtable.value = "true";
      document.groupForm.submit()
    }
    // -->
  </SCRIPT>
  <meta http-equiv="Content-Language" content="en-us">
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  <meta name="GENERATOR" content="Microsoft FrontPage 4.0">
  <meta name="ProgId" content="FrontPage.Editor.Document">
  <title>HFACS-ME Graph</title>
  <link rel="stylesheet" type="text/css" href="../scripts/style.css">
  <meta name="Microsoft Border" content="none">
</head>

<body onLoad="getSize()">
<!-- #include FILE = "../common/header.htm" -->
<div align="center">
<table border="0" width="700">
  <tr>
    <td><font size="4">Grouping Selection Page</font></td>
  </tr>
</table>
</div>
```

```

        <td><font size="2">Use the drop down boxes below to select the desired Grouping
for the dataset.&nbsp;   Primary Grouping will group all data by the selected category
and Secondary Grouping will group data within each primary category. &nbsp;  </font><p>
    </td>
</tr>
</table>
</div>
<form method="post" action="graphpage.asp" style="text-align: left" name="groupForm">
<div align="center">
    <center>
<table border="0" cellpadding="0" cellspacing="0" border="0" width="500">
<!-- #include FILE = "../common/column_select_INC.asp" -->
    <tr>
        <td valign="bottom" colspan="6">&nbsp;  
        </td>
    </tr>
    <tr>
        <td valign="bottom" colspan="6"><p align="center">
            <input type="button" value="Show Graph"
onClick="javascript:graphonly()">&nbsp;  &nbsp;  
            <br><br><input type="button" value="Show Graph and Data Table"
onClick="graphandtable()">&nbsp;  &nbsp;  
            <br><br><input type="reset" value="Reset" name="Reset">
        </p>
        </td>
    </tr>
    <tr>
        <td valign="bottom" colspan="3"><b>&nbsp;  </b>
        </td>
        <td valign="bottom" colspan="3">
        </td>
    </tr>
</table>
</center>
</div>
<input type="hidden" name="cboAircraft" value="<%=Request.Form("cboAircraft")%>">
<input type="hidden" name="cboType" value="<%=Request.Form("cboType")%>">
<input type="hidden" name="cboClass" value="<%=Request.Form("cboClass")%>">
<input type="hidden" name="cboLocation" value="<%=Request.Form("cboLocation")%>">
<input type="hidden" name="cboService" value="<%=Request.Form("cboService")%>">
<input type="hidden" name="cboYear" value="<%=Request.Form("cboYear")%>">

<input type="hidden" name="cbo1stLevelFactors"
value="<%=Request.Form("cbo1stLevelFactors")%>">
<input type="hidden" name="cbo2ndLevelFactors"
value="<%=Request.Form("cbo2ndLevelFactors")%>">
<input type="hidden" name="cbo3rdLevelFactors"
value="<%=Request.Form("cbo3rdLevelFactors")%>">
<input type="hidden" name="screenx" value="">
<input type="hidden" name="screeny" value="">
<input type="hidden" name="showtable" value="false"
</form>
<%
'Close connections
conn.close
set conn=nothing
%>
<!-- #include FILE = "../common/footer.htm" -->
<p>
</body>

```

showGraphTable.asp

ShowGraphTable.asp displays the data table associated with the graph output. This page opens in a pop-up window if the user selects "Display Graph and Data Table" on the grouping_select.asp page.

```
<%@ Language=VBScript %>
<%Option Explicit%>
<%
    dim sTablecontents,sTabletitle
    sTablecontents = request("tablecontents")
    sTabletitle = request("tabletitle")
%>
<html>
<head>
<title> Mishap Table </title>
<SCRIPT language="javascript">
    <!-- //
    // -->
</SCRIPT>
</head>
<body>
    <div id="graph" style="position: absolute; top: 10px; left: 10px;">
        <table border=0>
            <tr valign=bottom>
                <td bgcolor="#ffffff">
                    <h3><u><%=sTabletitle%></u></h3></td>
                </tr>
            </table>
        </div>
        <div id="graph" style="position: absolute; top: 60px; left: 10px;">
            <%=sTableContents%>
        </div>
    </body>
</html>
```

MISHAP SUBDIRECTORY

query.asp

Query.asp generates the Mishap Data display. If no user criteria is passed to the page via Request.Form variables (including hidden variables) then the page displays only the nine criteria select drop-down boxes and the Submit button. If criteria are provided by the user, the page dynamically builds the SQL string needed to "pre-filter" the mishap data before running the stored procedure to generate the table data. The resulting table displays basic mishap data including: MishapID, Aircraft type, Mishap Type, Mishap Class, Location Service and Mishap Date. A hyperlink is automatically created for each MishapID. Each hyperlink passes the corresponding MishapID to the expanded.asp page, using a getstring variable.

```
<%@ Language=VBScript %>
<%Option Explicit%>
<%Response.Buffer=true%>
<%
Dim cmd          'command object
Dim rsMishaps    'recordset for Mishap Breakdown table counts
Dim strSP_Call   'string to hold stored procedure name
Dim strParam     'string to hold stored procedure parameters
Dim blnMultiple  'bool to indicate multiple criteria
Dim blnFactors   'bool to indicate multiple criteria
Dim tempstring   'string to hold string values from multiselect dropdown boxes
Dim m_year       'string to hold year values from multiselect
Dim strSelect    'string to hold SQL SELECT
Dim intCount     'integer to hold number of Mishap records returned
%>
<!-- #include FILE = "../common/adovbs.inc" -->
<!-- #include FILE = "../common/connection.asp" -->
<%
Set cmd = Server.CreateObject("ADODB.Command")
cmd.ActiveConnection = conn
%>
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Table view</title>
<link rel="stylesheet" type="text/css" href="../scripts/style.css">
<meta name="Microsoft Border" content="none">
</head>

<body>
<!-- #include FILE = "../common/header.htm" -->

<div align="center">
  <center>
<table border="0" cellspacing="0" cellpadding="0" width="700">
  <tr>
    <td><font size="4">Mishap Data Selection Page</font></td>
  </tr>
```

```
|  |
| --- |
| Use the drop down boxes below to select the desired Mishap Criteria and Factors Criteria to include in the dataset. Multiple items within a particular criteria (ie. Aircraft Type = F14, F18) may be selected by holding down the Ctrl button and left clicking on the desired items. This will result in a query that will return data matching ANY of the selected items.  If criteria are selected in several criteria boxes (ie. Aircraft type and Mishap Class), the resulting dataset will be only those records that match ALL criteria. For example, if the user selects Aircraft Type=F18, F14 and Mishap Class=A, then the database will return all records involving either a F14 or F18 and resulted in a Class A mishap. |

```

```

<form method="POST" action="query.asp" style="text-align: left">
<!-- #include FILE = "../common/criteria_dropdown_INC.asp" -->
<div align="center">
<table border="0" cellspacing="0" cellpadding="0" width="700">
|  |
| --- |
| &nbsp; |
| |                                                                                                                                        |  |  |  |  |  | |----------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|--| | <input name="QuerySubmit" type="submit" value="Submit Query"/> <input type="reset" value="Reset"/> <input type="button" value="Back"/> |  |  |  |  |  | |----------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|--| |

```

```

<%
If Request.form("QuerySubmit")<>"" then
strParam=""

blnMultiple = False
blnFactors = False

'Build parameter list with Mishap Factors drop-down selections (NULL 'values not
stored)

tempstring = Prepstring(Request.Form("cboAircraft"))
If tempstring <> "" Then
    strParam = "tblMishaps.Aircraft_FK IN ('" & tempstring & "')"
    blnMultiple=True
End if

tempstring = Prepstring(Request.Form("cboType"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
    End if
    strParam = strParam & "tblMishaps.Type_FK IN ('" & tempstring & "')"
    blnMultiple=True

```

```

End If

tempstring = Prepstring(Request.Form("choClass"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
    End if
    strParam = strParam & "tblMishaps.Class_FK IN ('" & tempstring & "')"
    blnMultiple=True
End If

tempstring = Prepstring(Request.Form("choLocation"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
    End if
    strParam = strParam & "tblMishaps.LocationID_FK IN ('" & tempstring & "')"
    blnMultiple=True
End If

tempstring = Prepstring(Request.Form("choService"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
    End if
    strParam = strParam & "tblMishaps.OrgID_FK IN ('" & tempstring & "')"
    blnMultiple=True
End If

'-----Call multi-select function-----
'multiple select of an integer
m_year = Request.Form("choYear")
If m_year <> "" Then
    Do While Left(m_year,1)=", "
        m_year=Trim(Mid(m_year,2))
    Loop

    If blnMultiple Then
        strParam = strParam & " AND "
    End if

    strParam = strParam & "Year(DateAdd(month,3,tblMishaps.MishapDate)) IN (" & m_year
& ")")
    blnMultiple=True
End If

tempstring = Prepstring(Request.Form("cho1stLevelFactors"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "
    End if
    strParam = strParam & "tblFactors.[1stLevelCode] IN ('" & tempstring & "')"
    blnMultiple=True
    blnFactors=True
End If

tempstring = Prepstring(Request.Form("cho2ndLevelFactors"))
If tempstring <> "" Then
    If blnMultiple Then
        strParam = strParam & " AND "

```



```

        End if
        strParam = strParam & "tblFactors.[2ndLevelCode] IN ('" & tempstring & "')"
        blnMultiple=True
        blnFactors=True
    End If

    tempstring = Prepstring(Request.Form("cbo3rdLevelFactors"))
    If tempstring <> "" Then
        If blnMultiple Then
            strParam = strParam & " AND "
        End if
        strParam = strParam & "tblFactors.[3rdLevelCode] IN ('" & tempstring & "')"
        blnMultiple=True
        blnFactors=True
    End If

    strSelect = "SELECT DISTINCT MishapID, Aircraft_FK as Aircraft, Type_FK as Type,
    Class_FK as Class,LocationID_FK"
    strSelect = strSelect & " as Location, OrgID_FK as Service, MishapDate as Date FROM
    tblMishaps"

    If blnFactors Then
        strSelect = strSelect & " INNER JOIN tblMishapFactors ON tblMishaps.MishapID
    = tblMishapFactors.MishapID_FK"
        strSelect = strSelect & " INNER JOIN tblFactors ON
    tblMishapFactors.[3rdLevelCode_FK] = tblFactors.[3rdLevelCode]"
    End if

    strSelect = strSelect & " WHERE NOT(tblMishaps.DatabaseType = 'C')"
```

```

    If blnMultiple Then
        strSelect = strSelect & " AND " & Trim(strParam)
    End if

    cmd.CommandText = strSelect
    conn.CursorLocation = adUseClient
    Set rsMishaps=cmd.Execute
    intCount=rsMishaps.recordcount
    If Err.Number <> 0 Then
        Response.Write "An error has occurred!<br>"
        Response.Write "Error number: " & Err.number & "<br>"
        Response.Write "Error description: " & Err.description & "<br>"
    ElseIf rsMishaps.EOF Then
        Response.Write "<center><b><font color=""#FF0000"">Criteria too restrictive. No
    matching records found.</font></b></center>"
    Else
        BuildTable rsMishaps,intcount
    End If

End if

'Close connections
set rsMishaps=nothing
conn.close
set conn=nothing

Sub BuildTable(rsMishaps,count)
%>

```

```

<div align="center">
<center>
<table border="1" cellpadding="0" cellspacing="0" width="700">
  <tr>
    <td align="center" bgcolor="#FFFFFF" width="700" colspan="7">
      <h2><font color="#FF0000"><%=count%></font> Mishaps matched input
criteria</h2>
      <font color="#0000FF">Click on Mishap ID to view detailed Mishap Factor
data</font><br>
    </td>
  </tr>
  <tr>
    <td align="center" bgcolor="#FFFFFF" width="99"><b>Mishap ID</b></td>
    <td align="center" bgcolor="#FFFFFF" width="99"><b>Aircraft</b></td>
    <td align="center" bgcolor="#FFFFFF" width="100"><b>Type</b></td>
    <td align="center" bgcolor="#FFFFFF" width="100"><b>Class</b></td>
    <td align="center" bgcolor="#FFFFFF" width="100"><b>Location</b></td>
    <td align="center" bgcolor="#FFFFFF" width="100"><b>Service</b></td>
    <td align="center" bgcolor="#FFFFFF" width="100"><b>Date</b></td>
  </tr>
  <tr>
    <% Do While NOT rsMishaps.EOF%>
      <td align="center" bgcolor="#FFFFFF" width="99">
        <a href="expanded.asp?MishapID=<%=rsMishaps.Fields("MishapID")%>">
          <%=rsMishaps.Fields("MishapID")%></a></td>
      <td align="center" bgcolor="#FFFFFF"
width="99"><%=rsMishaps.Fields("Aircraft")%></td>
      <td align="center" bgcolor="#FFFFFF"
width="100"><%=rsMishaps.Fields("Type")%></td>
      <td align="center" bgcolor="#FFFFFF"
width="100"><%=rsMishaps.Fields("Class")%></td>
      <td align="center" bgcolor="#FFFFFF"
width="100"><%=rsMishaps.Fields("Location")%></td>
      <td align="center" bgcolor="#FFFFFF"
width="100"><%=rsMishaps.Fields("Service")%></td>
      <td align="right" bgcolor="#FFFFFF"
width="100"><%=rsMishaps.Fields("Date")%></td>
    </tr>
    <% 'get next record
      rsMishaps.Movenext
    Loop
  %>
</table>
<%
rsMishaps.close
End Sub

Function Prepstring(string)
  Do While Left(string,1)=","
    string=Trim(Mid(string,2))
  Loop
  If string <> "" Then
    string = Replace( string, ", ", "','") ' add apostrophes to CSV string
  End if
  Prepstring = string
End Function
%>
</form>
<!-- #include FILE = "../common/footer.htm" -->

```

expanded.asp

Expanded.asp generates a detailed Mishap Data display. The appropriate MishapID is retrieved from either the Request.Form or Querystring collections and is passed as a parameter to the spAllMishaps and the spFactorDetails stored procedures. The spAllMishaps procedure returns the basic mishap data and the spFactorDetails procedure returns all factors associated with the mishap. There is a checkbox and button that allow the user to toggle between summary (short) descriptions and long descriptions for the mishap and factor descriptions.

```
<%@ Language=VBScript%>
<%Option Explicit%>
<%Response.Buffer=False%>

<!-- #include FILE = "../common/adovbs.inc" -->
<!-- #include FILE = "../common/connection.asp" -->
<%
dim intMishapID
Dim Cmd
Dim rsMishap
Dim rsFactor
Dim strLong

If (Request.form("MishapID") <> "") Then
    intMishapID = Cint(Request.form("MishapID"))
ElseIf (Request.Querystring("MishapID") <> "") Then
    intMishapID = Request.Querystring("MishapID")
Else
    Response.write "Error. No Mishap ID identified"
End if

If (Request.Form("chkLongDesc") = "on") Then
    strLong = "True"
Else
    strLong = "False"
End If

Set cmd = Server.CreateObject("ADODB.Command")
cmd.ActiveConnection = conn

'Get data to populate Mishap Data
cmd.CommandText = "spAllMishaps @MishapID=" & intMishapID
Set rsMishap = cmd.Execute

'Get data to populate the Mishap Factors data
cmd.CommandText = "spFactorDetail @MishapID=" & intMishapID
Set rsFactor = cmd.Execute
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Mishap Details</title>
</head>

<body>
```

```

<!-- #include FILE = "../common/header.htm" -->

<form target="_self" method="POST" action="expanded.asp">
<div align="center">
<center>
<table BORDER="0" width="700" height="154" cellspacing="0">
  <tr>
    <td height="19" width="150" valign="middle" align="left"><b><font size="2">Mishap
ID:</font></b></td>
    <td height="19" width="110" valign="middle" align="left"><font
size="2"><%=rsMishap.Fields("MishapID")%></font></td>
    <td height="19" width="100" valign="middle" align="left"></td>
    <td height="19" width="100" valign="middle" align="left"></td>
    <td height="19" valign="middle" align="left"></td>
    <td height="19" valign="middle" align="left"></td>
  </tr>
  <tr>
    <td height="19" width="150" valign="middle" align="left"><b><font size="2">Mishap
Date:</font></b></td>
    <td height="19" width="110" valign="middle" align="left"><font
size="2"><%=rsMishap.Fields("MishapDate")%></font></td>
    <td height="19" width="100" valign="middle" align="left"><b><font
size="2">Aircraft:</font></b></td>
    <td height="19" width="100" valign="middle" align="left"><font
size="2"><%=rsMishap.Fields("Aircraft_FK")%></font></td>
    <td height="19" valign="middle" align="left"><b><font
size="2">&nbsp;&nbsp;&nbsp;&nbsp;</font></b></td>
    <td height="19" valign="middle" align="left"><b> <font size="2"> <input
type="checkbox" name="chkLongDesc" <%If strLong then response.write " checked"%>
Display Long Descriptions</font></b></td>
  </tr>
  <tr>
    <td height="18" width="150" valign="middle" align="left"><b><font
size="2">Class:</font></b></td>
    <td height="18" width="110" valign="middle" align="left"><font
size="2"><%=rsMishap.Fields("Class_FK")%></font></td>
    <td height="18" width="100" valign="middle" align="left"><b><font
size="2">Type:</font></b></td>
    <td height="18" width="100" valign="middle" align="left"><font
size="2"><%=rsMishap.Fields("Type_FK")%></font></td>
    <td height="37" valign="middle" rowspan="2" align="left"></td>
    <td height="37" valign="middle" rowspan="2" align="left"><b>
<font size="2">
<input type="submit" value="Refresh" name="Refresh"></font></b></td>
  </tr>
  <tr>
    <td height="19" width="150" valign="middle" align="left"><b><font
size="2">Service:</font></b></td>
    <td height="19" width="110" valign="middle" align="left"><font
size="2"><%=rsMishap.Fields("OrgID_FK")%></font></td>
    <td height="19" width="100" valign="middle" align="left"><b><font
size="2">Location:</font></b></td>
    <td height="19" width="100" valign="middle" align="left"><font
size="2"><%=rsMishap.Fields("MishapLocation")%></font></td>
  </tr>
  <tr>
    <td height="20" width="150" valign="baseLine" align="left"><font
size="2">&nbsp;&nbsp;&nbsp;&nbsp;</font></td>
    <td colspan="5" height="20" valign="baseLine" align="left">
<font size="2">

```

[illegible]

```

        <td width="100" height="19" valign="middle" align="center" style="border-left: 1
solid #C0C0C0; border-bottom: 1 solid #C0C0C0"><b><font size="2">1st
Level</font></B></TD>
        <td width="100" height="19" valign="middle" align="center" style="border-bottom: 1
solid #C0C0C0"><b><font size="2">2nd Level</font></B></TD>
        <td width="100" height="19" valign="middle" align="center" style="border-right: 1
solid #C0C0C0; border-bottom: 1 solid #C0C0C0"><b><font size="2">3rd
Level</font></B></TD>
    <%
Do While Not rsFactor.EOF
%>
    <tr>
        <td width="400" height="40" style="border: 1 solid #C0C0C0">
            <font size="2">
                <%
                If Not strLong Then
                    response.write rsFactor.Fields("FactorSummary")
                Else
                    response.write rsFactor.Fields("FactorDescription")
                End if
                %>
            </font>
        </td>
        <td width="100" height="40" style="border-top: 1 solid #C0C0C0; border-bottom: 1
solid #C0C0C0" align="center"><font
size="2"><%=rsFactor.Fields("1stLevelDesc")%></font></td>
        <td width="100" height="40" style="border-top: 1 solid #C0C0C0; border-bottom: 1
solid #C0C0C0" align="center"><font
size="2"><%=rsFactor.Fields("2ndLevelDesc")%></font></td>
        <td width="100" height="40" style="border-right: 1 solid #C0C0C0; border-top: 1
solid #C0C0C0; border-bottom: 1 solid #C0C0C0" align="center"><font
size="2"><%=rsFactor.Fields("3rdLevelDesc")%></font></td>

        <%
        rsFactor.MoveNext
        Loop
    %>
</tr>
</table>
</center>
</div>
<input type="hidden" name="MishapID" value=<%=intMishapID%>>
</form>

<p align="center"><input type="button" value="Back to Table View"
onClick="self.history.back()"></p>
</body>

```

reportcall.asp

[illegible]

reports.asp

The reports.asp page retrieves the querystring value passed by the reportcall.asp page and uses a Select Case statement to determine which stored procedure to call to generate the report data. Once the data is returned from the stored procedure, the report.asp page formats it into individual report tables based on the category subdivisions. Hyperlinks are created at the top of each report page to aid the user in navigating all reports.

```
<%@ Language=VBScript %>
<%Option Explicit%>
<%Response.Buffer=true%>
<%
Dim cmd                'command object
Dim rsCodes            'recordset for Mishap Breakdown table counts
Dim intTotal          '
Dim strCmd             '
Dim strReportType
Dim strFieldName
Dim strGroupName
Dim strLinks

'Declare all Mishap Count variables
Dim intMG
Dim intORG, intSUP, intPRO, intDOC, intDES
Dim intRES, intIDQ, intOPS, intPRB, intMIS
Dim intMC
Dim intMED, intMNT, intPHY, intLIM
Dim intCRW, intCOM, intASS, intADA
Dim intRDY, intTRG, intCRT, intINF
Dim intWC
Dim intENV, intLGT, intWXE, intEHZ
Dim intEQP, intDMG, intUNA, intDUC
Dim intWRK, intCON, intOBS, intINA
Dim intMA
Dim intERR, intATT, intJDG, intKNW, intSKL
Dim intVIO, intROU, intIFC, intFLG, intEXC
%>
<!-- #include FILE = "../common/adovbs.inc" -->
<!-- #include FILE = "../common/connection.asp" -->
<head>
</head>
<%
Sub BuildTable(rsCodes, strLinks)

%>
<!-------Build Report Tables----->
<div align="center">

<%
Do While Not rsCodes.EOF

    'Avoid divide by zero error
    If rsCodes.Fields("TotalMishaps")=0 Then
        intTotal = 1
    Else
        intTotal = rsCodes.Fields("TotalMishaps")
```


End If

'store MG factor counts in local variables

```
intMG = rsCodes.Fields("MG")
intORG = rsCodes.Fields("ORG")
intSUP = rsCodes.Fields("SUP")
intPRO = rsCodes.Fields("PRO")
intDOC = rsCodes.Fields("DOC")
intDES = rsCodes.Fields("DES")
intRES = rsCodes.Fields("RES")
intIDQ = rsCodes.Fields("IDQ")
intOPS = rsCodes.Fields("OPS")
intPRB = rsCodes.Fields("PRB")
intMIS = rsCodes.Fields("MIS")
```

'store MC factor counts in local variables

```
intMC = rsCodes.Fields("MC")
intMED = rsCodes.Fields("MED")
intMNT = rsCodes.Fields("MNT")
intPHY = rsCodes.Fields("PHY")
intLIM = rsCodes.Fields("LIM")
intCRW = rsCodes.Fields("CRW")
intCOM = rsCodes.Fields("COM")
intASS = rsCodes.Fields("ASS")
intADA = rsCodes.Fields("ADA")
intRDY = rsCodes.Fields("RDY")
intTRG = rsCodes.Fields("TRG")
intCRT = rsCodes.Fields("CRT")
intINF = rsCodes.Fields("INF")
```

'store WC factor counts in local variables

```
intWC = rsCodes.Fields("WC")
intENV = rsCodes.Fields("ENV")
intLGT = rsCodes.Fields("LGT")
intWXE = rsCodes.Fields("WXE")
intEHZ = rsCodes.Fields("EHZ")
intEQP = rsCodes.Fields("EQP")
intDMG = rsCodes.Fields("DMG")
intUNA = rsCodes.Fields("UNA")
intDUC = rsCodes.Fields("DUC")
intWRK = rsCodes.Fields("WRK")
intCON = rsCodes.Fields("CON")
intOBS = rsCodes.Fields("OBS")
intINA = rsCodes.Fields("INA")
```

'store MA factor counts in local variables

```
intMA = rsCodes.Fields("MA")
intERR = rsCodes.Fields("ERR")
intATT = rsCodes.Fields("ATT")
intJDG = rsCodes.Fields("JDG")
intKNW = rsCodes.Fields("KNW")
intSKL = rsCodes.Fields("SKL")
intVIO = rsCodes.Fields("VIO")
intROU = rsCodes.Fields("ROU")
intIFC = rsCodes.Fields("IFC")
intFLG = rsCodes.Fields("FLG")
intEXC = rsCodes.Fields("EXC")
```

```
strGroupName = rsCodes.Fields("'" & strFieldName & "'")
```

[illegible]

[illegible]


```

right-color: #C0C0C0; border-right-width: 1"><b>
<font size="1" face="MS Sans Serif"><%=FormatPercent((IntROU/intTotal),0)%>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</font></b></td>
</tr>
<tr>
<td width="30%" bgcolor="#DBDCAD" valign="top"><b>
<font size="1" face="MS Sans Serif">Infraction</font></b></td>
<td width="10%" bgcolor="#DBDCAD" align="center"><b>
<font size="1"><font face="MS Sans Serif"><%=IntINF%></font></font></b></td>
<td width="10%" bgcolor="#DBDCAD" align="right" style="border-
right-color: #C0C0C0; border-right-width: 1"><b>
<font size="1" face="MS Sans Serif"><%=FormatPercent((IntINF/intTotal),0)%>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</font></b></td>
</tr>
<tr>
<td width="30%" bgcolor="#DBDCAD" valign="top"><b>
<font size="1" face="MS Sans Serif">Flagrant</font></b></td>
<td width="10%" bgcolor="#DBDCAD" align="center"><b>
<font size="1"><font face="MS Sans Serif"><%=IntFLG%></font></font></b></td>
<td width="10%" bgcolor="#DBDCAD" align="right" style="border-
right-color: #C0C0C0; border-right-width: 1"><b>
<font size="1" face="MS Sans Serif"><%=FormatPercent((IntFLG/intTotal),0)%>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</font></b></td>
</tr>
<tr>
<td width="30%" bgcolor="#DBDCAD" valign="top"><b>
<font size="1" face="MS Sans Serif">Exceptional</font></b></td>
<td width="10%" bgcolor="#DBDCAD" align="center"><b>
<font size="1"><font face="MS Sans Serif"><%=IntEXC%></font></font></b></td>
<td width="10%" bgcolor="#DBDCAD" align="right" style="border-
right-color: #C0C0C0; border-right-width: 1"><b>
<font size="1" face="MS Sans Serif"><%=FormatPercent((IntEXC/intTotal),0)%>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</font></b></td>
</tr>
</table>
<br><span class="Lighttext">.: <a href="#top">back to top</a> :.<br><br><br>
</div>
<%
rsCodes.MoveNext
Loop
%>
</div>
</span>
<%
End Sub
%>
<!-- #include FILE = "../common/header.htm" -->
<%
Set cmd = Server.CreateObject("ADODB.Command")
cmd.ActiveConnection = conn

strReportType=Request.QueryString("Name")

Select Case strReportType
Case "Aircraft Type"
strCmd="spReport_By_Aircraft"
strFieldName="Aircraft_FK"
Case "Mishap Type"
strCmd="spReport_By_Type"
strFieldName="Type_FK"

```

```

Case "Mishap Class"
    strCmd="spReport_By_Class"
    strFieldName="Class_FK"
Case "Location"
    strCmd="spReport_By_Location"
    strFieldName="LocationID_FK"
Case "Organization"
    strCmd="spReport_By_Organization"
    strFieldName="Organization"
Case "Fiscal Year"
    strCmd="spReport_By_FiscalYear"
    strFieldName="Year"
Case Else
    Response.end
End Select

cmd.CommandText = strCmd
Set rsCodes=cmd.Execute

If Err.Number <> 0 Then
    Response.Write "An error has occurred!<br>"
    Response.Write "Error number:      " & Err.number & "<br>"
    Response.Write "Error description: " & Err.description & "<br>"

ElseIf rsCodes.EOF Then
    Response.Write "<center><b><font color='\"#FF0000\"'>"
    Response.Write "Criteria too restrictive. "
    Response.Write "No matching records found.</font></b></center>"

Else%>
    <center><font face="MS Sans Serif" size="5"><b>
Mishap Factors Report by <%=strReportType%></font></center></b>
<div align="center">
<%
    strLinks = ""
    Do While Not rsCodes.EOF
        strGroupName = rsCodes.Fields("'" & strFieldName & "'")
        strLinks = strLinks & " &nbsp;<a href=#" & strGroupName & ">" & strGroupName &
"</a>&nbsp;  ";
        rsCodes.MoveNext
    Loop
    rsCodes.MoveFirst
    BuildTable rsCodes, strLinks
End If

'Close connections
rsCodes.close
set rsCodes=nothing
conn.close
set conn=nothing
%><!-- #include FILE = "../common/footer2.htm" -->

```

CONTACT SUBDIRECTORY

index.asp

index.asp provides HFACS–ME Web point of contact information and email links to each point of contact.

```
<!--#include FILE = "../common/header.htm" -->

<div align="center">
  <center>

    <table border="0" cellpadding="5" cellspacing="0" bgcolor="#ffffff" width="500">
      <tr>
        <td align="left" valign="top" colspan="2">
          <p align="center"><font size="3"><b>The HFACS–ME Web is a prototype and as such, any
constructive feedback is welcomed.</b></font></p>
          <p align="center"><br>
            <font size="2">Comments relating to site design, content and
functionality.</font></p>
          <p align="center">
            <a href="mailto:arboex@nps.navy.mil?subject=HFACS–ME Web Feedback">
              <font size="2">HFACS–ME Webmaster</font></a></p>
          <p align="center"><br>
            <font size="2">Comments relating to website access and/or availability
            </font></p>
          <p align="center">
            <a href="mailto:gazolla@nps.navy.mil?subject=HFACS–ME Web Access/Availability">
              <font size="2">HFACS–ME Web Site Administrator</font></a></p>
          <p align="center"><br>
            <font size="2">Comments relating to the HFACS taxonomy and continuing research in
this area</font></p>
          <p align="center">
            <a href="mailto:Rfiglock@nps.navy.mil?subject=HFACS–ME Taxonomy/Research">
              <font size="2">HFACS–ME Taxonomy/Research</font></a></p>
          <p align="center">&nbsp;</p>
        </td>
      </tr>
    </table>

  </center>
</div>

<!--#include FILE = "../common/footer2.htm" -->
```

HOME SUBDIRECTORY

index.asp

Index.asp is the HFACS–ME Web homepage. It provides a brief description of the purpose of the HFACS–ME website and has a menu bar to enable the user to move to other parts of the website.

```
<html>
<!-- #include FILE = "../common/header.htm" -->

<div align="center">
<center>
<table border="0" cellpadding="5" cellspacing="0" bgcolor="#ffffff" width="700">
<tr>
    <td align="left" valign="top" colspan="2">
        
        <font size="2">The purpose of this site is to provide a web based safety
information system that will facilitate data collection, organization, query,
analysis, and reporting of maintenance errors that contribute to Naval Aviation
mishaps, equipment damage, and personnel injury, using the Human Factors Analysis and
Classification System – Maintenance Extension (HFACS–ME) taxonomy contained in OPNAV
3750.6R.</font><p>
        <font size="2">The ultimate goal is to allow authorized end users to
effortlessly access the centralized Aviation Mishap database and obtain valuable
information, which can then be used in training, hazard identification and trend
analysis in an effort to prevent future mishaps.</font>
        <p>
        &nbsp;
    </td>
</tr>
</table>
</center>
</div>
<div align="center">
<center>
<table border="0" cellpadding="0" cellspacing="0" bgcolor="#ffffff">
<tr>
    <td valign="top" align="center">
        <br><br>
        <span class="lighttext">
        :: <a href="#top">back to top</a> ::
        <br><br><br>
        <b><a href="http://www.nps.navy.mil/disclaimer/" target="_blank">Government
Disclaimer</a></b>
        <br><br>
        <br>Last Modified 8/29/2001
        </span><br><br></td>
</tr>
</table>
</center>
</div>
</html>
```

SCRIPTS SUBDIRECTORY

button_functions.js

Button_function.js is a Javascript file which defines client-side script to be executed when certain user-interaction events (related to user clicking on buttons/graphics) occur.

```
// Mouse Rollover Functions, by Boris Belobrad, 1999
// These functions work only when IMAGEon.src, IMAGEoff.src and
// IMAGEclk.src are defined in document. IMG tag must contain a //'name' attribute to
work correctly and must be the same as //IMAGE***.
// Function to 'activate' images.
function imgOver(imgName) {
    if (document.images && document[imgName]) {
        document[imgName].src = eval(imgName + "on.src");
        return true;
    } else {
        return false;
    }
}

// Function to 'deactivate' images.
function imgOut(imgName) {
    if (document.images && document[imgName]) {
        document[imgName].src = eval(imgName + "off.src");
        return true;
    } else {
        return false;
    }
}

// Function to 'click' images.
function imgClick(imgName) {
    if (document.images && document[imgName]) {
        document[imgName].src = eval(imgName + "clk.src");
        return true;
    } else {
        return false;
    }
}
```

img.js

Img.js is a Javascript file which defines client-side script to be executed when certain user-interaction events (related to mouseover of buttons/graphics) occur.

```
<!--
    if (document.images) {

Link0on = new Image();
    Link0off = new Image();
    Link0on.src="../images/home_on.gif";
    Link0off.src="../images/home_off.gif";

Link1on = new Image();
    Link1off = new Image();
    Link1on.src="../images/reports_on.gif";
    Link1off.src="../images/reports_off.gif";

Link2on = new Image();
    Link2off = new Image();
    Link2on.src="../images/mishap_data_on.gif";
    Link2off.src="../images/mishap_data_off.gif";

Link3on = new Image();
    Link3off = new Image();
    Link3on.src="../images/factor_analysis_on.gif";
    Link3off.src="../images/factor_analysis_off.gif";

Link4on = new Image();
    Link4off = new Image();
    Link4on.src="../images/graph_data_on.gif";
    Link4off.src="../images/graph_data_off.gif";

Link5on = new Image();
    Link5off = new Image();
    Link5on.src="../images/online_tutorial_on.gif";
    Link5off.src="../images/online_tutorial_off.gif";

Link6on = new Image();
    Link6off = new Image();
    Link6on.src="../images/help_on.gif";
    Link6off.src="../images/help_off.gif";

Link7on = new Image();
    Link7off = new Image();
    Link7on.src="../images/access_policy_on.gif";
    Link7off.src="../images/access_policy_off.gif";

Link8on = new Image();
    Link8off = new Image();
    Link8on.src="../images/contact_us_on.gif";
    Link8off.src="../images/contact_us_off.gif";

    }
//-->
```

style.css

Style.css defines Cascading Style Sheet implementation options used in HFACS–ME Web pages.

```
a          { color: #006699; text-decoration: Underline }
a:hover    { color: #0099CC; text-decoration: none }
a:visited  { color: #006699; text-decoration: Underline }
a:visited:hover{ color: #0099CC; text-decoration: none }

body       { font-family: MS Sans Serif; font-size: 8pt; color: #333333 }
td         { font-family: MS Sans Serif; font-size: 8pt; color: #333333 }

.bodytext  { font-family: MS Sans Serif; font-size: 8pt; line-height: 1.5em; color:
#333333 }
.smalltext { font-family: MS Sans Serif; font-size: 8pt; color: #999999 }
.lighttext { font-family: MS Sans Serif; font-size: 7pt; color: #999999 }
.rightfeatures { font-family: MS Sans Serif; font-size: 8pt; font-weight: Bold; color:
#ffffff }
.subheaders { font-family: MS Sans Serif; font-size: 8pt; font-weight: Bold; color:
#666666 }
.headers   { font-family: MS Sans Serif; font-size: 12pt; font-weight: Bold; color:
#666666 }
.select    {font-family: Arial; font-size: 8pt; color: #000000; font-weight:
medium;}
```

APPENDIX D. SQL SERVER 2000 CODE

A. SQL SERVER 2000 STORED PROCEDURES

sp1st Level Factors List

```
Alter Procedure sp1st_Level_Factors_List
```

```
As
```

```
SET nocount on
```

```
SELECT DISTINCT [1stLevelCode], [1stLevelDesc]  
FROM tblFactors  
WHERE NOT(tblFactors.[1stLevelCode] = 'UN')
```

```
return
```

sp2nd Level Factors List

```
Alter Procedure sp2nd_Level_Factors_List
```

```
As
```

```
SET nocount on
```

```
SELECT DISTINCT [2ndLevelCode], [2ndLevelDesc]  
FROM tblFactors  
WHERE NOT(tblFactors.[2ndLevelCode] = 'UNK')
```

```
return
```

sp3rd Level Factors List

```
Alter Procedure sp3rd_Level_Factors_List
```

```
As
```

```
SET nocount on
```

```
SELECT DISTINCT [3rdLevelCode], [3rdLevelDesc]  
FROM tblFactors  
WHERE NOT(tblFactors.[3rdLevelCode] = 'UNK')  
ORDER BY [3rdLevelDesc]
```

```
return
```


spAircraft_List

Alter Procedure **spAircraft_List**

```
As
SET nocount on

SELECT DISTINCT Aircraft_FK AS ACType
FROM tblMishaps
WHERE NOT(tblMishaps.DatabaseType = 'C')

return
```

spAllMishaps

Alter Procedure **spAllMishaps**

```
( @MishapID int = NULL )
```

```
As
SET nocount on

SELECT MishapID, MishapDate,
Aircraft_FK,
Class_FK,
Type_FK,
MishapLocation,
OrgID_FK,
ShortDescription,
LongDescription

FROM tblMishaps
INNER JOIN tblMishapLocation
ON tblMishaps.LocationID_FK = tblMishapLocation.MishapLocationID

WHERE MishapID=COALESCE(@MishapID, tblMishaps.MishapID)
ORDER BY MishapID

return
```

spMishap_Year

Alter Procedure **spMishap_Year**

As

SET **nocount** on

```
SELECT DISTINCT Year(DateAdd(month, 3, MishapDate)) AS MishapYear
FROM tblMishaps
WHERE Year(DateAdd(month, 3, MishapDate)) IS NOT NULL AND
      NOT(tblMishaps.DatabaseType = 'C')
```

return

spOrg_List

Alter Procedure **spOrg_List**

As

SET **nocount** on

```
SELECT DISTINCT OrgID_FK AS OrgID, OrgName
FROM tblMishaps
INNER JOIN tblOrganization
ON   tblMishaps.OrgID_FK = tblOrganization.OrgID
WHERE NOT(tblOrganization.DatabaseType = 'C')
ORDER BY OrgID Desc
```

return

spFactorDetail

Alter Procedure **spFactorDetail**

(@MishapID int = NULL)

As

SET **nocount** on

SELECT tblMishapFactors.FactorSummary,
tblMishapFactors.
FactorDescription,
tblFactors.[1stLevelDesc],
tblFactors.[2ndLevelDesc],
tblFactors.[3rdLevelDesc]

FROM tblMishapFactors

INNER JOIN tblMishaps

ON tblMishapFactors.MishapID_FK = tblMishaps.MishapID

INNER JOIN tblFactors

ON tblMishapFactors.[3rdLevelCode_FK] = tblFactors.[3rdLevelCode]

WHERE (tblMishaps.MishapID = COALESCE(@MishapID, tblMishapFactors.MishapID_FK))

ORDER BY tblFactors.[1stLevelDesc],

tblFactors.[2ndLevelDesc],

tblFactors.[3rdLevelDesc]

return

spMishapCount Filtered with Factors @query

```
Alter Procedure spMishapCount_Filtered_with_Factors_@query
( @query varchar(1000) = NULL )

As

SET nocount on

Exec(@query)

-----Build MishapCount resultset-----
SELECT
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
tblFactors.[1stLevelCode] = 'MG'))))) AS MG,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
tblFactors.[1stLevelCode] = 'MC'))))) AS MC,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
tblFactors.[1stLevelCode] = 'WC'))))) AS WC,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
tblFactors.[1stLevelCode] = 'MA'))))) AS MA,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
tblFactors.[2ndLevelCode] = 'ORG'))))) AS ORG,
```

```

(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
    FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
    WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
        tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'SUP'))))) AS SUP,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
    FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
    WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
        tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'MED'))))) AS MED,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
    FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
    WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
        tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'CRW'))))) AS CRW,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
    FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
    WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
        tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'RDY'))))) AS RDY,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
    FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
    WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
        tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'ENW'))))) AS ENW,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
    FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
    WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
        tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'EQP'))))) AS EQP,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
    FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors

```

```

WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
      tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
      tblFactors.[2ndLevelCode] = 'WRK')))) AS WRK,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
      SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
      FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
      WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
            tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'ERR')))) AS ERR,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
      SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
      FROM ##tblTemp_Filter_Table, tblFactors, tblMishapFactors
      WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND
            tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK] AND (
            tblFactors.[2ndLevelCode] = 'VIO')))) AS VIO,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
      SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
      FROM ##tblTemp_Filter_Table, tblMishapFactors
      WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'PRO')))) AS PRO,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
      SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
      FROM ##tblTemp_Filter_Table, tblMishapFactors
      WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'DOC')))) AS DOC,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
      SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
      FROM ##tblTemp_Filter_Table, tblMishapFactors
      WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'DES')))) AS DES,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
      SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
      FROM ##tblTemp_Filter_Table, tblMishapFactors
      WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'RES')))) AS RES,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
      SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
      FROM ##tblTemp_Filter_Table, tblMishapFactors
      WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (

```

```

tblMishapFactors.[3rdLevelCode_FK] = 'IDQ'))))) AS IDQ,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'OPS'))))) AS OPS,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'PRB'))))) AS PRB,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'MIS'))))) AS MIS,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'MNT'))))) AS MNT,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'PHY'))))) AS PHY,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'LIM'))))) AS LIM,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'COM'))))) AS COM,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table

```

```

WHERE (((#tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT #tblTemp_Filter_Table.MishapID
    FROM #tblTemp_Filter_Table, tblMishapFactors
    WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
        tblMishapFactors.[3rdLevelCode_FK] = 'ASS')))) AS ASS,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'ADA')))) AS ADA,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'TRG')))) AS TRG,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'CRT')))) AS CRT,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'INF')))) AS INF,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'LGT')))) AS LGT,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'WXE')))) AS WXE,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (

```



```

tblMishapFactors.[3rdLevelCode_FK] = 'EHZ')))) AS EHZ,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'DMG')))) AS DMG,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'UNA')))) AS UNA,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'DUC')))) AS DUC,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'CON')))) AS CON,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'OBS')))) AS OBS,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'INA')))) AS INA,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table
WHERE (((##tblTemp_Filter_Table.MishapID) In (
SELECT DISTINCT ##tblTemp_Filter_Table.MishapID
FROM ##tblTemp_Filter_Table, tblMishapFactors
WHERE ##tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
tblMishapFactors.[3rdLevelCode_FK] = 'ATT')))) AS ATT,
(
SELECT Count([MishapID])
FROM ##tblTemp_Filter_Table

```

```

WHERE (((#tblTemp_Filter_Table.MishapID) In (
    SELECT DISTINCT #tblTemp_Filter_Table.MishapID
    FROM #tblTemp_Filter_Table, tblMishapFactors
    WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
        tblMishapFactors.[3rdLevelCode_FK] = 'JDG')))) AS JDG,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'KNW')))) AS KNW,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'SKL')))) AS SKL,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'ROU')))) AS ROU,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'IFC')))) AS IFC,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'FLG')))) AS FLG,
(
    SELECT Count([MishapID])
    FROM #tblTemp_Filter_Table
    WHERE (((#tblTemp_Filter_Table.MishapID) In (
        SELECT DISTINCT #tblTemp_Filter_Table.MishapID
        FROM #tblTemp_Filter_Table, tblMishapFactors
        WHERE #tblTemp_Filter_Table.MishapID = tblMishapFactors.MishapID_FK AND (
            tblMishapFactors.[3rdLevelCode_FK] = 'EXC')))) AS EXC,
(
    SELECT Count(#tblTemp_Filter_Table.[MishapID])
    FROM #tblTemp_Filter_Table)
AS TotalMishaps;

return

```

spReport By Aircraft

Alter Procedure spReport_By_Aircraft

As

SET NOCOUNT ON

CREATE TABLE #nResult3

```
( Aircraft_FK varchar(255),  
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,  
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,  
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,  
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,  
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,  
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,  
  KNW int DEFAULT 0, LGT int DEFAULT 0, LIM int DEFAULT 0,  
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,  
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,  
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,  
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,  
  UNK int DEFAULT 0, WXE int DEFAULT 0 )
```

CREATE TABLE #nResult2

```
( Aircraft_FK varchar(255),  
  CRW int DEFAULT 0, ENV int DEFAULT 0, EQP int DEFAULT 0,  
  ERR int DEFAULT 0, MED int DEFAULT 0, ORG int DEFAULT 0,  
  RDY int DEFAULT 0, SUP int DEFAULT 0, UNK int DEFAULT 0,  
  VIO int DEFAULT 0, WRK int DEFAULT 0 )
```

CREATE TABLE #nResult1

```
( Aircraft_FK varchar(255),  
  MA int DEFAULT 0, MC int DEFAULT 0, MG int DEFAULT 0,  
  UN int DEFAULT 0, WC int DEFAULT 0 )
```

CREATE TABLE #nResultFinal

```
( Aircraft_FK varchar(255),  
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,  
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,  
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,  
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,  
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,  
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,  
  KNW int DEFAULT 0, LIM int DEFAULT 0, LGT int DEFAULT 0,  
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,  
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,  
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,  
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,  
  WXE int DEFAULT 0, CRW int DEFAULT 0, WRK int DEFAULT 0,  
  ENV int DEFAULT 0, EQP int DEFAULT 0, ERR int DEFAULT 0,  
  MED int DEFAULT 0, ORG int DEFAULT 0, RDY int DEFAULT 0,  
  SUP int DEFAULT 0, VIO int DEFAULT 0, MA int DEFAULT 0,  
  MC int DEFAULT 0, MG int DEFAULT 0, WC int DEFAULT 0 )
```

-----FOR THIRD LEVEL FACTORS

--Build a temp table and update the null values to 'None'

SELECT MishapID, [3rdLevelCode], Aircraft_FK INTO #nTemp3

```

FROM [wReport_By_Aircraft_3]

UPDATE #nTemp3
SET Aircraft_FK = 'None'
WHERE Aircraft_FK is null
--Now run the crosstab
INSERT #nResult3
EXEC dbo.rac @grpcol='Aircraft_FK', @pvtcol='[3rdLevelCode]', @transform='count(*)',
@from='#nTemp3', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

-----FOR SECOND LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID, [2ndLevelCode], Aircraft_FK INTO #nTemp2
FROM [wReport_By_Aircraft_2]

UPDATE #nTemp2
SET Aircraft_FK = 'None'
WHERE Aircraft_FK is null
--Now run the crosstab
INSERT #nResult2
EXEC dbo.rac @grpcol='Aircraft_FK', @pvtcol='[2ndLevelCode]', @transform='count(*)',
@from='#nTemp2', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

-----FOR FIRST LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID, [1stLevelCode], Aircraft_FK INTO #nTemp1
FROM [wReport_By_Aircraft_1]

UPDATE #nTemp1
SET Aircraft_FK = 'None'
WHERE Aircraft_FK is null
--Now run the crosstab
INSERT #nResult1
EXEC dbo.rac @grpcol='Aircraft_FK', @pvtcol='[1stLevelCode]', @transform='count(*)',
@from='#nTemp1', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'
-----

INSERT #nResultFinal
SELECT dbo.#nResult3.Aircraft_FK, dbo.#nResult3.ADA, dbo.#nResult3.ASS,
dbo.#nResult3.ATT, dbo.#nResult3.COM, dbo.#nResult3.CON, dbo.#nResult3.CRT,
dbo.#nResult3.DES, dbo.#nResult3.DMG, dbo.#nResult3.DOC, dbo.#nResult3.DUC,
dbo.#nResult3.EHZ, dbo.#nResult3.EXC, dbo.#nResult3.FLG, dbo.#nResult3.IDQ,
dbo.#nResult3.IFC, dbo.#nResult3.INA, dbo.#nResult3.INF, dbo.#nResult3.JDG,
dbo.#nResult3.KNW, dbo.#nResult3.LIM, dbo.#nResult3.LGT, dbo.#nResult3.MIS,
dbo.#nResult3.MNT, dbo.#nResult3.OBS, dbo.#nResult3.OPS, dbo.#nResult3.PHY,
dbo.#nResult3.PRB, dbo.#nResult3.PRO, dbo.#nResult3.RES, dbo.#nResult3.ROU,
dbo.#nResult3.SKL, dbo.#nResult3.TRG, dbo.#nResult3.UNA, dbo.#nResult3.WXE,
dbo.#nResult2.CRW, dbo.#nResult2.WRK, dbo.#nResult2.ENV, dbo.#nResult2.EQP,
dbo.#nResult2.ERR, dbo.#nResult2.MED, dbo.#nResult2.ORG, dbo.#nResult2.RDY,
dbo.#nResult2.SUP, dbo.#nResult2.VIO, dbo.#nResult1.MA,
dbo.#nResult1.MC, dbo.#nResult1.MG, dbo.#nResult1.WC
FROM dbo.#nResult3
INNER JOIN dbo.#nResult2
ON dbo.#nResult3.Aircraft_FK = dbo.#nResult2.Aircraft_FK
INNER JOIN dbo.#nResult1
ON dbo.#nResult3.Aircraft_FK = dbo.#nResult1.Aircraft_FK

```

```

SELECT  tblMishaps.Aircraft_FK,
        Count(tblMishaps.MishapID) AS TotalMishaps
        INTO #nResultTotal
FROM    dbo.tblMishaps
GROUP BY tblMishaps.Aircraft_FK

SELECT  dbo.#nResultFinal.Aircraft_FK, dbo.#nResultFinal.ADA,  dbo.#nResultFinal.ASS,
        dbo.#nResultFinal.ATT,  dbo.#nResultFinal.COM,  dbo.#nResultFinal.CON,
        dbo.#nResultFinal.CRT,  dbo.#nResultFinal.DES,      dbo.#nResultFinal.DMG,
        dbo.#nResultFinal.DOC,  dbo.#nResultFinal.DUC,  dbo.#nResultFinal.EHZ,
        dbo.#nResultFinal.EXC,  dbo.#nResultFinal.FLG,  dbo.#nResultFinal.IDQ,
        dbo.#nResultFinal.IFC,  dbo.#nResultFinal.INA,  dbo.#nResultFinal.INF,
        dbo.#nResultFinal.JDG,  dbo.#nResultFinal.KMW,  dbo.#nResultFinal.LIM,
        dbo.#nResultFinal.LGT,  dbo.#nResultFinal.MIS,  dbo.#nResultFinal.MNT,
        dbo.#nResultFinal.OBS,      dbo.#nResultFinal.OPS,  dbo.#nResultFinal.PHY,
        dbo.#nResultFinal.PRB,  dbo.#nResultFinal.PRO,  dbo.#nResultFinal.RES,
        dbo.#nResultFinal.ROU,  dbo.#nResultFinal.SKL,      dbo.#nResultFinal.TRG,
        dbo.#nResultFinal.UNA,  dbo.#nResultFinal.WXE,  dbo.#nResultFinal.CRW,
        dbo.#nResultFinal.WRK,  dbo.#nResultFinal.ENV,  dbo.#nResultFinal.EQP,
        dbo.#nResultFinal.ERR,      dbo.#nResultFinal.MED,  dbo.#nResultFinal.ORG,
        dbo.#nResultFinal.RDY,  dbo.#nResultFinal.SUP,  dbo.#nResultFinal.VIO,
        dbo.#nResultFinal.MA,  dbo.#nResultFinal.MC,  dbo.#nResultFinal.MG,
        dbo.#nResultFinal.WC,  dbo.#nResultTotal.TotalMishaps

FROM    dbo.#nResultFinal
        INNER JOIN dbo.#nResultTotal
        ON  dbl.#nResultFinal.Aircraft_FK=dbo.#nResultTotal.Aircraft_FK
ORDER BYdbo.#nResultFinal.Aircraft_FK

DROP TABLE #nResultFinal
DROP TABLE #nResultTotal
DROP TABLE #nResult3
DROP TABLE #nResult2
DROP TABLE #nResult1

return

```

spReport By Class

Alter Procedure **spReport_By_Class**
As

SET **NOCOUNT** ON

```
CREATE TABLE #nResult3
( Class_FK varchar(255),
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LGT int DEFAULT 0, LIM int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  UNK int DEFAULT 0, WXE int DEFAULT 0 )

CREATE TABLE #nResult2
( Class_FK varchar(255),
  CRW int DEFAULT 0, ENV int DEFAULT 0, EQP int DEFAULT 0,
  ERR int DEFAULT 0, MED int DEFAULT 0, ORG int DEFAULT 0,
  RDY int DEFAULT 0, SUP int DEFAULT 0, UNK int DEFAULT 0,
  VIO int DEFAULT 0, WRK int DEFAULT 0 )

CREATE TABLE #nResult1
( Class_FK varchar(255),
  MA int DEFAULT 0, MC int DEFAULT 0, MG int DEFAULT 0,
  UN int DEFAULT 0, WC int DEFAULT 0 )

CREATE TABLE #nResultFinal
( Class_FK varchar(255),
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LIM int DEFAULT 0, LGT int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  WXE int DEFAULT 0, CRW int DEFAULT 0, WRK int DEFAULT 0,
  ENV int DEFAULT 0, EQP int DEFAULT 0, ERR int DEFAULT 0,
  MED int DEFAULT 0, ORG int DEFAULT 0, RDY int DEFAULT 0,
  SUP int DEFAULT 0, VIO int DEFAULT 0, MA int DEFAULT 0,
  MC int DEFAULT 0, MG int DEFAULT 0, WC int DEFAULT 0 )
```

```
-----FOR THIRD LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID,
[3rdLevelCode],
Class_FK
```

```

INTO #nTemp3
FROM [vwReport_By_Class_3]

UPDATE #nTemp3
SET Class_FK = 'None'
WHERE Class_FK is null
--Now run the crosstab
INSERT #nResult3
EXEC dbo.rac @grpcol='Class_FK', @pvtcol='[3rdLevelCode]', @transform='count(*)',
@from='#nTemp3', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

-----FOR SECOND LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID, [2ndLevelCode], Class_FK INTO #nTemp2
FROM [vwReport_By_Class_2]

UPDATE #nTemp2
SET Class_FK = 'None'
WHERE Class_FK is null
--Now run the crosstab
INSERT #nResult2
EXEC dbo.rac @grpcol='Class_FK', @pvtcol='[2ndLevelCode]', @transform='count(*)',
@from='#nTemp2', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

-----FOR FIRST LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID, [1stLevelCode], Class_FK INTO #nTemp1
FROM [vwReport_By_Class_1]

UPDATE #nTemp1
SET Class_FK = 'None'
WHERE Class_FK is null
--Now run the crosstab
INSERT #nResult1
EXEC dbo.rac @grpcol='Class_FK', @pvtcol='[1stLevelCode]', @transform='count(*)',
@from='#nTemp1', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

-----

INSERT #nResultFinal
SELECT dbo.#nResult3.Class_FK, dbo.#nResult3.ADA, dbo.#nResult3.ASS,
dbo.#nResult3.ATT, dbo.#nResult3.COM, dbo.#nResult3.CON, dbo.#nResult3.CRT,
dbo.#nResult3.DES, dbo.#nResult3.DMG, dbo.#nResult3.DOC, dbo.#nResult3.DUC,
dbo.#nResult3.EHZ, dbo.#nResult3.EXC, dbo.#nResult3.FLG, dbo.#nResult3.IDQ,
dbo.#nResult3.IFC, dbo.#nResult3.INA, dbo.#nResult3.INF, dbo.#nResult3.JDG,
dbo.#nResult3.KNW, dbo.#nResult3.LIM, dbo.#nResult3.LGT, dbo.#nResult3.MIS,
dbo.#nResult3.MNT, dbo.#nResult3.OBS, dbo.#nResult3.OPS, dbo.#nResult3.PHY,
dbo.#nResult3.PRB, dbo.#nResult3.PRO, dbo.#nResult3.RES, dbo.#nResult3.ROU,
dbo.#nResult3.SKL, dbo.#nResult3.TRG, dbo.#nResult3.UNA, dbo.#nResult3.WXE,
dbo.#nResult2.CRW, dbo.#nResult2.WRK, dbo.#nResult2.ENV, dbo.#nResult2.EQP,
dbo.#nResult2.ERR, dbo.#nResult2.MED, dbo.#nResult2.ORG, dbo.#nResult2.RDY,
dbo.#nResult2.SUP, dbo.#nResult2.VIO, dbo.#nResult1.MA, dbo.#nResult1.MC,
dbo.#nResult1.MG, dbo.#nResult1.WC
FROM dbo.#nResult3
INNER JOIN dbo.#nResult2
ON dbo.#nResult3.Class_FK = dbo.#nResult2.Class_FK
INNER JOIN dbo.#nResult1
ON dbo.#nResult3.Class_FK = dbo.#nResult1.Class_FK

```

```

SELECT  tblMishaps.Class_FK,
Count(tblMishaps.MishapID) AS TotalMishaps
INTO #nResultTotal
FROM dbo.tblMishaps
GROUP BY tblMishaps.Class_FK

```

```

SELECT dbo.#nResultFinal.Class_FK, dbo.#nResultFinal.ADA,  dbo.#nResultFinal.ASS,
dbo.#nResultFinal.ATT,  dbo.#nResultFinal.COM,  dbo.#nResultFinal.CON,
dbo.#nResultFinal.CRT,  dbo.#nResultFinal.DES,  dbo.#nResultFinal.DMG,
dbo.#nResultFinal.DOC,  dbo.#nResultFinal.DUC,  dbo.#nResultFinal.EHZ,
dbo.#nResultFinal.EXC,  dbo.#nResultFinal.FLG,  dbo.#nResultFinal.IDQ,
dbo.#nResultFinal.IFC,  dbo.#nResultFinal.INA,  dbo.#nResultFinal.INF,
dbo.#nResultFinal.JDG,  dbo.#nResultFinal.KNW,  dbo.#nResultFinal.LIM,
dbo.#nResultFinal.LGT,  dbo.#nResultFinal.MIS,  dbo.#nResultFinal.MNT,
dbo.#nResultFinal.OBS,  dbo.#nResultFinal.OPS,  dbo.#nResultFinal.PHY,
dbo.#nResultFinal.PRB,  dbo.#nResultFinal.PRO,  dbo.#nResultFinal.RES,
dbo.#nResultFinal.ROU,  dbo.#nResultFinal.SKL,  dbo.#nResultFinal.TRG,
dbo.#nResultFinal.UNA,  dbo.#nResultFinal.WXE,  dbo.#nResultFinal.CRW,
dbo.#nResultFinal.WRK,  dbo.#nResultFinal.ENV,  dbo.#nResultFinal.EQP,
dbo.#nResultFinal.ERR,  dbo.#nResultFinal.MED,  dbo.#nResultFinal.ORG,
dbo.#nResultFinal.RDY,  dbo.#nResultFinal.SUP,  dbo.#nResultFinal.VIO,
dbo.#nResultFinal.MA,  dbo.#nResultFinal.MC,  dbo.#nResultFinal.MG,
dbo.#nResultFinal.WC,  dbo.#nResultTotal.TotalMishaps

```

```

FROM  dbo.#nResultFinal
INNER JOIN  dbo.#nResultTotal
ON  dbl.#nResultFinal.Class_FK=dbo.#nResultTotal.Class_FK

```

```

ORDER BY  dbo.#nResultFinal.Class_FK

```

```

DROP TABLE #nResultFinal
DROP TABLE #nResultTotal
DROP TABLE #nResult3
DROP TABLE #nResult2
DROP TABLE #nResult1

```

```

return

```


spReport By FiscalYear

Alter Procedure spReport_By_FiscalYear

As

SET NOCOUNT ON

CREATE TABLE #nResult3

```
( Year int,
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LGT int DEFAULT 0, LIM int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  UNK int DEFAULT 0, WXE int DEFAULT 0 )
```

CREATE TABLE #nResult2

```
( Year int,
  CRW int DEFAULT 0, ENV int DEFAULT 0, EQP int DEFAULT 0,
  ERR int DEFAULT 0, MED int DEFAULT 0, ORG int DEFAULT 0,
  RDY int DEFAULT 0, SUP int DEFAULT 0, UNK int DEFAULT 0,
  VIO int DEFAULT 0, WRK int DEFAULT 0 )
```

CREATE TABLE #nResult1

```
( Year int,
  MA int DEFAULT 0, MC int DEFAULT 0, MG int DEFAULT 0,
  UN int DEFAULT 0, WC int DEFAULT 0 )
```

CREATE TABLE #nResultFinal

```
( Year int,
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LIM int DEFAULT 0, LGT int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  WXE int DEFAULT 0, CRW int DEFAULT 0, WRK int DEFAULT 0,
  ENV int DEFAULT 0, EQP int DEFAULT 0, ERR int DEFAULT 0,
  MED int DEFAULT 0, ORG int DEFAULT 0, RDY int DEFAULT 0,
  SUP int DEFAULT 0, VIO int DEFAULT 0, MA int DEFAULT 0,
  MC int DEFAULT 0, MG int DEFAULT 0, WC int DEFAULT 0 )
```

```
-----FOR THIRD LEVEL FACTORS
--Build a temp table and update the null values to 'None"
SELECT MishapID, [3rdLevelCode], Year INTO #nTemp3
FROM [vwReport_By_FiscalYear_3]
```

```

UPDATE #nTemp3
SET Year = '0'
WHERE Year is null
--Now run the crosstab
INSERT #nResult3
EXEC dbo.rac @grpcol='Year', @pvtcol='[3rdLevelCode]', @transform='count(*)',
@from='#nTemp3', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

----- FOR SECOND LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID, [2ndLevelCode], Year INTO #nTemp2
FROM [vwReport_By_FiscalYear_2]

UPDATE #nTemp2
SET Year = '0'
WHERE Year is null
--Now run the crosstab
INSERT #nResult2
EXEC dbo.rac @grpcol='Year', @pvtcol='[2ndLevelCode]', @transform='count(*)', @from
='#nTemp2', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

-----FOR FIRST LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID,
[1stLevelCode],
Year
INTO #nTemp1
FROM [vwReport_By_FiscalYear_1]

UPDATE #nTemp1
SET Year = '0'
WHERE Year is null
--Now run the crosstab
INSERT #nResult1
EXEC dbo.rac @grpcol='Year', @pvtcol='[1stLevelCode]', @transform='count(*)', @from
='#nTemp1', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

-----

INSERT #nResultFinal
SELECT dbo.#nResult3.Year, dbo.#nResult3.ADA, dbo.#nResult3.ASS, dbo.#nResult3.ATT,
dbo.#nResult3.COM, dbo.#nResult3.CON, dbo.#nResult3.CRT, dbo.#nResult3.DES,
dbo.#nResult3.DMG, dbo.#nResult3.DOC, dbo.#nResult3.DUC, dbo.#nResult3.EHZ,
dbo.#nResult3.EXC, dbo.#nResult3.FLG, dbo.#nResult3.IDQ, dbo.#nResult3.IFC,
dbo.#nResult3.INA, dbo.#nResult3.INF, dbo.#nResult3.JDG, dbo.#nResult3.KNW,
dbo.#nResult3.LIM, dbo.#nResult3.LGT, dbo.#nResult3.MIS, dbo.#nResult3.MNT,
dbo.#nResult3.OBS, dbo.#nResult3.OPS, dbo.#nResult3.PHY, dbo.#nResult3.PRB,
dbo.#nResult3.PRO, dbo.#nResult3.RES, dbo.#nResult3.ROU, dbo.#nResult3.SKL,
dbo.#nResult3.TRG, dbo.#nResult3.UNA, dbo.#nResult3.WXE, dbo.#nResult2.CRW,
dbo.#nResult2.WRK, dbo.#nResult2.ENV, dbo.#nResult2.EQP, dbo.#nResult2.ERR,
dbo.#nResult2.MED, dbo.#nResult2.ORG, dbo.#nResult2.RDY, dbo.#nResult2.SUP,
dbo.#nResult2.VIO, dbo.#nResult1.MA, dbo.#nResult1.MC, dbo.#nResult1.MG,
dbo.#nResult1.WC
FROM dbo.#nResult3
INNER JOIN dbo.#nResult2
ON dbo.#nResult3.Year = dbo.#nResult2.Year
INNER JOIN dbo.#nResult1
ON dbo.#nResult3.Year = dbo.#nResult1.Year

```

```

SELECT  #nTemp3.Year,
Count(Distinct #nTemp3.MishapID) AS TotalMishaps
INTO #nResultTotal
FROM #nTemp3
GROUP BY #nTemp3.Year

```

```

SELECT dbo.#nResultFinal.Year, dbo.#nResultFinal.ADA, dbo.#nResultFinal.ASS,
dbo.#nResultFinal.ATT, dbo.#nResultFinal.COM, dbo.#nResultFinal.CON,
dbo.#nResultFinal.CRT, dbo.#nResultFinal.DES, dbo.#nResultFinal.DMG,
dbo.#nResultFinal.DOC, dbo.#nResultFinal.DUC, dbo.#nResultFinal.EHZ,
dbo.#nResultFinal.EXC, dbo.#nResultFinal.FLG, dbo.#nResultFinal.IDQ,
dbo.#nResultFinal.IFC, dbo.#nResultFinal.INA, dbo.#nResultFinal.INF,
dbo.#nResultFinal.JDG, dbo.#nResultFinal.KNW, dbo.#nResultFinal.LIM,
dbo.#nResultFinal.LGT, dbo.#nResultFinal.MIS, dbo.#nResultFinal.MNT,
dbo.#nResultFinal.OBS, dbo.#nResultFinal.OPS, dbo.#nResultFinal.PHY,
dbo.#nResultFinal.PRB, dbo.#nResultFinal.PRO, dbo.#nResultFinal.RES,
dbo.#nResultFinal.ROU, dbo.#nResultFinal.SKL, dbo.#nResultFinal.TRG,
dbo.#nResultFinal.UNA, dbo.#nResultFinal.WXE, dbo.#nResultFinal.CRW,
dbo.#nResultFinal.WRK, dbo.#nResultFinal.ENV, dbo.#nResultFinal.EQP,
dbo.#nResultFinal.ERR, dbo.#nResultFinal.MED, dbo.#nResultFinal.ORG,
dbo.#nResultFinal.RDY, dbo.#nResultFinal.SUP, dbo.#nResultFinal.VIO,
dbo.#nResultFinal.MA, dbo.#nResultFinal.MC, dbo.#nResultFinal.MG,
dbo.#nResultFinal.WC, dbo.#nResultTotal.TotalMishaps
FROM dbo.#nResultFinal
INNER JOIN dbo.#nResultTotal
ON dbl.#nResultFinal.Year = dbo.#nResultTotal.Year
ORDER BY dbo.#nResultFinal.Year

```

```

DROP TABLE #nResultFinal
DROP TABLE #nResultTotal
DROP TABLE #nResult3
DROP TABLE #nResult2
DROP TABLE #nResult1

```

return

spReport By Location

Alter Procedure spReport_By_Location

As

SET NOCOUNT ON

CREATE TABLE #nResult3

```
( LocationID_FK varchar(255),  
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,  
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,  
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,  
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,  
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,  
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,  
  KNW int DEFAULT 0, LGT int DEFAULT 0, LIM int DEFAULT 0,  
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,  
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,  
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,  
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,  
  UNK int DEFAULT 0, WXE int DEFAULT 0 )
```

CREATE TABLE #nResult2

```
( LocationID_FK varchar(255),  
  CRW int DEFAULT 0, ENV int DEFAULT 0, EQP int DEFAULT 0,  
  ERR int DEFAULT 0, MED int DEFAULT 0, ORG int DEFAULT 0,  
  RDY int DEFAULT 0, SUP int DEFAULT 0, UNK int DEFAULT 0,  
  VIO int DEFAULT 0, WRK int DEFAULT 0 )
```

CREATE TABLE #nResult1

```
( LocationID_FK varchar(255),  
  MA int DEFAULT 0, MC int DEFAULT 0, MG int DEFAULT 0,  
  UN int DEFAULT 0, WC int DEFAULT 0 )
```

CREATE TABLE #nResultFinal

```
( LocationID_FK varchar(255),  
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,  
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,  
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,  
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,  
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,  
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,  
  KNW int DEFAULT 0, LIM int DEFAULT 0, LGT int DEFAULT 0,  
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,  
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,  
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,  
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,  
  WXE int DEFAULT 0, CRW int DEFAULT 0, WRK int DEFAULT 0,  
  ENV int DEFAULT 0, EQP int DEFAULT 0, ERR int DEFAULT 0,  
  MED int DEFAULT 0, ORG int DEFAULT 0, RDY int DEFAULT 0,  
  SUP int DEFAULT 0, VIO int DEFAULT 0, MA int DEFAULT 0,  
  MC int DEFAULT 0, MG int DEFAULT 0, WC int DEFAULT 0 )
```

-----FOR THIRD LEVEL FACTORS
--Build a temp table and update the null values to 'None'
e the null values to 'None'

```
SELECT MishapID, [3rdLevelCode], LocationID_FK  
INTO #nTemp3
```

```

FROM [vwReport_By_Location_3]

UPDATE #nTemp3
SET LocationID_FK = 'None'
WHERE LocationID_FK is null
--Now run the crosstab
INSERT #nResult3
EXEC dbo.rac @grpcol= 'LocationID_FK', @pvtcol='[3rdLevelCode]',
@transform='count(*)', @from = '#nTemp3', @where='', @printagg='n', @grand_totals='n',
@row_totals='n', @emptycell='0'

----- FOR SECOND LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID, [2ndLevelCode], LocationID_FK INTO #nTemp2
FROM [vwReport_By_Location_2]

UPDATE #nTemp2
SET LocationID_FK = 'None'
WHERE LocationID_FK is null
--Now run the crosstab
INSERT #nResult2
EXEC dbo.rac @grpcol= 'LocationID_FK', @pvtcol='[2ndLevelCode]',
@transform='count(*)', @from = '#nTemp2', @where='', @printagg='n', @grand_totals='n',
@row_totals='n', @emptycell='0'

-----FOR FIRST LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID,
[1stLevelCode],
LocationID_FK
INTO #nTemp1
FROM [vwReport_By_Location_1]

UPDATE #nTemp1
SET LocationID_FK = 'None'
WHERE LocationID_FK is null
--Now run the crosstab
INSERT #nResult1
EXEC dbo.rac @grpcol='LocationID_FK', @pvtcol='[1stLevelCode]', @transform='count(*)',
@from = '#nTemp1', @where='', @printagg='n', @grand_totals='n', @row_totals='n',
@emptycell='0'
-----

INSERT #nResultFinal
SELECT dbo.#nResult3.LocationID_FK, dbo.#nResult3.ADA, dbo.#nResult3.ASS,
dbo.#nResult3.ATT, dbo.#nResult3.COM, dbo.#nResult3.CON, dbo.#nResult3.CRT,
dbo.#nResult3.DES, dbo.#nResult3.DMG, dbo.#nResult3.DOC, dbo.#nResult3.DUC,
dbo.#nResult3.EHZ, dbo.#nResult3.EXC, dbo.#nResult3.FLG, dbo.#nResult3.IDQ,
dbo.#nResult3.IFC, dbo.#nResult3.INA, dbo.#nResult3.INF, dbo.#nResult3.JDG,
dbo.#nResult3.KIW, dbo.#nResult3.LIM, dbo.#nResult3.LGT, dbo.#nResult3.MIS,
dbo.#nResult3.MNT, dbo.#nResult3.OBS, dbo.#nResult3.OPS, dbo.#nResult3.PHY,
dbo.#nResult3.PRB, dbo.#nResult3.PRO, dbo.#nResult3.RES, dbo.#nResult3.ROU,
dbo.#nResult3.SKL, dbo.#nResult3.TRG, dbo.#nResult3.UNA, dbo.#nResult3.WXE,
dbo.#nResult2.CRW, dbo.#nResult2.WRK, dbo.#nResult2.ENV, dbo.#nResult2.EQP,
dbo.#nResult2.ERR, dbo.#nResult2.MED, dbo.#nResult2.ORG, dbo.#nResult2.RDY,
dbo.#nResult2.SUP, dbo.#nResult2.VIO, dbo.#nResult1.MA, dbo.#nResult1.MC,
dbo.#nResult1.MG, dbo.#nResult1.WC
FROM dbo.#nResult3
INNER JOIN dbo.#nResult2
ON dbo.#nResult3.LocationID_FK = dbo.#nResult2.LocationID_FK

```

```

INNER JOIN dbo.#nResult1
ON dbo.#nResult3.LocationID_FK = dbo.#nResult1.LocationID_FK

SELECT  tblMishaps.LocationID_FK,
Count(tblMishaps.MishapID) AS TotalMishaps
INTO #nResultTotal
FROM  dbo.tblMishaps
GROUP BY tblMishaps.LocationID_FK

SELECT  dbo.#nResultFinal.LocationID_FK, dbo.#nResultFinal.ADA, dbo.#nResultFinal.ASS,
dbo.#nResultFinal.ATT, dbo.#nResultFinal.COM, dbo.#nResultFinal.CON,
dbo.#nResultFinal.CRT, dbo.#nResultFinal.DES, dbo.#nResultFinal.DMG,
dbo.#nResultFinal.DOC, dbo.#nResultFinal.DUC, dbo.#nResultFinal.EHZ,
dbo.#nResultFinal.EXC, dbo.#nResultFinal.FLG, dbo.#nResultFinal.IDQ,
dbo.#nResultFinal.IFC, dbo.#nResultFinal.INA, dbo.#nResultFinal.INF,
dbo.#nResultFinal.JDG, dbo.#nResultFinal.KNW, dbo.#nResultFinal.LIM,
dbo.#nResultFinal.LGT, dbo.#nResultFinal.MIS, dbo.#nResultFinal.MNT,
dbo.#nResultFinal.OBS, dbo.#nResultFinal.OPS, dbo.#nResultFinal.PHY,
dbo.#nResultFinal.PRB, dbo.#nResultFinal.PRO, dbo.#nResultFinal.RES,
dbo.#nResultFinal.ROU, dbo.#nResultFinal.SKL, dbo.#nResultFinal.TRG,
dbo.#nResultFinal.UNA, dbo.#nResultFinal.WXE, dbo.#nResultFinal.CRW,
dbo.#nResultFinal.WRK, dbo.#nResultFinal.ENV, dbo.#nResultFinal.EQP,
dbo.#nResultFinal.ERR, dbo.#nResultFinal.MED, dbo.#nResultFinal.ORG,
dbo.#nResultFinal.RDY, dbo.#nResultFinal.SUP, dbo.#nResultFinal.VIO,
dbo.#nResultFinal.MA, dbo.#nResultFinal.MC, dbo.#nResultFinal.MG,
dbo.#nResultFinal.WC, dbo.#nResultTotal.TotalMishaps,
dbo.tblMishapLocation.MishapLocation
FROM  dbo.#nResultFinal
INNER JOIN dbo.#nResultTotal
ON dbo.#nResultFinal.LocationID_FK = dbo.#nResultTotal.LocationID_FK
INNER JOIN dbo.tblMishapLocation
ON dbo.#nResultFinal.LocationID_FK = dbo.tblMishapLocation.MishapLocationID
ORDER BY dbo.#nResultFinal.LocationID_FK

DROP TABLE #nResultFinal
DROP TABLE #nResultTotal
DROP TABLE #nResult3
DROP TABLE #nResult2
DROP TABLE #nResult1

return

```

spReport By Organization

Alter Procedure spReport_By_Organization

As

SET NOCOUNT ON

```
CREATE TABLE #nResult3
( OrgID_FK varchar(255),
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LGT int DEFAULT 0, LIM int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  UNK int DEFAULT 0, WXE int DEFAULT 0 )

CREATE TABLE #nResult2
( OrgID_FK varchar(255),
  CRW int DEFAULT 0, ENV int DEFAULT 0, EQP int DEFAULT 0,
  ERR int DEFAULT 0, MED int DEFAULT 0, ORG int DEFAULT 0,
  RDY int DEFAULT 0, SUP int DEFAULT 0, UNK int DEFAULT 0,
  VIO int DEFAULT 0, WRK int DEFAULT 0 )

CREATE TABLE #nResult1
( OrgID_FK varchar(255),
  MA int DEFAULT 0, MC int DEFAULT 0, MG int DEFAULT 0,
  UN int DEFAULT 0, WC int DEFAULT 0 )

CREATE TABLE #nResultFinal
( OrgID_FK varchar(255),
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LIM int DEFAULT 0, LGT int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  WXE int DEFAULT 0, CRW int DEFAULT 0, WRK int DEFAULT 0,
  ENV int DEFAULT 0, EQP int DEFAULT 0, ERR int DEFAULT 0,
  MED int DEFAULT 0, ORG int DEFAULT 0, RDY int DEFAULT 0,
  SUP int DEFAULT 0, VIO int DEFAULT 0, MA int DEFAULT 0,
  MC int DEFAULT 0, MG int DEFAULT 0, WC int DEFAULT 0 )

-----FOR THIRD LEVEL FACTORS
--Build a temp table and update the null values to 'None"
SELECT MishapID, [3rdLevelCode], OrgID_FK
INTO #nTemp3
FROM [vwReport_By_Organization_3]
```

```

UPDATE #nTemp3
SET OrgID_FK = 'None'
WHERE OrgID_FK is null
--Now run the crosstab
INSERT #nResult3
EXEC dbo.rac @grpcol= 'OrgID_FK', @pvtcol='[3rdLevelCode]', @transform='count(*)',
@from = '#nTemp3', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

```

----- FOR SECOND LEVEL FACTORS

--Build a temp table and update the null values to 'None'

```

SELECT MishapID, [2ndLevelCode], OrgID_FK INTO #nTemp2
FROM [vwReport_By_Organization_2]

```

```

UPDATE #nTemp2
SET OrgID_FK = 'None'
WHERE OrgID_FK is null
--Now run the crosstab
INSERT #nResult2
EXEC dbo.rac @grpcol= 'OrgID_FK', @pvtcol='[2ndLevelCode]', @transform='count(*)',
@from = '#nTemp2', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

```

-----FOR FIRST LEVEL FACTORS

--Build a temp table and update the null values to 'None'

```

SELECT MishapID,
[1stLevelCode],
OrgID_FK
INTO #nTemp1
FROM [vwReport_By_Organization_1]

```

```

UPDATE#nTemp1
SET OrgID_FK = 'None'
WHERE OrgID_FK is null
--Now run the crosstab
INSERT #nResult1
EXEC dbo.rac @grpcol='OrgID_FK', @pvtcol='[1stLevelCode]', @transform='count(*)',
@from = '#nTemp1', @where='', @printagg='n',@grand_totals='n', @row_totals='n',
@emptycell='0'

```

```

INSERT #nResultFinal
SELECT dbo.#nResult3.OrgID_FK, dbo.#nResult3.ADA, dbo.#nResult3.ASS,
dbo.#nResult3.ATT, dbo.#nResult3.COM, dbo.#nResult3.CON, dbo.#nResult3.CRT,
dbo.#nResult3.DES, dbo.#nResult3.DMG, dbo.#nResult3.DOC, dbo.#nResult3.DUC,
dbo.#nResult3.EHZ, dbo.#nResult3.EXC, dbo.#nResult3.FLG, dbo.#nResult3.IDQ,
dbo.#nResult3.IFC, dbo.#nResult3.INA, dbo.#nResult3.INF, dbo.#nResult3.JDG,
dbo.#nResult3.KNW, dbo.#nResult3.LIM, dbo.#nResult3.LGT, dbo.#nResult3.MIS,
dbo.#nResult3.MNT, dbo.#nResult3.OBS, dbo.#nResult3.OPS, dbo.#nResult3.PHY,
dbo.#nResult3.PRB, dbo.#nResult3.PRO, dbo.#nResult3.RES, dbo.#nResult3.ROU,
dbo.#nResult3.SKL, dbo.#nResult3.TRG, dbo.#nResult3.UNA, dbo.#nResult3.WXE,
dbo.#nResult2.CRW, dbo.#nResult2.WRK, dbo.#nResult2.ENV, dbo.#nResult2.EQP,
dbo.#nResult2.ERR, dbo.#nResult2.MED, dbo.#nResult2.ORG, dbo.#nResult2.RDY,
dbo.#nResult2.SUP, dbo.#nResult2.VIO, dbo.#nResult1.MA, dbo.#nResult1.MC,
dbo.#nResult1.MG, dbo.#nResult1.WC
FROM dbo.#nResult3
INNER JOIN dbo.#nResult2
ON dbo.#nResult3.OrgID_FK = dbo.#nResult2.OrgID_FK
INNER JOIN dbo.#nResult1

```



```
ON dbo.#nResult3.OrgID_FK = dbo.#nResult1.OrgID_FK
```

```
SELECT tblMishaps.OrgID_FK,  
Count(tblMishaps.MishapID) AS TotalMishaps  
INTO #nResultTotal  
FROM dbo.tblMishaps  
GROUP BY tblMishaps.OrgID_FK
```

```
SELECT dbo.#nResultFinal.OrgID_FK, dbo.#nResultFinal.ADA, dbo.#nResultFinal.ASS,  
dbo.#nResultFinal.ATT, dbo.#nResultFinal.COM, dbo.#nResultFinal.CON,  
dbo.#nResultFinal.CRT, dbo.#nResultFinal.DES, dbo.#nResultFinal.DMG,  
dbo.#nResultFinal.DOC, dbo.#nResultFinal.DUC, dbo.#nResultFinal.EHZ,  
dbo.#nResultFinal.EXC, dbo.#nResultFinal.FLG, dbo.#nResultFinal.IDQ,  
dbo.#nResultFinal.IFC, dbo.#nResultFinal.INA, dbo.#nResultFinal.INF,  
dbo.#nResultFinal.JDG, dbo.#nResultFinal.KNW, dbo.#nResultFinal.LIM,  
dbo.#nResultFinal.LGT, dbo.#nResultFinal.MIS, dbo.#nResultFinal.MNT,  
dbo.#nResultFinal.OBS, dbo.#nResultFinal.OPS, dbo.#nResultFinal.PHY,  
dbo.#nResultFinal.PRB, dbo.#nResultFinal.PRO, dbo.#nResultFinal.RES,  
dbo.#nResultFinal.ROU, dbo.#nResultFinal.SKL, dbo.#nResultFinal.TRG,  
dbo.#nResultFinal.UNA, dbo.#nResultFinal.WXE, dbo.#nResultFinal.CRW,  
dbo.#nResultFinal.WRK, dbo.#nResultFinal.ENV, dbo.#nResultFinal.EQP,  
dbo.#nResultFinal.ERR, dbo.#nResultFinal.MED, dbo.#nResultFinal.ORG,  
dbo.#nResultFinal.RDY, dbo.#nResultFinal.SUP, dbo.#nResultFinal.VIO,  
dbo.#nResultFinal.MA, dbo.#nResultFinal.MC, dbo.#nResultFinal.MG,  
dbo.#nResultFinal.WC, dbo.#nResultTotal.TotalMishaps,  
dbo.tblMishapOrganization.MishapOrganization  
FROM dbo.#nResultFinal  
INNER JOIN dbo.#nResultTotal  
ON dbo.#nResultFinal.OrgID_FK = dbo.#nResultTotal.OrgID_FK  
INNER JOIN dbo.tblMishapOrganization  
ON dbo.#nResultFinal.OrgID_FK = dbo.tblMishapOrganization.MishapOrganizationID  
ORDER BY dbo.#nResultFinal.OrgID_FK
```

```
DROP TABLE #nResultFinal  
DROP TABLE #nResultTotal  
DROP TABLE #nResult3  
DROP TABLE #nResult2  
DROP TABLE #nResult1
```

```
return
```

spReport By Type

Alter Procedure **spReport_By_Type**

As

SET **NOCOUNT** ON

```
CREATE TABLE #nResult3
( Type_FK varchar(255),
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LGT int DEFAULT 0, LIM int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  UNK int DEFAULT 0, WXE int DEFAULT 0 )

CREATE TABLE #nResult2
( Type_FK varchar(255),
  CRW int DEFAULT 0, ENV int DEFAULT 0, EQP int DEFAULT 0,
  ERR int DEFAULT 0, MED int DEFAULT 0, ORG int DEFAULT 0,
  RDY int DEFAULT 0, SUP int DEFAULT 0, UNK int DEFAULT 0,
  VIO int DEFAULT 0, WRK int DEFAULT 0 )

CREATE TABLE #nResult1
( Type_FK varchar(255),
  MA int DEFAULT 0, MC int DEFAULT 0, MG int DEFAULT 0,
  UN int DEFAULT 0, WC int DEFAULT 0 )

CREATE TABLE #nResultFinal
( Type_FK varchar(255),
  ADA int DEFAULT 0, ASS int DEFAULT 0, ATT int DEFAULT 0,
  COM int DEFAULT 0, CON int DEFAULT 0, CRT int DEFAULT 0,
  DES int DEFAULT 0, DMG int DEFAULT 0, DOC int DEFAULT 0,
  DUC int DEFAULT 0, EHZ int DEFAULT 0, EXC int DEFAULT 0,
  FLG int DEFAULT 0, IDQ int DEFAULT 0, IFC int DEFAULT 0,
  INA int DEFAULT 0, INF int DEFAULT 0, JDG int DEFAULT 0,
  KNW int DEFAULT 0, LIM int DEFAULT 0, LGT int DEFAULT 0,
  MIS int DEFAULT 0, MNT int DEFAULT 0, OBS int DEFAULT 0,
  OPS int DEFAULT 0, PHY int DEFAULT 0, PRB int DEFAULT 0,
  PRO int DEFAULT 0, RES int DEFAULT 0, ROU int DEFAULT 0,
  SKL int DEFAULT 0, TRG int DEFAULT 0, UNA int DEFAULT 0,
  WXE int DEFAULT 0, CRW int DEFAULT 0, WRK int DEFAULT 0,
  ENV int DEFAULT 0, EQP int DEFAULT 0, ERR int DEFAULT 0,
  MED int DEFAULT 0, ORG int DEFAULT 0, RDY int DEFAULT 0,
  SUP int DEFAULT 0, VIO int DEFAULT 0, MA int DEFAULT 0,
  MC int DEFAULT 0, MG int DEFAULT 0, WC int DEFAULT 0 )

-----FOR THIRD LEVEL FACTORS
--Build a temp table and update the null values to 'None"
e the null values to 'None"
SELECT MishapID, [3rdLevelCode], Type_FK
INTO #nTemp3
```

```

FROM [vwReport_By_Type_3]

UPDATE #nTemp3
SET Type_FK = 'None'
WHERE Type_FK is null
--Now run the crosstab
INSERT #nResult3
EXEC dbo.rac @grpcol= 'Type_FK', @pvtcol='[3rdLevelCode]', @transform='count(*)',
@from = '#nTemp3', @where='', @printagg='n', @grand_totals='n', @row_totals='n',
@emptycell='0'

----- FOR SECOND LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID, [2ndLevelCode], Type_FK INTO #nTemp2
FROM [vwReport_By_Type_2]

UPDATE #nTemp2
SET Type_FK = 'None'
WHERE Type_FK is null
--Now run the crosstab
INSERT #nResult2
EXEC dbo.rac @grpcol= 'Type_FK', @pvtcol='[2ndLevelCode]', @transform='count(*)',
@from = '#nTemp2', @where='', @printagg='n', @grand_totals='n', @row_totals='n',
@emptycell='0'

-----FOR FIRST LEVEL FACTORS
--Build a temp table and update the null values to 'None'
SELECT MishapID,
[1stLevelCode],
Type_FK
INTO #nTemp1
FROM [vwReport_By_Type_1]

UPDATE #nTemp1
SET Type_FK = 'None'
WHERE Type_FK is null
--Now run the crosstab
INSERT #nResult1
EXEC dbo.rac @grpcol='Type_FK', @pvtcol='[1stLevelCode]', @transform='count(*)', @from
='#nTemp1', @where='', @printagg='n', @grand_totals='n', @row_totals='n',
@emptycell='0'

-----
INSERT #nResultFinal
SELECT dbo.#nResult3.Type_FK, dbo.#nResult3.ADA, dbo.#nResult3.ASS,
dbo.#nResult3.ATT, dbo.#nResult3.COM, dbo.#nResult3.CON, dbo.#nResult3.CRT,
dbo.#nResult3.DES, dbo.#nResult3.DMG, dbo.#nResult3.DOC, dbo.#nResult3.DUC,
dbo.#nResult3.EHZ, dbo.#nResult3.EXC, dbo.#nResult3.FLG, dbo.#nResult3.IDQ,
dbo.#nResult3.IFC, dbo.#nResult3.INA, dbo.#nResult3.INF, dbo.#nResult3.JDG,
dbo.#nResult3.KNW, dbo.#nResult3.LIM, dbo.#nResult3.LGT, dbo.#nResult3.MIS,
dbo.#nResult3.MNT, dbo.#nResult3.OBS, dbo.#nResult3.OPS, dbo.#nResult3.PHY,
dbo.#nResult3.PRB, dbo.#nResult3.PRO, dbo.#nResult3.RES, dbo.#nResult3.ROU,
dbo.#nResult3.SKL, dbo.#nResult3.TRG, dbo.#nResult3.UNA, dbo.#nResult3.WXE,
dbo.#nResult2.CRW, dbo.#nResult2.WRK, dbo.#nResult2.ENV, dbo.#nResult2.EQP,
dbo.#nResult2.ERR, dbo.#nResult2.MED, dbo.#nResult2.ORG, dbo.#nResult2.RDY,
dbo.#nResult2.SUP, dbo.#nResult2.VIO, dbo.#nResult1.MA, dbo.#nResult1.MC,
dbo.#nResult1.MG, dbo.#nResult1.WC
FROM dbo.#nResult3
INNER JOIN dbo.#nResult2
ON dbo.#nResult3.Type_FK = dbo.#nResult2.Type_FK
INNER JOIN dbo.#nResult1

```

```
ON dbo.#nResult3.Type_FK = dbo.#nResult1.Type_FK
```

```
SELECT tblMishaps.Type_FK,  
Count(tblMishaps.MishapID) AS TotalMishaps  
INTO #nResultTotal  
FROM dbo.tblMishaps  
GROUP BY tblMishaps.Type_FK
```

```
SELECT dbo.#nResultFinal.Type_FK, dbo.#nResultFinal.ADA, dbo.#nResultFinal.ASS,  
dbo.#nResultFinal.ATT, dbo.#nResultFinal.COM, dbo.#nResultFinal.CON,  
dbo.#nResultFinal.CRT, dbo.#nResultFinal.DES, dbo.#nResultFinal.DMG,  
dbo.#nResultFinal.DOC, dbo.#nResultFinal.DUC, dbo.#nResultFinal.EHZ,  
dbo.#nResultFinal.EXC, dbo.#nResultFinal.FLG, dbo.#nResultFinal.IDQ,  
dbo.#nResultFinal.IFC, dbo.#nResultFinal.INA, dbo.#nResultFinal.INF,  
dbo.#nResultFinal.JDG, dbo.#nResultFinal.KNW, dbo.#nResultFinal.LIM,  
dbo.#nResultFinal.LGT, dbo.#nResultFinal.MIS, dbo.#nResultFinal.MNT,  
dbo.#nResultFinal.OBS, dbo.#nResultFinal.OPS, dbo.#nResultFinal.PHY,  
dbo.#nResultFinal.PRB, dbo.#nResultFinal.PRO, dbo.#nResultFinal.RES,  
dbo.#nResultFinal.ROU, dbo.#nResultFinal.SKL, dbo.#nResultFinal.TRG,  
dbo.#nResultFinal.UNA, dbo.#nResultFinal.WXE, dbo.#nResultFinal.CRW,  
dbo.#nResultFinal.WRK, dbo.#nResultFinal.ENV, dbo.#nResultFinal.EQP,  
dbo.#nResultFinal.ERR, dbo.#nResultFinal.MED, dbo.#nResultFinal.ORG,  
dbo.#nResultFinal.RDY, dbo.#nResultFinal.SUP, dbo.#nResultFinal.VIO,  
dbo.#nResultFinal.MA, dbo.#nResultFinal.MC, dbo.#nResultFinal.MG,  
dbo.#nResultFinal.WC, dbo.#nResultTotal.TotalMishaps,  
dbo.tblMishapType.MishapTypeDefinition  
FROM dbo.#nResultFinal  
INNER JOIN dbo.#nResultTotal  
ON dbo.#nResultFinal.Type_FK = dbo.#nResultTotal.Type_FK  
INNER JOIN dbo.tblMishapType  
ON dbo.#nResultFinal.Type_FK = dbo.tblMishapType.MishapLocationID  
ORDER BY dbo.#nResultFinal.Type_FK
```

```
DROP TABLE #nResultFinal  
DROP TABLE #nResultTotal  
DROP TABLE #nResult3  
DROP TABLE #nResult2  
DROP TABLE #nResult1
```

```
return
```

RAC (Replacement for Access Crosstab) Version 1.50 beta for SQL Server 2000

Written by Steve Dassin (2001) Used with permission

Alter procedure rac /* RAC */

--Basic Parameters

```
@transform varchar(7000)=null,
@grpcol varchar(500),
@from varchar(2500),
@where varchar(4000)='',
@pvtcol varchar(500)=null,
@grpsorttype varchar(1)='a',
@grpsortsub varchar(350)='',
@pvtsorttype varchar(1)='a',
@pvtsortsub varchar(150)='',
@xtab varchar(75)='#',
@worktable1 varchar(75)='',
@worktable2 varchar(75)='',
@worktable3 varchar(75)='',
@racheck varchar(1)='n',
@return varchar(1)='y',
@tablecater varchar(1)='n',
@limit varchar(50)='',
@translabel varchar(100)='Funct',
@printagg varchar(5)='y',
@row_totals varchar(1)='y',
@grand_totals varchar(1)='y',
@emptycell varchar(20)='',
@nullcell varchar(20)='',
@grandtotalposition varchar(5)='end',
@rowtotalposition varchar(5)='begin',
@display varchar(5)='m',
@blocktype varchar(5)='seq',
@rowbreak varchar(1)='y',
@multibreak varchar(1)='f',
@rowcnters varchar(1000)='',
@printrowcnters varchar(1)='y',
@wherecters varchar(1000)='',
@space int=0,
@functionlen varchar(4)='25',
@milen varchar(3)='50',
@mxlenagg varchar(4)='50',
@getmxlenagg varchar(1)='n',
@datelen varchar(2)='8',
@style varchar(3)='1',
@exec varchar(1)='y',
@tabledef varchar(8000) =null output,
@xtabfields varchar(8000) =null output,
@user1 varchar(600)='',
@user2 varchar(600)='',
@user3 varchar(600)='',
@user4 varchar(600)='',
@user5 varchar(600)='',
@user6 varchar(600)='',
@user7 varchar(600)='',
@user8 varchar(600)='',
@user9 varchar(600)='',
@user10 varchar(600)='',
@user11 varchar(600)='',
@user12 varchar(600)='',
```

```

@user13 varchar(600)='',
@user14 varchar(600)='',
@user15 varchar(600)='',
-- Advanced Parameters
@printqry varchar(1)='n',
@forcerange varchar(100)='',
@force varchar(2400)='',
@forcetest varchar(1)='y',
@forcerows varchar(1)='n',
@grpart varchar(50)='',
@pvtpart varchar(50)='',
@cpercents varchar(100)='',
@rpercents varchar(100)='',
@rpercentstotals varchar(100)='',
@totals_only varchar(1)='n',
@rowfunctions varchar(500)='',
@displayrowfunctions varchar(1)='s',
@rowrunslabel varchar(100)='[Runs]',
@rowruns varchar(1000)='',
@diffs varchar(1000)='',
@colruns varchar(100)='',
@colretain varchar(100)='',
@all_cols varchar(75)='[All_cols]',
@separator varchar(20)=';',
@colretainpvt varchar(1)='n',
@cutpvt varchar(1)='n',
@rank varchar(50)=null,
@ranklimit varchar(4)='0',
@sortagg varchar(1)='n',
@fieldadd1 varchar(50)='',
@fieldvalue1 varchar(50)='',
@fieldadd2 varchar(50)='',
@fieldvalue2 varchar(50)='',
@fieldadd3 varchar(50)='',
@fieldvalue3 varchar(50)='',
@grpsortnum varchar(1)='n',
@pvtortnum varchar(2)='n',
@split varchar(1)='n',
@pformat varchar(1)='n',
@convert varchar(25)='',
@burst varchar(1)='n',
@burstlen int=7500,
@multicell varchar(1)='n',
@multicellrun2 varchar(1)='',
@rotate varchar(10)='n'
as
-- Table for multi field @grpcol
declare @groupstable# table (rd int identity primary key,grp varchar(150))
-- Table for multi field associated sort fields (from @grpsortsub )
declare @groupstablesort# table (rd int identity primary key,grp varchar(150))
-- Used for multi row expressions in update for rowbreak=y
declare @groupwork# table (rd int identity primary key,agrp varchar(150),bgrp
varchar(150),ard int,brd int)
--
declare @trans# table
(rd int identity,agg varchar(300),
funct as case when charindex('(',agg)=0 then 'max('+agg+')' else
case when charindex(' sa )',reverse(agg))>0 then
rtrim(ltrim(reverse(substring(reverse(agg),charindex(' sa
)',reverse(agg))+4,len(agg))))))

```

```

else rtrim(ltrim(agg)) end end,
alias as case when charindex('(',agg)=0 then agg else
case when
len(ltrim(reverse(replace(substring(reverse(agg),1,charindex(' sa '),reverse(agg))),
' sa '),'')))>0
then
ltrim(reverse(replace(substring(reverse(agg),1,charindex(' sa '),reverse(agg))),
' sa '),''))
else
rtrim(ltrim(reverse(substring(reverse(agg),charindex('('),reverse(agg)),len(agg))))
end end)
--
declare @rowfuncts# table (rd int identity,rowfuncterm varchar(100),
rowfunct as substring(rowfuncterm,1,charindex('(',rowfuncterm)-1),
rtrans as substring(rowfuncterm,charindex('(',rowfuncterm)+1,len(rowfuncterm)-
charindex('(',rowfuncterm)-1 ))
--
declare @rowsums# table (rd int identity,runfuncterm varchar(150),
runcol as rtrim(ltrim(substring(replace(runfuncterm,case when
charindex('{',runfuncterm)>0 and charindex('}',runfuncterm)>0 then
substring(runfuncterm,charindex('{',runfuncterm),
((charindex('}',runfuncterm)) - (charindex('{',runfuncterm)-1))) else ''
end,''),1,charindex('(',replace(runfuncterm,case when charindex('{',runfuncterm)>0 and
charindex('}',runfuncterm)>0 then
substring(runfuncterm,charindex('{',runfuncterm),
((charindex('}',runfuncterm)) - (charindex('{',runfuncterm)-1))) else '' end,''))-
1))),
rtrans as rtrim(ltrim(substring(replace(runfuncterm,case when
charindex('{',runfuncterm)>0 and charindex('}',runfuncterm)>0 then
substring(runfuncterm,charindex('{',runfuncterm),
((charindex('}',runfuncterm)) - (charindex('{',runfuncterm)-1))) else ''
end,''),charindex('(',replace(runfuncterm,case when charindex('{',runfuncterm)>0 and
charindex('}',runfuncterm)>0 then
substring(runfuncterm,charindex('{',runfuncterm),
((charindex('}',runfuncterm)) - (charindex('{',runfuncterm)-1))) else '' end,''))+1,
len(replace(runfuncterm,case when charindex('{',runfuncterm)>0 and
charindex('}',runfuncterm)>0 then
substring(runfuncterm,charindex('{',runfuncterm),
((charindex('}',runfuncterm)) - (charindex('{',runfuncterm)-1))) else '' end,''))-
charindex('(',replace(runfuncterm,case when charindex('{',runfuncterm)>0 and
charindex('}',runfuncterm)>0 then
substring(runfuncterm,charindex('{',runfuncterm),
((charindex('}',runfuncterm)) - (charindex('{',runfuncterm)-1))) else '' end,''))-1
))),
rowfield as rtrim(ltrim(case when charindex('{',runfuncterm)>0 and
charindex('}',runfuncterm)>0 then
substring(runfuncterm,charindex('{',runfuncterm)+1,
((charindex('}',runfuncterm)-1) - (charindex('{',runfuncterm)))) else '' end )))
--
declare @transfields# table (rd int identity,field varchar(600))
--
declare @user# table (rd int identity primary key,userid as cast('@user'+cast(rd as
varchar(2)) as varchar(7)),
usercode varchar(600))
--
declare @tabledef# table (rd int identity primary key,field varchar(7800))
--
declare @check int,@str1 varchar(8000),@rd int,@mxlengrp varchar(4),@mxlenpvt
varchar(4),

```



```

@mxlencol varchar(4),@fields varchar(8000),@col varchar(7930),@str2
nvarchar(4000),@maxlen varchar(4),
@grpsort varchar(1000),@pivot sort varchar(2500),@sortgrp varchar(10),@sortpvt
varchar(10),
@key1 int,@rowcount int,@passes int,@work int,@qrystatement1 varchar(1000),
@qrystatement2 varchar(1000),@grpcolinsert varchar(150),@pvtcolinsert varchar(150),
@pivotinlen varchar(2),@grpfldtype varchar(20),@pvtfldtype varchar(20),
@numtrans int,@wh varchar(100),@qrytrans varchar(7000),@k int,@test int,
@grpcolvalue varchar(10),@valinsert varchar(7000),@valcreate varchar(7000),
@numvalues int,@char varchar(3),@colpercents varchar(750),@type varchar(10),
@funct varchar(300),@aggfields varchar(125),@aggvalues varchar(300),
@forceparse varchar(2400),@rowfunctqrys varchar(1500),@rowfunctfield varchar(35),
@aliasck varchar(1),@runsfieldlen varchar(2),@runsfield varchar(135),@runsvalue
varchar(500),
@runs varchar(3000),@d1 int,@d2 int,@d1c varchar(25),@d2c varchar(25),@decimal
varchar(25),
@pvttable varchar(250),@grptable varchar(250),@partgrpjoin varchar(500),
@partpvtjoin varchar(500),@fieldsdef varchar(8000),@rowfunctlen varchar(3),
@rowfunctlen1 varchar(3),@rowp varchar(1),@printaggs varchar(300),
@fieldsforce varchar(8000),@fieldselect varchar(8000),@crunsvalue varchar(250),
@cruns varchar(1000),@runsfieldsingle varchar(3),@fieldspvt varchar(8000),
@fieldsemt varchar(8000),@startpivotflds varchar(100),@runs0 varchar(100),
@runsdeclare varchar(200),@updatecut varchar(300),@retain varchar(1000),
@retainvalall varchar(40),@retainfld varchar(75),
@rowpercents varchar(500),@rtotal float(2),@flag1 varchar(1),@rbreak varchar(1),
@block varchar(50),@rdstart varchar(8),@rank1 varchar(150),@where1 varchar(4000),
@multicelltable varchar(75),@char1 varchar(5),@multigrpsortnum varchar(1),
@multiptsortnum varchar(1),@returnmulticell varchar(1),@multicellforce varchar(2400),
@f1 varchar(2),@multiworktable1 varchar(75),@multiworktable2 varchar(75),
@multiworktable3 varchar(75),@worktable varchar(75),@multicellforcerange varchar(100),
@multifieldadd1 varchar(50),@multifieldvalue1 varchar(50),@multifieldadd2 varchar(50),
@multifieldvalue2 varchar(50),@multifieldadd3 varchar(50),@multifieldvalue3
varchar(50),
@icheck int,@str varchar(7000),@charx varchar(1),@runsfieldselect varchar(125),
@rowfunctfieldselect varchar(25),@printaggfieldselect varchar(125),@aliastable
varchar(2500),
@alias varchar(150),@mgrps varchar(1000),@mgrpsfieldsdef
varchar(500),@mgrpsfieldselect varchar(500),
@mgrpsupdate1 varchar(500),@mgrpsupdate2 varchar(500),@mgrpsupdate3 varchar(500),
@mgrpsupdate4 varchar(500),@mgrpsupdate5 varchar(500),@mgrpsqryselect varchar(500),
@mgrpsupdate6 varchar(500),@mgrpsupdate4breakvalues varchar(500),
@mgrpsupdate4breakfields varchar(500),@mgrpsupdateinit varchar(500),@lengthd1
varchar(3),
@rotatefields varchar(7500),@rotatefieldselect varchar(7500),@rotatefieldspvt
varchar(7500),
@rotatefieldspvtselect varchar(7500),@rotatetab varchar(75),@rotatequery
varchar(7950),
@checktable int,@rowfcnt int,@rowfmin int,@rowfmax int,@mgrpswhere varchar(500),
@limit1 int,@limit2 int,@convertfields varchar(7500),@convertfieldselect
varchar(7500),
@convertfieldspvtselect varchar(7500),@converttable varchar(75),@convertquery
varchar(7950),
@ctvalue varchar(500),
@ctfield varchar(250),
@ctfieldselect varchar(250),
/* @runsvalue */
/* @runsfield */
/* @runsfieldselect */
/* @runsfieldlen not using for cnter */
/* @runsdeclare */
/* @runs0 */
/* @runs */
@ctdeclare varchar(200),
@ct0 varchar(100),
@ct varchar(3000),

```



```

@rowcnterswork varchar(1000),@ctfields varchar(250), /* for fields definition */
@cntername varchar(150),
@difffields varchar(250),@difffieldselect varchar(250),@difffield varchar(10),@diffvalue
varchar(50),
@diffdeclare varchar(75),@diff0 varchar(35),@df varchar(3000),@diffonly varchar(500),
@testruns varchar(150),@kcheck int,@wherecntersupdate1
varchar(1000),@wherecntersupdate2 varchar(1000)
--
set fmtonly off
set nocount on
set ansi_warnings off
-- Set up LTTP (local temp table processing) by first fully specifying
-- or anchoring all tables.
create table #rac(rd int identity)
-- Check for partitioning
-- check for @grpart
if @grpart>' '
    create table #part1(rd int identity)
-- check for @pvtpart
if @pvtpart>' '
    create table #part2(rd int identity)
-- Check for rotation
--if @rotate='y'
--begin
--set @rotatetextab=@xtab
--set @xtab='#xtab'
--end
--
-- #trans definitions check for '( in @transform,if false it is transpose
-- Replace user variables,for now only @transform,@grpcol,@pvtcol,@from,@where
--
set @str=@user1+'^'+@user2+'^'+@user3+'^'+@user4+'^'+@user5+'^'+@user6+
'^'+@user7+'^'+@user8+'^'+@user9+'^'+@user10+'^'+@user11+'^'+@user12+'^'+@user13+
'^'+@user14+'^'+@user15
if len(@str)>14
goto user_table
resume4:
if len(@str)>14
select
@transform=replace(@transform,userid,usercode),
@grpcol=replace(@grpcol,userid,usercode),
@pvtcol=replace(@pvtcol,userid,usercode),
@from=replace(@from,userid,usercode),
@where=replace(@where,userid,usercode)
from @user#
order by rd desc
--
--Check for split, if yes eliminte empty strings (ie. /asp/print first / produces '')
if @split='y'
begin
if @where=''
begin
if @grpcol!='none'
set @where='datalength('+@grpcol+')>0 and response>~'
else
set @where='response>~'
end
else
set @where=@where+' and response>~'
end

```

```

--
-- For rowfunctions using table variable:@rowfunct#
-- For running sums using table variable:@rowsums#
-- For column runs using variable:@colruns
-- For concatenating/retain using variable:@colretain
-- For force using function racpovitin#
--
--                               Multicell specs
--
if @multicell='y'
begin
if @xtab!='#'
begin
-- Eliminate any further processing if @xtab table already exists (for recursive 2nd
RAC run)
exec @check=racsp;18 /* checkiftableexists */ @xtab=@xtab
if @check=-2
return(-3)
end
set @multicelltable=@xtab
set @xtab='#multicellzz1'
-- @row_totals needs to be active for correct #multicellzz1,possible logic conflict.
set @row_totals='n'
-- Can turn off @grand_totals
set @grand_totals='n'
-- Don't want @all_cols named changed for 1st rac run
set @all_cols ='[All_cols]'
set @colretainpvt='y' /* @colretain is set below from alias in #trans */
set @cutpvt='y'
set @printagg='n' /* don't need transform */
--Store @pvtsortnum for 2nd rac run where it can be applied to pvtcol ([column])
field.
--In 1st rac run cannot use @pvtsortnum because of concatenated field.@grpsortnum is
not being
--used in favor of subselect in order by that sorts 2nd run just like 1st run.
--@grpsortnum can be used in rac 1st run.It is set to 'n' in 2nd run.
set @multipvtsortnum=@pvtsortnum
set @pvtsortnum='n'
--Return can be 'y' (return multicell report only ),'1' (for #multicellzz1 only),
--'b' (both #multicellzz1 and multicell report (@xtab)) or 'n' (return neither 1st or
2nd run).
if @return='y'
begin
set @return='n'
set @returnmulticell='y'
end
else
if @return='b'
begin
set @return='y'
set @returnmulticell='y'
end
else
if @return='1'
begin
set @return='y'
set @returnmulticell='n'
end
else
-- Here return is 'n'

```

```

set @returnmulticell='n'
-- Force applies only to 2nd rac run because in 1st run
-- @pvtcol is a compound expression involving the column
-- plus concatenated string part. When using @force you have to eliminate same forced
-- values in @where which is applied to 1st rac run.
-- Wanted to check the forced values are in where too. Not only is it to complex but
where may
-- be inside a derived table too. To hard to check. This requirement is made clear in
racdoc.
set @multicellforcerange=@forcerange /* save forcerange for 2nd run */
set @forcerange=''
set @multicellforce=@force /* save force for 2nd run */
set @force='' /* shut force off for 1st run */
-- Don't want to use force in @where in 2nd run because 2nd run @where must be a
constant
-- (1=1), so setting force off for both runs.
set @forcetest='n'
-- The #temp tables formally used for holding 2nd run @xtabfields/@tabledef
-- (to be obtained from run 1) have been replaced by global cursors. Cursors are
declared
-- in run2 and opened/fetched in returning run1.
-- Shut off worktable for 1st run, only save it from 2nd run
set @multiworktable1=@worktable1
set @multiworktable2=@worktable2
set @multiworktable3=@worktable3
set @worktable1=''
set @worktable2=''
set @worktable3=''
-- Shut off fieldadd/fieldvalue for 1st run, only save it from 2nd run
set @multifieldadd1=@fieldadd1
set @multifieldvalue1=@fieldvalue1
set @multifieldadd2=@fieldadd2
set @multifieldvalue2=@fieldvalue2
set @multifieldadd3=@fieldadd3
set @multifieldvalue3=@fieldvalue3
set @fieldadd1=''
set @fieldvalue1=''
set @fieldadd2=''
set @fieldvalue2=''
set @fieldadd3=''
set @fieldvalue3=''
end
--
-- Check for splitting if yes create #split table
if @split='y'
begin
create table #split(pos int, response varchar(5000))
-- exec rsplit
execute @check=racsp;2 @row=@grpcol, @string=@pvtcol, @intable=@from, @char=@separator
if @check!=0
begin
raiserror('error in sp racsp;2 (rsplit)', 16, 1) with nowait
return(-3)
end
-- Define RAC parameters
set @from='#split'
set @transform='max(response) as split'
if @grpcol='none'
set @grpcol='recid'
set @pvtcol='pos'

```

```

if @rank is null
    set @rank=''
    set @row_totals='n'
    set @printagg='n'
end /* end of splitting */
-- Check for existence of worktables.if @worktables option is active.It can only be
-- global temp or permanent table.Note that for multicell 1st run worktable is always
set off.
--if @worktable1>' and @worktable2>'
-- No worktable3 for rotate
if (@worktable3>'') and (@rotate in ('y','nest'))
begin
raiserror('sorry,no @worktable3 is available for @rotate option',0,1)with nowait
return(-3)
end
--
if (select case when @worktable1>' then 1 else 0 end+
            case when @worktable2>' then 1 else 0 end+
            case when @worktable3>' then 1 else 0 end )>1
begin
raiserror('only 1 worktable (@worktable-1,2 or 3) can be specified',0,1)with nowait
return(-3)
end
if @worktable1>'
set @worktable=@worktable1
else
if @worktable2>'
set @worktable=@worktable2
else
set @worktable=@worktable3
--
if @worktable>'
begin
if charindex('#',@worktable)>0 and charindex('##',@worktable)=0
begin
raiserror('only a global temp table (or permanent table) can store worktable not %s',
0,1,@worktable)with nowait
return(-3)
end
-- Check if it already exists .
exec @check=racsp;18 /* checkiftableexists */ @xtab=@worktable
if @check=-2
return(-3)
end /* ends worktable>' */
--
-- All tables have now been fully created or anchored.
-- Check @display and other functions for invalid entries.
if @rotate in ('y','nest')
begin
if @blocktype!='seq'
begin
raiserror('you can only rotate with default value of @blocktype (='seq')',0,1) with
nowait
return(-3)
end
if @printagg!='y'
begin
raiserror('you can only rotate with default value of @printagg (='y')',0,1) with
nowait
return(-3)
end
end

```

```

end
if charindex('&',@grpcol)>0 and @forcerows='y'
begin
raiserror('sorry,multiple row fields are not available with the @forcerows='y''
option',0,1) with nowait
return(-3)
end
if (@rank is null) and (@sortagg='y')
begin
raiserror('sorry,@sortagg='y'' can only be used with the @rank option',0,1) with
nowait
return(-3)
end
--
if (@burst='y') and (@burstlen>7500)
begin
raiserror('the maximum value of @burstlen is 7500',0,1) with nowait
return(-3)
end
--
if @display='s'
begin
if @rowfunctions>' '
begin
raiserror('only displaying rowfunctions with @display='m'' ',0,1) with nowait
return(-3)
end
else
if @rotate in ('y','nest')
begin
raiserror('you can only rotate with @display='m'' ',0,1) with nowait
return(-3)
end
else
if @rowruns>' '
begin
raiserror('only displaying running sums with @display='m'' ',0,1) with nowait
return(-3)
end
else
if @cpercents>' '
begin
raiserror('only displaying col percents with @display='m'' ',0,1) with
nowait
return(-3)
end
else
if @rpercents>' '
begin
raiserror('only displaying all row percents with @display='m'' ',0,1) with
nowait
return(-3)
end
else
if @rpercentstotals>' '
begin
raiserror('only displaying total row percents with @display='m'' ',0,1) with
nowait
return(-3)
end
end
end

```

```

else
if @colruns>'
begin
raiserror('only displaying col runs with @display='m' ',0,1) with nowait
return(-3)
end
else
if @colretain>'
begin
raiserror('only concatenating/retaining column values with @display='m' ',0,1)
with nowait
return(-3)
end
end
-- Start of populating tables
--Transform aggregates are processed with table @trans#
set @transform=replace(@transform,'~',char(39))
-- Populate @trans# table
goto trans_table
resume1:
select @numtrans=max(rd) from @trans#
-- Check rotate and number of transforms (though rotate WILL work with 1 transform)
if @rotate in('y','nest')
begin
/* if @numtrans=1
begin
raiserror('using rotate with 1 transform is redundant (same table),use rotate with >=2
transforms',0,1) with nowait
return(-3)
end */
-- Check if aliases are specified in @transform.They are required for rotate.
if exists(select * from @trans# where funct=alias)
begin
raiserror('the rotate option requires each @transform to have an alias',0,1) with
nowait
return(-3)
end
end
-- For multicell report get funct/alias for colretain
if @multicell='y'
select @colretain=alias from @trans#
--
-- Check for transposing
--
if charindex('(',@transform)=0
begin
set @from=@from+',(select ~value~ as singlecol) as zza913'
set @pvtcol='singlecol'
set @row_totals='n'
set @grand_totals='n'
end
--
-- Check for select_query group by simulation
--
if charindex('select_query',@pvtcol)>0
begin
set @row_totals='n' /* must be off for select_query */
set @blocktype='seq' /* blocktype cannot be 'stack' */
set @display='m' /* necessary for >1 transform */
-- Check that all @transforms have an alias,no rowfunctions

```

```

if exists(select * from @trans# where funct=alias)
begin
raiserror('a select query requires each @transform to have an alias',0,1) with nowait
return(-3)
end
if @rowfunctions>' '
begin
raiserror('rowfunctions cannot be obtained with a select query,see @cutpvt option in
doc',0,1) with nowait
return(-3)
end
-- Check for number of @transforms
if @numtrans=1
begin
--Here we want to put @transform alias name in derived table and there's no rotation.
select @str1='(select ~'+ replace(replace(alias,[' ','']),']','') + '~ as
select_query) as tz1a'
from @trans# where rd=1
set @from=@from+' '+@str1
set @printagg='n' /* necessary when there's only 1 @transform */
set @rotate='n'
end
else
begin
-- Here there's multiple @transforms
set @from=@from+'(select ~dummyz1a~ as select_query) as tz1a'
set @printagg='y'
set @rotate='y'
end
end /* end of select_query */
--
set @grpcol=replace(@grpcol,'~',char(39))
set @pvtcol=replace(@pvtcol,'~',char(39))
set @from=replace(@from,'~',char(39))
-- Check if xtab is local temp table
-- And if rotate is active create #xtab and store original @xtab in @rotatextab..
--
set @checktable=len(@xtab)-len(replace(@xtab,'#',''))
-- No matter what,check if a save table exists
if @checktable!=1
begin
exec @check=racsp;18 /* checkiftableexists */ @xtab=@xtab
if @check=-2
return(-3)
end
--
-- Check for convert,If there's a saved table we want crosstab in a #temp table
-- so it can then be selected into saved @xtab table in converted form.
set @convertable=''
if @convert>' ' and @checktable!=1
begin
set @checktable=1
set @convertable=@xtab
end
--
if (@checktable=1) or (@rotate in ('y','nest')) /* using local temp table
'#xtab',exists for duration of proc only */
begin
if @multicell!='y'
begin

```

```

create table #xtab(rd int)
--
-- For rotate,if no @xtab wanted (## or permanent table) and pformat wanted we're
-- creating
-- special rotate table (#rotatez1a) to be altered and populated in sp rotate,
--
if @rotate in ('y','nest')    /* using rotatextab=#rotatez1a for all rotations where
no @xtab */
begin
if @checktable=1    /* meaning #xtab */
begin
--if charindex('select_query',@pvtcol)>0 and @pformat='n' /* this is only condition
for # in sp rotate */
--set @rotatextab='#'
--else
--begin
set @rotatextab='#rotatez1a'
create table #rotatez1a(rd int identity)
--end
end /* end @checktable=1 */
else
-- Here a saved xtab is wanted so we're saving it for sp rotate in rotatextab.
-- Or if no pformat and no saved table, rotatextab is '#' and #rotatez1a will NOT be
-- created.
set @rotatextab=@xtab    /* rotatextab will be '#' if no xtab specified and pformat is
no*/
end /* ends if @rotate in (y,nest) */
set @xtab='#xtab'    /* no matter what for rotate yes,initial @xtab is #xtab */
end /* ends multicell!=y */
else
-- Here is a multicell 1st run always using table #multicellzz1
create table #multicellzz1(rd int)
end
--
--          Check for a multi field @grpcol
--
set @mgrps=''
set @mgrpsfieldsdef=''
set @mgrpsupdate4breakvalues=' '
set @mgrpsupdateinit=''
if charindex('& ',@grpcol)>0
begin
-- Fill in @groupstable# with individual @grpcol (row) fields and
@groupstable#
-- if there are sort fields associated with any multi group fields.
goto multigrpcols
resume6:
-- Form new @grpcol (concatenating all fields) and @mgrps (concatenated expression
for query).
-- Checking each field for a '*' which indicates it is a datetime field.It will be
-- converted using *datelen and @style
-- Example of @mgrps statement
--case when grouping(ship_date)=0 then max(shipcountry) end ,
--case when grouping(ship_date)=0 then max(orderdate) end
-- Putting comma between fields for now.To get rid of comma just use '+'
-- Forming @mgrpsfieldsdef for table create
-- Forming @mgrpsfieldselect for table create
-- Forming @mgrpsupdate1 - this is assignment at end of update.Last field always
varies
-- so it is not needed.So for 2 fields,f1,f2, need only @f1=#rac.f1.

```



```

-- For 3 fields f1,f2,f3 need only @f1=#rac.f1,@f2=#rac.f2
--set @mlen='50' /* using this as default length of multi row fields(now a
parameter) */
set @mgrpsfieldselect=''
--set @mgrpsfieldsdef=''
set @mgrpsqryselect=''
set @mgrpsupdate1=''
set @mgrpsupdate2=''
set @mgrpsupdate3=''
set @mgrpsupdate4=''
set @mgrpsupdate5=''
set @mgrpsupdate6=''
set @mgrpswhere=''
-- select @k=max(rd) from @groupstable#
set @kcheck=@k /* for checking in rowruns.rowcnters */
--
-- Getting terms where table alias,if present is needed
set @grpcol=''
select
-- Here we want table if in expression.
@grpcol=@grpcol+case when @grpcol>' ' then '+' else '' end
+ case when charindex('*',grp)=0 then 'cast('+replace(grp,'(-d)','')+ ' as
varchar('+@mlen+')'
else 'convert(varchar('+@datelen+')'+ replace(replace(grp,'*', ''), '(-d)', '')
+', '+@style+')' end,
-- @mgrpsqryselect for query ie .max(shipcountry) as f1,max(a.customerid) as f2,
-- we want table alias here.
--@mgrpsqryselect=@mgrpsqryselect
--+ 'max('+ case when charindex('*',grp)=0 then replace(grp,'(-d)','')
--else 'convert(varchar('+@datelen+')'+ replace(replace(grp,'*', ''), '(-d)', '')
+', '+@style+')' end
--+) as f'+cast(rd as varchar(3)) +', '
-- New @mgrpsqryselect,casting as varchar(@mlen)
@mgrpsqryselect=@mgrpsqryselect
+ case when charindex('*',grp)=0 then 'cast(max('+ replace(grp,'(-d)','') +') as
varchar('+@mlen+')'
else 'max(convert(varchar('+@datelen+')'+ replace(replace(grp,'*', ''), '(-
d)', '') +', '+@style+')' end
+') as f'+cast(rd as varchar(3)) +', '
@mgrpswhere=@mgrpswhere+case when @mgrpswhere>' ' then '*' else '' end
+ 'datalength('+replace(grp,'(-d)','') +')'
from @groupstable#
order by rd
--
-- Now get @mgrps using @grpcol and alias if present.
-- mgrps for query. Need alias here ie..
--case when grouping(ship_date)=0 then max(shipcountry) end ,
--case when grouping(ship_date)=0 then max(orderdate) end
/* select @mgrps=@mgrps+case when @mgrps>' ' then ',' else '' end
+ 'case when grouping('+@grpcol+')=0 then max('+replace(replace(grp,'*', ''), '(-
d)', '')+') end '
+case when charindex('(-d)',grp)=0 then 'asc' else 'desc' end
from @groupstable#
order by rd */
-- Using sort field(s) if present to sort grouping individual grouping
fields
select @mgrps=@mgrps+case when @mgrps>' ' then ',' else '' end
+ 'case when grouping('+@grpcol+')=0 then max('+
case when b.grp is null or charindex('*',b.grp)>0 then
replace(replace(a.grp,'*', ''), '(-d)', '') else b.grp end +') end '

```

```

+case when charindex('(-d)',a.grp)=0 then 'asc' else 'desc' end
from @groupstable# as a left join @groupstablestorsort# as b on a.rd=b.rd
order by a.rd
--
-- Now we don't need '*' for date or table alias so we'll get rid of them with
update.
-- After update we only have clean field names.
update @groupstable#
set grp=case when charindex('.',grp)=0 then replace(replace(grp,'*',''),'(-d)','')
else substring( replace(replace(grp,'*',''),'(-d)','') ,
charindex('.',replace(replace(replace(grp,'*',''),'(-d)','')))+1,500) end
--
select
-- Using defaults.@mgrpsfieldsdef used in alter #rac and @fields def.
@mgrpsfieldsdef=@mgrpsfieldsdef+ grp+' varchar('+@mlen+')default''','',
--@mgrpsfieldsdef=@mgrpsfieldsdef+ case when @mgrpsfieldsdef>' then ',' else ''
end
-- + grp+' varchar(50)default''',
@mgrpsfieldselect=@mgrpsfieldselect+ case when @mgrpsfieldselect>' then ',' else ''
end
+ grp,
-- @mgrpsupdate1 : for update declare ie. @f1 varchar(50),@f2 varchar(50),
@mgrpsupdate1=@mgrpsupdate1+case when rd<@k then
'@f'+cast(rd as varchar(3)) + ' varchar('+@mlen+')' else '' end,
-- @mgrpsupdateinit : for update declare ie. @f1='',@f2='',
@mgrpsupdateinit=@mgrpsupdateinit + case when rd<@k then
'@f'+cast(rd as varchar(3)) + '=~,~' else '' end,
--@mgrpsupdate2 : for update ie. @f1=#rac.customerid (put with
@grp=#rac.@grpcolinsert
@mgrpsupdate2=@mgrpsupdate2+case when rd<@k then
case when @mgrpsupdate2>' then ',' else '' end
+ '@f'+ cast(rd as varchar(3))+='#rac.'+ grp else '' end,
-- @mgrpsupdate3 : for update ie. 'shipcountry,customerid' (using
@mgrpsfieldselect)
-- @mgrpsupdate4 : for update ie. #rac.shipcountry,#rac.customerid
-- Changing ',~' in case to ~,~
--@mgrpsupdate4=@mgrpsupdate4+ case when @mgrpsupdate4>' then '+,~+' else '' end
--+ '#rac.'+grp,
@mgrpsupdate4=@mgrpsupdate4+ case when @mgrpsupdate4>' then '+,~+' else '' end
+ ' char(39)+#rac.'+grp+'char(39) '
from @groupstable#
order by rd
set @mgrpsupdate3 =char(39)+@mgrpsfieldselect+char(39)
--
set @mgrpsupdate3=' case when #rac.@grpcolinsert=~@@@~ then ~'+(select grp from
@groupstable# where rd=1)+'~ else '+'
@mgrpsupdate3+' end '
--
-- Get expressions for full or partial breaks if @rowbreak=y
--
if @rowbreak='y' /* different update5 (fields) for different rowbreak (y/n) */
begin
--
if @multibreak='f' /* this is default, f(full) break */
begin
/* Examples of New variable definitions.
NEW @mgrpsupdate5
set @mgrpsupdate5=
' case when #rac.cgrps=~@@@~ then ~shipcountry~
when @f1=#rac.shipcountry then ~shipcountry,productid,lastname,shipvia~

```

```

when @f2!=#rac.productid then ~productid,lastname,shipvia~
when @f3!=#rac.lastname then ~lastname,shipvia~
                        else ~shipvia~ end '
NEW update6
set @mgrpsupdate6=' char(39)+#rac.shipvia+char(39) '
NEW @mgrpsupdate4breakvalues
set @mgrpsupdate4breakvalues=
' when @f1!=#rac.shipcountry then char(39)+#rac.shipcountry+char(39) +~,~+
char(39)+#rac.productid+char(39) +~,~+ char(39)+#rac.lastname+char(39) +~,~+
char(39)+#rac.shipvia+char(39)
when @f2!=#rac.productid then char(39)+#rac.productid+char(39)+~,~+
char(39)+#rac.lastname+char(39)+~,~+char(39)+#rac.shipvia+char(39)
when @f3!=#rac.lastname then char(39)+#rac.lastname+char(39)+~,~+
char(39)+#rac.shipvia+char(39) '
*/
-- Store field and associated fields within it.
insert @groupwork#(agrp,bgrp,ard,brd)
select a.grp,b.grp,a.rd,b.rd
--select a.rd as a_rd,cast(a.grp as varchar(12)) as a_grp,
--b.rd as b_rd,cast(b.grp as varchar(12)) as b_grp
from @groupstable# as a inner join @groupstable# as b
on b.rd>=a.rd
order by a.rd,b.rd
--
select @mgrpsupdate5=@mgrpsupdate5+case when @str1=agrp then ',' else case when
@mgrpsupdate5>' then '~' else '' end+' when @f'+cast(ard as varchar(5))+
'!=#rac.'+agrp+' then ~' end+bgrp,
@mgrpsupdate4breakvalues=@mgrpsupdate4breakvalues+
case when @str1=agrp then '+char(39)+~,~+char(39)+' else case when
@mgrpsupdate4breakvalues>' then '+char(39)' else '' end+
' when @f'+cast(ard as varchar(5))+'!=#rac.'+agrp+' then char(39)+' end
+'#rac.'+bgrp,
@str1=agrp
from @groupwork# where ard<@k /* don't want last field */
order by rd
-- Fields
set @mgrpsupdate5=
' case when #rac.cgrps=@@@~ then ~'+(select grp from @groupstable# where rd=1)+'~'
+ @mgrpsupdate5 +'~ else ~'+ (select grp from @groupstable# where rd=@k)+'~ end '
-- Values
set @mgrpsupdate4breakvalues=@mgrpsupdate4breakvalues+'+char(39) '
set @mgrpsupdate6=' char(39)+#rac.'+(select grp from @groupstable# where
rd=@k)+'+char(39) '
--
end /* ends multibreak=f */
else
begin
--
/* This is partial breaks.For fields:shipcountry,productid,lastname,shipvia we
get:
For update5 (fields)
case when #rac.cgrps='@@@' then char(39)+'Totals'+char(39)
when @f1!=#rac.shipcountry then char(39)+#rac.shipcountry+char(39) +','+
char(39)+#rac.productid+char(39) +','+ char(39)+#rac.lastname+char(39) +','+
char(39)+#rac.shipvia+char(39)
else case when @f2=#rac.productid then '' else
char(39)+#rac.productid+char(39)+', ' end
+case when @f3=#rac.lastname then '' else char(39)+#rac.lastname+char(39)+', ' end
+ char(39)+#rac.shipvia+char(39) end
For values:

```

```

@break?=
  when @f1!=#rac.shipcountry then char(39)+#rac.shipcountry+char(39) +~,~+
char(39)+#rac.productid+char(39) +~,~+ char(39)+#rac.lastname+char(39) +~,~+
char(39)+#rac.shipvia+char(39)
  update6 (u6)=
    case when @f2=#rac.productid then ~ else char(39)+#rac.productid+char(39)+~,~
end+case when @f3=#rac.lastname then ~ else char(39)+#rac.lastname+char(39)+~,~ end+
char(39)+#rac.shipvia+char(39)
  */
  --
  select @mgrpsupdate5=@mgrpsupdate5+case when @mgrpsupdate5>' then '+' else '' end
+ 'case when @f'+cast(rd as varchar(3))+='#rac.'+grp + ' then ~ else ~'+grp+',~
end '
  from @groupstable#
  where rd>1 and rd<@k
  order by rd
  if @mgrpsupdate5>'
  set @mgrpsupdate5=@mgrpsupdate5+'+'+(select grp from @groupstable# where
rd=@k)+'~'
  else
  set @mgrpsupdate5='~'+(select grp from @groupstable# where rd=@k)+'~'
  --
  set @mgrpsupdate4breakfields=
  ' when @f1!=#rac.'+(select grp from @groupstable# where rd=1)+' then '+'
char(39)+@mgrpsfieldselect+char(39)
  set @mgrpsupdate5=' case when #rac.@grpcolinsert=~@@@~ then ~'+(select grp from
@groupstable# where rd=1)+'~'
+@mgrpsupdate4breakfields+' else '+@mgrpsupdate5+' end '
  -- @mgrpsupdate6 : for update
  select @mgrpsupdate6=@mgrpsupdate6+case when @mgrpsupdate6>' then '+' else '' end
+ 'case when @f'+cast(rd as varchar(3))+='#rac.'+grp + ' then ~ else
char(39)+#rac.'+grp+ '+char(39)+~,~ end'
  from @groupstable#
  where rd>1 and rd<@k
  order by rd
  if @mgrpsupdate6>'
  set @mgrpsupdate6=@mgrpsupdate6+'+ char(39)+#rac.'+(select grp from @groupstable#
where rd=@k)+'+char(39) '
  else
  set @mgrpsupdate6=' char(39)+#rac.'+(select grp from @groupstable# where
rd=@k)+'+char(39) '
  --
  set @mgrpsupdate4breakvalues=
  'when @f1!=#rac.'+(select grp from @groupstable# where rd=1)+' then'+
@mgrpsupdate4
  end /* ends multibreak=p(artial) */
  end /* end of rowbreak=y */
  --
end
--
if @display='m'
select @numvalues=count(*) from @trans#
else
set @numvalues=1
--
-- Check for aliases in @from (needed only for forcerows/partitions).If present in
form:
-- 'sometable as somename' create a table with alias and corresponding
tablename.This is
-- handled by function racaliasinfo#.

```

```

set @aliasck='n'
set @aliastable=''
if @forcerows='y' or @grpart>' ' or @pvtpart>' '
begin
if (charindex(' as ',@from)>0) and
(charindex('.',@grpcol)>0 or charindex('.',@pvtcol)>0)
begin
set @aliasck='y'
set @aliastable=@from
if charindex('[',@aliastable)>0
goto aliasfillin
resume5:
set @aliastable=replace(@aliastable, ' ', ',')
end
end
--
--          Check for row running sums
if @rowruns>' ' and @display='m'
goto rowruns_table
resume3:
-- Check for retaining column values.
if @colretain>' ' and @display='m'
begin
-- Check @all_cols field name
if @all_cols!='[All_cols]'
begin
if left(@all_cols,1)!='[' and right(@all_cols,1)!=']'
set @all_cols='['+@all_cols+']'
end
end
--          Check for @force values
--          First check if there a @ForceRange
if @forcerange>' '
begin
execute @check=racsp;16 /* ForceRange */ @forcerange=@forcerange,@style=@style,
@force=@force output
if (@check!=0) or (@force='')
begin
raiserror('sp racsp;16 (ForceRange) error',16,1)with nowait
return (-3)
end
end /* forcerange */
if @force>' '
set @forceparse=replace(@force, '~', '')
--          Check for rowfunctions
if @rowfunctions>' ' and @display='m'
goto rowfunctions_table
resume2:
--Perform check for rotate active and displayrowfunctions='m'
set @rowfmax=1
if (@rotate in ('y','nest')) and (@rowfunctions>' ') and (@displayrowfunctions='m')
begin
--numvalues is number of transforms
-- Will pass rowfmax as count of rowfunctions for each transform to rotate.
-- For normal rotate @rowfmax=1.
select @rowfcnt=count(*),@rowfmin=min(cnt),@rowfmax=max(cnt)
from
(select rtrans,count(*) as cnt
from @rowfunct#
group by rtrans) as a

```

```

if (@numvalues!=@rowfcnt) or (@rowfmin!=@rowfmax)
begin
--Here is error
raiserror('not all transforms in rowfunctions or unequal # of rowfuncts,cannot
rotate',0,1)with nowait
return(-3)
end
end /* end rotate displayrowf test */
--
-- Get grpcol and pivot field names and actual expression
-- Get @grpcol name/expression from procedure in case of 1 grouping field only
if @mgrps>''
set @grpcolinsert='cgrps' /* nameing concatenated grouping field 'cgrps' */
else
begin
execute @check=racsp;9 /* fieldname */
@grporpivot=1,@field=@grpcol,@fieldout=@grpcolinsert output,
@expression=@grpcol output
if @check!=0
begin
raiserror('sp racsp;9 (fieldname) error for @grpcol',16,1)with nowait
return(-3)
end
end
--Insert @grpcol expression into @transfields# for where default processing.
if @where=''
begin
if @mgrps=''
set @mgrpswhere=@grpcol
-- For multiple row fields
@mgrpswhere='datalength(field1)*datalength(field2)*datalenth(...'
insert @transfields# values(@mgrpswhere)
end
-- Get pivotcol fieldname
execute @check=racsp;9 /* fieldname */
@grporpivot=2,@field=@pvtcol,@fieldout=@pvtcolinsert output,
@expression=@pvtcol output
if @check!=0
begin
raiserror('sp racsp;9 (fieldname) error for @pvtcol',16,1)with nowait
return(-3)
end
--Insert @pvtcol expression into @transfields# for where default processing.
if @where=''
insert @transfields# values(@pvtcol)
-- Define sort for grouping field if no multiple groups
if @mgrps=''
begin
set @str1=@grpcol
execute @check=racsp;10 /* xtabsort */ @qryfld=@str1,
@fldsorttype=@grpsorttype,@fldsortnum=@grpsortnum,@fldsort=@grpsort output,
@sort=@sortgrp output
if @check!=0
begin
raiserror('grpcol sp racsp;10 (xtabsort) error',16,1)with nowait
return(-3)
end
end
-- Check if @grpcol is function.If yes make modification to @grpsort.
-- there is different expression for numeric/date and character data.

```

```

--
-- Define sort for pivot field
if (@rank is null) or (@sortagg!='y')
begin
execute @check=racsp;10 /* xtabsort */ @qryfld=@pvtcol,
@fldsorttype=@pvtsorttype,@fldsortnum=@pvtsortnum,@fldsort=@pivotsort output,
@sort=@sortpvt output
if @check!=0
begin
raiserror('pvtcol sp racsp;10 (xtabsort) error',16,1)with nowait
return(-3)
end
-- Check for max(datepart string for a multicell 1st run a.This condition produces an
error
-- for multicell 1st run because datepart( )+~^+string tries to concatenate an
integer
-- max(datepart with strings.So we're changing datepart back to datename.
If @multicell='y' and charindex('max(',@pivotsort)>0 and
charindex('datename',@pvtcol)>0
and charindex('datepart',@pivotsort)>0 and @pvtsortsub=''
set @pivotsort=replace(@pivotsort,'datepart','datename')
end
else
if (@rank is not null) and (@sortagg='y')
begin
-- Here sorting aggregate function
if @pvtsorttype='a'
set @pivotsort='asc'
else
if @pvtsorttype='d'
set @pivotsort='desc'
if @pvtsortsub=''
begin
select @func=func from @trans# where rd=1
set @pivotsort=@func+' '+@pivotsort
end
else
-- Here there is what should be aggregate function in @pvtsortsub we want to sort
by.
set @pivotsort=ltrim(@pvtsortsub)+' '+@pivotsort
end /* ends (@rank is not null) and (@sortagg='y') */
-- Define sort for grp and pivot totals
if @grandtotalposition = 'end'
set @grandtotalposition = 'asc'
else
set @grandtotalposition = 'desc'
if @rowtotalposition = 'begin'
set @rowtotalposition='desc'
else
set @rowtotalposition = 'asc'
-- Get default where
if @where=''
begin
set @str1=''
select @where=@where+case when @where !='' then '*datalength(' else 'datalength(' end
+ @str1 + field +')'
from @transfields#
order by rd
set @where=@where+'>0'
end

```



```

set @where1=@where
-- If @force is present add to @where (whatever @where is) if @forcetest='y'
if @force>' ' and @forcetest='y'
begin
-- Check for pvtpart (pivot partitioning).If yes force involved partition description.
if @pvtpart>' '
set @where=@where+' and #part2.t2 in ('+replace(@force, '&', ',')+')'
else
begin
set @where=@where+' and '+@pvtcol+' in ('+replace(@force, '&', ',')+')'
set @where1=@where
end
end
-- Get query transform expression (@qrytrans) and
if @getmxlenagg='y'
begin
set @str2='select @mxlenagg=max(len(cellvalue)) from '+@from
execute @check=sp_executesql @str2, N'@mxlenagg varchar(4) output',
@mxlenagg=@mxlenagg output
if @check!=0
begin
raiserror('error computing @mxlenagg',16,1)with nowait
return(-3)
end
end
--
set @valinsert=''
set @valcreate=''
-- Here are multiple aggregates in @transform
set @qrytrans=''
if @display='s'
begin
select @qrytrans=@qrytrans+case when @qrytrans>' ' then
'+'+char(39)+space(@space)+char(39)+'+' else '' end
+char(39)+'(' '+char(39)+'+cast('+funct+' as
varchar('+@mxlenagg+'))'+char(39)+')'+char(39)
from @trans#
order by rd
set @qrytrans=@qrytrans+' as value1'
set @valinsert='value1'
set @valcreate=@valinsert+' varchar('+@mxlenagg+')'
end /* end of single row display */
else
begin
-- here are multiple rows
select
@valcreate=@valcreate+case when @valcreate>' ' then ', ' else '' end
+'value'+cast(rd as varchar(4))+ ' varchar('+@mxlenagg+')',
@valinsert=@valinsert+case when @valinsert>' ' then ', ' else '' end
+'value'+cast(rd as varchar(4)),
@qrytrans=@qrytrans+case when @qrytrans>' ' then ', ' else '' end
+'cast('+funct+' as varchar('+@mxlenagg+')) as value'+cast(rd as varchar(4))
from @trans#
order by rd
end /* ends multiple row display */
-- Get qrystatements and lengths for table #rac
-- There are now 2 sp's for this.Qrystring is for case when there is
-- partitioning or forcerows='y' (for @grpcols).Qrystring1 is for no
-- partitioning/forcerows for grpcol and it's therefore not necessary
-- to know specific table field/function belongs to.

```



```

if @grpart>' ' or @forcerows='y'
begin
execute @check=racsp;11 /* qrystring */
@grporpivot=1,@qryfld=@grpcol,@qrytbl=@from,@aliasck=@aliasck,
@functionlen=@functionlen,@forcerows=@forcerows,@datelen=@datelen,@style=@style,
@partition=@grpart,@aliastable=@aliastable,@qrystring=@qrystatement1
output,@fldtype=@grpfldtype output,
@db=@grptable output,@char=@mxlengrp output
if @check!=0
begin
raiserror('sp racsp;11 (qrystring) error for grpcol',16,1)with nowait
return(-3)
end
end
else
if @mgrps=''
begin
-- Here use sp qrystring1
execute @check=racsp;12 /* qrystring1 */ @grporpivot=1,@qryfld=@grpcol,@qrytbl=@from,
@where1=@where1,@functionlen=@functionlen,@datelen=@datelen,@style=@style,
@qrystring=@qrystatement1 output,@fldtype=@grpfldtype output,@char=@mxlengrp output
if @check!=0
begin
raiserror('sp racsp;12 (qrystring1) error for grpcol',16,1)with nowait
return(-3)
end
end
else
begin
--
-- Here are multiple fields in @grpcols
--
execute @check=racsp;15 /* mqrystring1 */ @qryfld=@grpcol,@qrytbl=@from,
@where1=@where1,@qrystring=@qrystatement1 output,@fldtype=@grpfldtype output,
@char=@mxlengrp output
if @check!=0
begin
raiserror('sp racsp;15 (mqrystring1) error for multiple @grpcol fields',16,1)with
nowait
return(-3)
end
end
-- Check if @grpcol is function.If yes make modification to @grpsort.
-- there is different expression for numeric/date and character data.
/* if charindex('(',@grpcol)>0 and @mgrps=''
begin
if @grpfldtype='datetime'
set @grpsort='case when grouping('+@grpcol+')=1 then 9999 else
'+replace(@grpsort,')'),'')) end '
else
if charindex('int',@grpfldtype)>0
set @grpsort='case when grouping('+@grpcol+')=1 then 9999 else max('+@grpcol+') end '+
@sortgrp
else
set @grpsort='case when grouping('+@grpcol+')=1 then '@@@' else
cast(max('+@grpcol+') as varchar('+@functionlen+')) end '+' @sortgrp
end
end of grpcol is function and 1 row field */
if (charindex('(',@grpcol)>0 or @grpsortsub>'') and @mgrps=''
begin

```

```

if @grpsortsub=''
begin
set @str1=@grpcol
if @grpfldtype='datetime'
set @str1=replace(@str1, 'datetime', 'datepart')
end
else
set @str1=@grpsortsub
set @grpsort='case when grouping('+@grpcol+')=0 then max('+@str1+') end
'+@sortgrp
end

--
-- Check for grpcol partitioning
--
set @partgrpjoin=''
if @grpart>' '
begin
execute @check=racsp;6 /* partition */
@part=@grpart, @grpordpivot=1, @expression=@grpcol,
@field=@grpcolinsert, @fldtype=@grpfldtype, @table=@grptable, @aliasck=@aliasck,
@functionlen=@functionlen, @datelen=@datelen, @style=@style, @aliastable=@aliastable,
@partjoin=@partgrpjoin output
if @check!=0
begin
raiserror('sp racsp;6 (partition) error for grpcol',16,1)with nowait
return(-3)
end
-- Find max length of grpcol partition expression
set @str2=
'select @max=cast(max(len(t2)) as varchar(3)) from #part1'
exec sp_executesql @str2, N'@max varchar(3) output', @max=@mxlengrp output
--The length of @grpcol is the greater of @mxlengrp or 'totals'(6)
if cast(@mxlengrp as int) < 6
set @mxlengrp='6'
set @qrystatement1=
'case when grouping(@field1)=1 then cast(~@@@~ as varchar(@mxlengrp))
else cast(max(@field2) as varchar(@mxlengrp)) end as xgrp'
set @qrystatement1=replace(@qrystatement1, '@field1', '#part1.t1')
set @qrystatement1=replace(@qrystatement1, '@field2', '#part1.t2')
set @qrystatement1=replace(@qrystatement1, '@mxlengrp', @mxlengrp)
set @grpsort=replace(@grpsort, @grpcol, '#part1.t1')
end
-- Get @pvtcol qrystatement
set @partpvtjoin=''
if @pvtpart>' '
begin
execute @check=racsp;11 /* qrystring */
@grpordpivot=2, @qryfld=@pvtcol, @qrytbl=@from, @aliasck=@aliasck,
@functionlen=@functionlen, @forcerows=@forcerows, @datelen=@datelen, @style=@style,
@partition=@pvtpart, @aliastable=@aliastable, @qrystring=@qrystatement2
output, @db=@pvttable output,
@fldtype=@pvtfldtype output, @char=@mxlenpvt output
if @check!=0
begin
raiserror('sp racsp;11 (qrystring) error for pivotcol',16,1)with nowait
return(-3)
end
--
-- Now get pvtcol partitions
execute @check=racsp;6 /* partition */
@part=@pvtpart, @grpordpivot=2, @expression=@pvtcol,

```

```

@field=@pvtcolinsert,@fldtype=@pvtfldtype,@table=@pvttable,@aliasck=@aliasck,
@functionlen=@functionlen,@datelen=@datelen,@style=@style,@aliastable=@aliastable,
@partjoin=@partpvtjoin output
if @check!=0
begin
raiserror('sp racsp;6 (partition) error for pvtcol',16,1)with nowait
return(-3)
end
-- Modify @qrystatement2, same form as grpcol
-- Find max length of pivotcol partition expression
set @str2=
'select @max=cast(max(len(t2)) as varchar(3)) from #part2'
exec sp_executesql @str2,N'@max varchar(3) output',@max=@mxlenpvt output
if cast(@mxlenpvt as int) < 5
set @mxlenpvt='5'
set @qrystatement2=
'case when grouping(@field1)=1 then cast(~-@@@~ as varchar(@mxlenpvt))
else cast(max(@field2) as varchar(@mxlenpvt)) end as xcol'
set @qrystatement2=replace(@qrystatement2,'@field1','#part2.t1')
set @qrystatement2=replace(@qrystatement2,'@field2','#part2.t2')
set @qrystatement2=replace(@qrystatement2,'@mxlenpvt',@mxlenpvt)
-- Modify grpsort for partitioning.
set @pivotsort=replace(@pivotsort,@pvtcol,'#part2.t1')
end /* end of @pvtpart>' ' */
else
begin
-- Here no partitioning use sp qrystring1
execute @check=racsp;12 /* qrystring1 */
@grporpivot=2,@qryfld=@pvtcol,@qrytbl=@from,@where1=@where1,
@functionlen=@functionlen,@datelen=@datelen,@style=@style,
@qrystring=@qrystatement2 output,@fldtype=@pvtfldtype output,@char=@mxlenpvt output
if @check!=0
begin
raiserror('sp racsp;12 (qrystring1) error for pvtcol',16,1)with nowait
return(-3)
end
end
--
-- Modify pivotsort to reflect ordering in @force (if present)
if @force>' ' /* and @forcetest='y' */
begin
set @pivotsort='(select rd from racpovitin#
('+char(39)+@forceparse+char(39)+' '+'char(39)+'&'+char(39)+'') where
pivotin='+@pvtcol+') '+'@sortpvt
--'select * from racpovitin#
('+char(39)+@forceparse+char(39)+' '+'char(39)+'&'+char(39)+'')
if @pvtpart>' '
set @pivotsort=replace(@pivotsort,@pvtcol,'(select max(z3.t2) from #part2 as z3 where
z3.t1=#part2.t1)')
end
-- Construct table #rac definition
-- Appr. length of col=8000-(4 bytes rd+4 bytes key1+@mxlenpvt+@mxlengrp+
-- 50 [value with possible multiple aggregs])
set @work=7950-(10+(@numvalues*cast(@mxlenagg as int))+cast(@mxlengrp as int)
+cast(@mxlenpvt as int))
-- Further adjust @work for multiple groups.Using 50 chars per field for now
if @mgrps>' '
begin
--set @check=53*(( len(@mgrpsupdate3)-len(replace(@mgrpsupdate3,',','')))+1) /* finds
# of fields */

```

```

-- @mlen is set length of each multiple row field.Now a parameter.
set @check=(cast(@mlen as int)+3) * (( len(@mgrpsupdate3)-
len(replace(@mgrpsupdate3,',','')))+1) /* finds # of fields */
set @work=@work-@check
end
set @mxlencol=cast(@work as varchar(4))
set @str1=
'alter table #rac add '+@mgrpsfieldsdef + '@grpcolinsert varchar(@mxlengrp),
@pvtcolinsert varchar(@mxlenpvt),@valcreate,key1 int null,
col varchar(@mxlencol) default ~'
set @str1=replace(@str1,'@grpcolinsert',@grpcolinsert)
set @str1=replace(@str1,'@pvtcolinsert',@pvtcolinsert)
set @str1=replace(@str1,'@valcreate',@valcreate)
set @str1=replace(@str1,'@mxlengrp',@mxlengrp)
set @str1=replace(@str1,'@mxlenpvt',@mxlenpvt)
set @str1=replace(@str1,'@mxlencol',@mxlencol)
set @str1=replace(@str1,'~',char(39))
exec(@str1)
set @check=@@error
if @check!=0
begin
raiserror('#rac table definition error',16,1)with nowait
return(-3)
end
--Modify grpcol multicell 2nd run to reflect ordering in multicell 1st run
if @multicellrun2='y' and @multicell='n'
set @grpsort='(select a.rd from #multicellzz1 as a where a.'+@grpcol+'=b.'+@grpcol+')'
--Note for above:the 'b' in ""=b.'+@grpcol+')" refers to alias 'b' in the @from
string
--passed in for the 2nd multicell run.It is the alias for the derived table.The above
query
-- returns @grpcol in the exact order it was sorted in the multicell 1st run.No
asc/desc is
--needed to append to the query.
--Insert into #rac
set @str1=
'insert into #rac(@mgrpsfieldselect @grpcolinsert,@pvtcolinsert,@valinsert)
select @mgrpsqryselect
@qrystatement1 ,
@qrystatement2 ,
@qrytrans
from @from
where @where
group by @grpcol,@pvtcol with cube
order by
grouping(@grpcol)@grandtotalposition,@grpsort,grouping(@pvtcol)@rowtotalposition,@pi
votsort'
if @totals_only='y'
set @str1=replace(@str1,'with cube',
'with cube having grouping('+@grpcol+')>0')
if @row_totals='n' and @rank is null
set @str1=replace(@str1,'with cube',
'with cube having grouping('+@pvtcol+')=0')
if @rank is not null
begin
if @row_totals='y'
set @str1=replace(@str1,'with cube',
'with cube having grouping('+@grpcol+')=0')
else
set @str1=replace(@str1,'with cube',

```

```

        'with cube having grouping('+@grpcol+')=0 and grouping('+@pvtcol+')=0')
end
-- Modify pvtort if they are datetime and pvtortnum=y
-- No longer using @grpsortnum in 2nd multicell run (using sort order of #multicellzz1
table)
-- to get correct sort order by rows.Only using @pvtortnum
--if @grpfldtype='datetime' and @grpsortnum='y'
--set @grpsort=replace(@grpsort,'as int'),'as datetime')
if @pvtfldtype='datetime' and @pvtortnum='y'
set @pvtort=replace(@pvtort,'as int'),'as datetime')
--
set @str1=replace(@str1,'@grpcolinsert',@grpcolinsert)
set @str1=replace(@str1,'@pvtcolinsert',@pvtcolinsert)
set @str1=replace(@str1,'@qrystatement1',@qrystatement1)
set @str1=replace(@str1,'@qrystatement2',@qrystatement2)
set @str1=replace(@str1,'@valinsert',@valinsert)
set @str1=replace(@str1,'@qrytrans',@qrytrans)
if @mgrps>' '
begin
-- Multiple groups
set @grpsort=@mgrps
set @str1=replace(@str1,'@mgrpsfieldselect',@mgrpsfieldselect+',')
set @str1=replace(@str1,'@mgrpsqryselect',@mgrpsqryselect)
end
else
begin
-- No multiple groups
set @str1=replace(@str1,'@mgrpsfieldselect',' ')
set @str1=replace(@str1,'@mgrpsqryselect',' ')
end
--if @grpsortsub>' '
--set @grpsort='max('+@grpsortsub+')'+@sortgrp
set @str1=replace(@str1,'@grpsort',@grpsort)
--
if (@sortagg!='y') and (@pvtpart='') and (@force='') and (@pvtortsub>' ')
set @pvtort='max('+@pvtortsub+')'+@sortpvt
--
set @str1=replace(@str1,'@pvtort',@pvtort)
set @str1=replace(@str1,'@grandtotalposition',@grandtotalposition)
set @str1=replace(@str1,'@rowtotalposition',@rowtotalposition)
if @pvtpart=''
set @str1=replace(@str1,'@pvtcol',@pvtcol)
else
set @str1=replace(@str1,'@pvtcol','#part2.t1')
if @grpart=''
set @str1=replace(@str1,'@grpcol',@grpcol)
else
set @str1=replace(@str1,'@grpcol','#part1.t1')
-- @pvt is for pivot partitioning,its '' for non-partitioning
set @str1=replace(@str1,'@from',@from+@partgrpjoin+@partpvtjoin)
set @str1=replace(@str1,'@where',@where)
set @str1=replace(@str1,'~',char(39))
if len(@str1)>7979
begin
raiserror('you query is too big,for RAC to process',0,1) with nowait
return(-3)
end
-- Check that @from and @where is not an empty table
set @check=2
set @str2=

```

```

'If not exists(select top 1 * from @from where @where)
set @check=1
set @str2=replace(@str2, '@from', @from)
set @str2=replace(@str2, '@where', @where)
set @str2=replace(@str2, '~', char(39))
execute @k=sp_executesql @str2, N'@check int output', @check=@check output
if @k!=0
begin
raiserror('error checking for empty table(s)',16,1) with nowait
return(-3)
end
if @check=1
begin
raiserror('your table(s) and where clause do not return any rows',0,1) with nowait
return(-3)
end
--
if @printqry='y'
print 'RAC query: '+char(13)+right(@str1, len(@str1)-(charindex('select', @str1)-
1))+char(13)+char(13)
--
exec(@str1)
set @check=@@error
if @check!=0
begin
raiserror('#rac insert error',16,1)with nowait
return(-3)
end
--
--          Create Indexes on #rac
--
set @str1='create unique clustered index grpcol_pk on #rac (rd)'
exec(@str1)
if @@error!=0
begin
raiserror('#rac clustered index on (rd) error',16,1)with nowait
return(-3)
end
--
/* set @str1='create unique index grpcol_pk1 on #rac
(rd, '+@grpcolinsert+', '+@pvtcolinsert+')'
exec(@str1)
if @@error!=0
begin
raiserror('#rac create index grpcol_pk1 error',16,1)with nowait
return(-3)
end
--
-- Not sure the covering index is still needed but keeping it 12/03/00
--
-- Create covering index if grpcolvalue is involved
if @rowruns>' ' or @rowfunctions>' '
begin
-- Create covering index for each particular @grpcol, @pvtcol and value
set @str1=''
select
    @str1=@str1+case when @str1>' ' then ' ' else '' end
    +'create index grpvtvalue'+cast(rd as varchar(5))+ ' on #rac ('+
    @grpcolinsert+', '+@pvtcolinsert+', value'+cast(rd as varchar(5))+')'
from

```

```

        (select a.rd as rd
        from @trans# as a inner join @rowfuncts# as b on b.rtrans=a.alias
        union
        select d.rd
        from @trans# as d inner join @rowsums# as e on e.rtrans=d.alias) as c
        order by rd
--set @str1='create index grpvalues1 on #rac (product_id,store_id,value1)
--      create index grpvalues2 on #rac (product_id,store_id,value2)
--      create index grpvalues3 on #rac (product_id,store_id,value3)'
        exec(@str1)
        if @@error!=0
        begin
            raiserror('create covering index error',16,1)with nowait
            return(-3)
        end
    end
*/
--
--      Create xtab table definition
-- First get max length of [value] and @grpcol
-- Get max column length of [value] for xtab definition
-- from #rac.Compare this length with default value (@emptycell).The actual
-- max length is the greater of the two.
if @numtrans=1 or @display='s'
begin
    set @str2=
    'select @maxlen=cast(max(len(value1)) as varchar(4)) from #rac'
    execute sp_executesql @str2,N'@maxlen varchar(4) output',@maxlen=@maxlen output
end
else
    begin
        -- here is multiple value columns
        set @rd=1
        while @rd<=@numvalues
        begin
            --select @str1=valuei from #values where rd=@rd
            select @str1='value'+cast(@rd as varchar(5))
            set @str2=
            'select @max=case when max(len(@str1))>isnull(@max,0) then max(len(@str1)) else @max
            end
            from #rac'
            set @str2=replace(@str2,'@str1',@str1)
            exec sp_executesql @str2,N'@max int output',@max=@maxlen output
            set @rd=@rd+1
        end
        /* end of multi-row display */
        -- get length of row running sum field
        if @rowruns>' ' and @display='m'
        begin
            set @flag1=''
            -- store maxlen for individual row run
            set @runsfieldsingle=@maxlen
            -- Check for special case where max is used in transform
            -- to get runs for individual values (in this case maxlen is wrong).
            if @numvalues=1 and (select charindex('max',func) from @trans#
                                where rd=1)>0
            begin
                set @runsfieldsingle='12'
                set @runsfieldlen='13'
                set @flag1='y' /* tells sp_rowruns to change decimal( ) format */
            end
        end
    end

```



```

end
else
begin
-- find max count over all functions (here assuming agg's of sum,count
select @check=max(cnt) from
(select count(*) as cnt from @rowsums#
group by rtrans) as a
set @runsfieldlen=@check*(cast(@maxlen as int))+(@check-1)
end
end
-- Check for @rowfunctions
set @rowfunctfield=''
set @rowfunctfieldselect=''
if @rowfunctions>' '
begin
-- using this formula to estimate size of rowf. field
-- maxcntrowf*(9+@maxlen+mxlenpvt)
select @check=max(cnt) from
(select count(*) as cnt from @rowfuncts#
group by rtrans) as a
set @check=(@check*(9+cast(@maxlen as int))+cast(@mxlenpvt as int))
set @rowfunctlen=cast(@check as varchar(3))
set @rowfunctfield=[Rowfunct] varchar('+@rowfunctlen+')default~, '
set @rowfunctfieldselect=[Rowfunct], '
--estimate single rowfunction if displayrowfunction='m'
if @displayrowfunctions='m'
begin
if exists(select * from @rowfuncts# where rowfunct in('min','max'))
set @rowfunctlen1=cast(@maxlen as int)+cast(@mxlenpvt as int)+6
else
if exists(select * from @rowfuncts# where rowfunct ='avg')
set @rowfunctlen1=cast(@maxlen as int)+5
else
--here its count
set @rowfunctlen1=cast(@maxlen as int)+7
end /* end displayfunctions='m' */
end /* ends rowfunction>' ' */
-- Adjust @maxlen if cruntrans>' '
if @colruns>' ' and @display='m'
set @maxlen=(2*cast(@maxlen as int))+1
-- Check cpercents
if (@cpercents>' ' or @rpercents>' ') and @display='m'
begin
-- Check if there's aggregates other than count and sum in #trans.
if exists(select * from @trans#
where charindex('count',funct)=0 and charindex('sum',funct)=0)
begin
if @cpercents>' '
set @maxlen=cast(@maxlen as int)+10
if @rpercents>' '
set @maxlen=cast(@maxlen as int)+10
end
else
begin
if @cpercents>' '
set @maxlen=cast(@maxlen as int)+7
if @rpercents>' '
set @maxlen=cast(@maxlen as int)+7
end
end
end
end

```



```

-- Change @maxlen if < max of (@emptycell,@nullcell)
if len(@nullcell)<len(@emptycell)
begin
if cast(@maxlen as int) < len(@emptycell)
set @maxlen=cast(len(@emptycell) as varchar(3))
end
else
begin
if cast(@maxlen as int) < len(@nullcell)
set @maxlen=cast(len(@nullcell) as varchar(3))
end
-- Get pvtcol fields for @xtab if @cutpvtcols='n'
if @cutpvt='n'
begin
set @fields=''
set @fieldselect=''
if @force='' and @rank is null
begin
-- Get fields with usual technique for all runs except multicell 2nd run
-- Following logic tests for 2nd RAC run.
if @multicellrun2='y' and @multicell='n'
begin
-- This is new technique that gets exact length for the columns (@pvtcol)
-- First the correct order column fields are obtained using usual method.Then for
each column
-- the varchar length definition and default is obtained and inserted into the the
@fields str.
set @str2=
'select @fields=@fields+case when @fields>~ then ~,~ else ~ end
+[-+@pvtcolinsert+ ~]~
from #rac
where @grpcolinsert=@@@~
order by rd '
set @str2=replace(@str2, '@grpcolinsert', @grpcolinsert)
set @str2=replace(@str2, '@pvtcolinsert', @pvtcolinsert)
set @str2=replace(@str2, '~', char(39))
exec @check=sp_executesql @str2, N'@fields varchar(8000) output', @fields=@fields output
if @check!=0
begin
raiserror('Error for obtaining correct fields in order for multicell
report',16,1)with nowait
return(-3)
end
-- Store fieldselect
set @fieldselect=@fields
--
-- This part uses grouping query to find max length (of value1) for each column.
-- This technique is good for only 1 transform value.The order the columns come back
in
-- is unimportant as the correct order was obtained in the first part.The replace
function
-- is used to insert the varchar length and default.We don't have to worry here about
the
-- 'Totals' column as multicell report doesn't have them.
--
set @str2=
'select @fields=replace(@fields, ~[-+@pvtcolinsert+ ~]~, ~[-+@pvtcolinsert+ ~]varchar(~+
case when max(len(value1))>=len(~@emptycell~) then
cast(max(len(value1)) as varchar(4)) else cast(len(~@emptycell~) as varchar(2))
end

```

```

        +-)default~~@emptycell~~)
from #rac
where @grpcolinsert!=@@@~
group by @pvtcolinsert
order by @pvtcolinsert'
set @str2=replace(@str2,'@pvtcolinsert',@pvtcolinsert)
set @str2=replace(@str2,'@emptycell',@emptycell)
set @str2=replace(@str2,'@grpcolinsert',@grpcolinsert)
set @str2=replace(@str2,'~',char(39))
exec @check=sp_executesql @str2,N'@fields varchar(8000) output',@fields=@fields output
if @check!=0
begin
    raiserror('Error for obtaining correct fields definition for multicell
report',16,1)with nowait
    return(-3)
end
end /* ends new technique for multicell report */
else
begin
-- Usual technique for obtaining @fields definition.
set @str2=
'select @fields=@fields+case when @fields>~ then ~,~ else ~ end
+[-+@pvtcolinsert+ ~]varchar(@maxlen)default~~@emptycell~~,
@fieldselect=@fieldselect+case when @fieldselect>~ then ~,~ else ~ end
+[-+@pvtcolinsert+ ~]~
from #rac
where @grpcolinsert=@@@~
order by rd '
set @str2=replace(@str2,'@grpcolinsert',@grpcolinsert)
set @str2=replace(@str2,'@pvtcolinsert',@pvtcolinsert)
set @str2=replace(@str2,'@maxlen',@maxlen)
set @str2=replace(@str2,'@emptycell',@emptycell)
set @str2=replace(@str2,'@pivotstort',@pivotstort)
set @str2=replace(@str2,'~',char(39))
exec @check=sp_executesql @str2,N'@fields varchar(8000) output,@fieldselect
varchar(8000) output',
@fields=@fields output,@fieldselect=@fieldselect output
if @check!=0
begin
    raiserror('Error for obtaining fields/fields definition',16,1)with nowait
    return(-3)
end
set @fieldselect=replace(@fieldselect,'-@@@','Totals')
-- Store fieldselect in @fieldspvt in case displayrowfunctions='m' (for sp
displayrowfunctions)
set @fieldspvt=@fieldselect
end /* ends usual techique for @fields definition */
end /* ends @force='' and @rank is null */
else
if @force='' and @rank is not null
begin
-- If @ranklimit>0 then at this row_totals are wrong (they relect all pvt columns
not
-- just those in the limit.So for now we'll shut off @row_totals.This situation
holds
-- no matter what value is for @sortagg.
--if @ranklimit>'0'
--set @row_totals='n'
--print '@rowtotals: '+@row_totals
-- Get fields for rank based on max count of @grpcol/@ranklimit (if active).

```

```

set @str2=
' if @ranklimit>0
  set @maxcnt=@ranklimit
  else
  begin
    select @maxcnt=max(cnt1)
    from (select count(*) as cnt1 from #rac
    group by @grpcolinsert) as a
    if @row_totals=y~
    set @maxcnt=@maxcnt-1
  end
while @k<=@maxcnt
begin
  select @fields=@fields+case when @fields>~ then ~,~ else ~ end
  +[~+@rank+cast(@k as varchar(4))+ ~]varchar(@maxlen)default~@emptycell~,
  @fieldselect=@fieldselect+case when @fieldselect>~ then ~,~ else ~ end+
  +[~+@rank+cast(@k as varchar(4))+ ~]~
  set @k=@k+1
end
set @str2=replace(@str2, '@grpcolinsert', @grpcolinsert)
set @str2=replace(@str2, '@maxlen', @maxlen)
set @str2=replace(@str2, '@emptycell', @emptycell)
set @str2=replace(@str2, '~', char(39))
exec @check=sp_executesql @str2, N'@rank varchar(50), @k int, @maxcnt int, @row_totals
varchar(1), @ranklimit int,
@fields varchar(8000) output, @fieldselect varchar(8000)
output', @k=1, @maxcnt=0, @rank=@rank, @row_totals=@row_totals,
@ranklimit=@ranklimit, @fields=@fields output, @fieldselect=@fieldselect output
if @check!=0
begin
  raiserror('Error for obtaining correct ranked fields/fields
definition', 16, 1) with nowait
  return(-3)
end
--Append totals if necessary to ranked fields
if @row_totals='y'
begin
  if @rowtotalposition='desc'
  begin
    set @fields='[-@@@]varchar('+@maxlen+')default'+'''+@emptycell+'''+', '+@fields
    set @fieldselect='[Totals], '+ @fieldselect
  end
  if @rowtotalposition='asc'
  begin
    set @fields=@fields+', [-@@@]varchar('+@maxlen+')default'+'''+@emptycell+'''+
    set @fieldselect=@fieldselect + ', [Totals]'
  end
end
end
end
else
begin
  --
  --
  -- This is Force
  --
  -- Requiring fields be specified in order so order them by rd.
  -- Get fields with usual technique for force for all runs except multicell 2nd run
  -- Following logic tests for 2nd RAC run.
  if @multicellrun2='y' and @multicell='n'
  begin
    -- This is new technique that gets exact length for the columns (@pvtcol)

```

```
-- First the correct order column fields are obtained using usual method. Then for
each column
-- the varchar length definition and default is obtained and inserted into the the
@fields str.
-- Third for columns in force not in data insert varchar() and default. Use length of
emptycell
-- as length of varchar.
-- was 'from #povitin ' (below)
set @str2=
'select @fields=@fields+case when @fields>~ then ~,~ else ~ end
+~[~pivotin+ ~]~
from racpovitin# (@aug1,@aug2)
order by rd'
set @str2=replace(@str2,'~',char(39))
exec sp_executesql @str2,N'@fields varchar(8000) output,@aug1 varchar(2400),@aug2
varchar(1)',
@aug1=@forceparse,@aug2='&,@fields=@fields output
if @check!=0
begin
raiserror('Error for obtaining correct force fields for multicell report',16,1)with
nowait
return(-3)
end
set @fieldselect=@fields
-- Now find max(length) of each column and replace column with column plus varchar
length
-- and default emptycell.
set @str2=
'select @fields=replace(@fields,~[~+@pvtcolinsert+ ~],~[~+@pvtcolinsert+ ~]varchar(~+
case when max(len(value1))>=len(~@emptycell~) then
cast(max(len(value1)) as varchar(4)) else cast(len(~@emptycell~) as varchar(2))
end
+~)default~@emptycell~)
from #rac
where @grpcolinsert!=@@@~
group by @pvtcolinsert
order by @pvtcolinsert'
set @str2=replace(@str2,'@pvtcolinsert',@pvtcolinsert)
set @str2=replace(@str2,'@emptycell',@emptycell)
set @str2=replace(@str2,'@grpcolinsert',@grpcolinsert)
set @str2=replace(@str2,'~',char(39))
exec @check=sp_executesql @str2,N'@fields varchar(8000) output,@aug1 varchar(2400),
@aug2 varchar(1)',@aug1=@forceparse,@aug2='&,@fields=@fields output
if @check!=0
begin
raiserror('Error for obtaining correct fields definition for force for multicell
report',16,1)with nowait
return(-3)
end
--Replace any fields in force and not in #rac (data) with length/default of emptycell
set @f1=len(@emptycell)
if @f1=0
set @f1=1
set
@fields=replace(@fields+',','[',',']varchar('+@f1+')default'+'''+@emptycell+'''+',')
--Check if last character is ','. If true get rid of it.
if right(@fields,1)=','
set @fields=substring(@fields,1,len(@fields)-1)
end /* end of multicell force computation */
else
```

```

begin
--      Usual technique for force
-- #povitin was here (below)
set @str2=
'select @fields=@fields+case when @fields>~ then ~,~ else ~ end
+[-+pivotin+]varchar(@maxlen)default~@emptycell~,
@fieldselect=@fieldselect+case when @fieldselect>~ then ~,~ else ~ end
+[-+pivotin+ ~]~
from racpovitin# (@aug1,@aug2)
order by rd'
-- Note that racpovitin# (@aug1,@aug2) was formally:
--racpovitin# ('+char(39)+@forceparse+char(39)+', '+char(39)+'&'+char(39)+').Now
arguments are passed in.
set @str2=replace(@str2, '@grpcolinsert', @grpcolinsert)
set @str2=replace(@str2, '@pvtcolinsert', @pvtcolinsert)
set @str2=replace(@str2, '@maxlen', @maxlen)
set @str2=replace(@str2, '@emptycell', @emptycell)
set @str2=replace(@str2, '@pivotsort', @pivotsort)
set @str2=replace(@str2, '~', char(39))
exec @check=sp_executesql @str2,N'@fields varchar(8000) output,@fieldselect
varchar(8000) output,@aug1 varchar(2400),
@aug2 varchar(1)',@aug1=@forceparse,@aug2='&',@fields=@fields
output,@fieldselect=@fieldselect output
if @check!=0
begin
raiserror('Error for obtaining correct force fields definition',16,1)with nowait
return(-3)
end
end /* end of usual technique for force */
end /* end of this is @force */
if @force>' ' and @row_totals='y'
begin
if @rowtotalsposition='desc'
begin
set @fields='[-@@@]varchar('+@maxlen+')default'+'''+@emptycell+'''+', '+@fields
set @fieldselect='[Totals], '+ @fieldselect
end
if @rowtotalsposition='asc'
begin
set @fields=@fields+', [-@@@]varchar('+@maxlen+')default'+'''+@emptycell+'''+
set @fieldselect=@fieldselect + ', [Totals]'
end
end
end /* end @cutpvt='n' */
else
begin
--      here @cutpvt='y'
set @fields='[-@@@]varchar('+@maxlen+')default'+'''+@emptycell+'''+
set @fieldselect='[totals]'
end /* ends @cutpvt='y' */
-- Correct totals fields (-@@@) for colpercents
if @rpercentstotals='' and @cpercents>' '
begin
if exists(select * from @trans#
where charindex('count',func)=0 and charindex('sum',func)=0)
set @fields=replace(@fields, '[-@@@]varchar('+@maxlen+')',
'[-@@@]varchar('+cast( (@maxlen as int)-10) as varchar(5))+')')
else
set @fields=replace(@fields, '[-@@@]varchar('+@maxlen+')',
'[-@@@]varchar('+cast( (@maxlen as int)-7) as varchar(5))+')')

```

```

end
else
if @rpercentstotals>' ' and @cpercents=' '
begin
if exists(select * from @trans#
        where charindex('count',func)=0 and charindex('sum',func)=0)
    set @fields=replace(@fields,'[-@@@]varchar('+@maxlen+')',
        '[-@@@]varchar('+cast( (cast(@maxlen as int)+10) as varchar(5))+')')
    else
    set @fields=replace(@fields,'[-@@@]varchar('+@maxlen+')',
        '[-@@@]varchar('+cast( (cast(@maxlen as int)+7) as varchar(5))+')')
    end
-- Check for retain,if yes compute field size and
-- Append field 'All_cols' to fields to hold final retain/concatenation.
if @colretain>' '
begin
    set @str2=
        'select @cnt=max(cnt1)
        from (select count(*) as cnt1 from #rac
        where @grpcolinsert not in (~@@@~,~~@@@~)
        group by @grpcolinsert) as a '
        set @str2=replace(@str2,'@grpcolinsert',@grpcolinsert)
        set @str2=replace(@str2,'~',char(39))
        execute sp_executesql @str2,N'@cnt int output',@cnt=@check output
        set @check=(@check*cast(@maxlen as int))+ ((@check-1)*len(replace(@separator,
        ',','x'))))
        if @cutpvt='n' and @row_totals='y'
        begin
            set @fields=stuff(@fields,charindex(', ',@fields)+1,0,
                @all_cols+'varchar('+cast(@check as
varchar(4))+')default'+'''+@emptycell+'''+',')
            set @fieldselect=stuff(@fieldselect,charindex(', ',@fieldselect)+1,0,@all_cols+',')
        end
        else
        if @cutpvt='n' and @row_totals='n'
        begin
            set @fields=@all_cols+'varchar('+cast(@check as
varchar(4))+')default'+'''+@emptycell+'''+','+@fields
            set @fieldselect=@all_cols+', '+@fieldselect
        end
        else
        if @cutpvt='y' and @row_totals='y'
        begin
            set @fields=@fields+', '+@all_cols+'varchar('+cast(@check as
varchar(4))+')default'+'''+@emptycell+'''+','+@fields
            set @fieldselect=@fieldselect+', '+@all_cols
        end
        else
        if @cutpvt='y' and @row_totals='n'
        begin
            set @fields=@all_cols+'varchar('+cast(@check as
varchar(4))+')default'+'''+@emptycell+'''+','+@fields
            set @fieldselect=@all_cols
        end
        else
        begin
            raiserror('error for @cutpvt specification(s)',16,1)with nowait
            return(-3)
        end
    end
end

```

```

-- Check for @printagg
if @printagg='n'
begin
set @str1=''
set @printaggfieldselect=''
end
else
begin
if @display='m'
begin
-- determine largest field size of aggregates
select @str1=cast(max(len(alias)) as varchar(75)) from @trans#
-- set @str1='Funct varchar('+@str1+')default~, '
set @str1=@translabel+' varchar('+@str1+')default~, '
set @printaggfieldselect=@translabel+', '
end
else
begin
-- Here display='s'
set @printaggs=''
select @printaggs=@printaggs+case when @printaggs!='' then ','
else '' end +[alias]
from @trans#
set @str1=cast(len(@printaggs) as varchar(3))
-- set @str1='Funct varchar('+@str1+')default~, '
set @str1=@translabel+' varchar('+@str1+')default~, '
set @printaggfieldselect=@translabel+', '
end /* end display='s' */
end
--
-- Check @rowruns
--
set @runsfield=''
set @runsfieldselect=''
if @rowruns>' ' and @display='m'
begin
--
if charindex('{',@rowruns)>0 and @mgrps=''
begin
raiserror('optional {rowfield(s)} in @rowruns not allowed for a one field
@grpcol',0,1)with nowait
return(-3)
end
--
-- Check that fastest varying (right most) @grpcol field is not in @rowruns
select @testruns=ltrim(rtrim(grp)) from @groupstable# where rd=@kcheck
if (exists(select 1 from @rowsums# where rowfield=@testruns)) and (@testruns>' ')
begin
raiserror('fastest varying rowfield '%s' is illegal in @rowruns,omit it and run
over table',0,1,@testruns)with nowait
return(-3)
end
--
set @testruns=''
select top 1 @testruns=runfuncterm
from @rowsums#
--
-- A right left parenthesis is missing around @transform
expression/alias.
where (charindex('(',runfuncterm) * charindex(')',runfuncterm)=0)
or
--
-- Missing right or left squiggly brackets around rowfield

```

```

(exists(select 1 from @groupstable#
        where (charindex(ltrim(rtrim(grp)),runfunctionterm)>0)
               and (charindex('{',runfunctionterm)*charindex('}',runfunctionterm)=0)))
or
--          Rowfield is not in @grpcols
(rowfield>' and not exists(select 1 from @groupstable#
                           where rowfield=ltrim(rtrim(grp))))
or
--          Rtrans(exp/alias) not present in @transform expression/alias
(not exists(select 1 from @trans#
               where rtrans=ltrim(rtrim(alias))))
order by rd
if @testruns>'
begin
raiserror('@rowruns expression '%s' contains an error,check pivot
column,(alias),{rowfield}',0,1,@testruns)with nowait
return (-3)
end
--
--          Test that runcol's are in pivot fields
set @testruns=''
select top 1 @testruns=runcol
from @rowsums#
where charindex(['+runcol+'],'',replace(@fieldselect,'-@@@','Totals'))=0
order by rd
if @testruns>'
begin
raiserror('column '%s' in @rowruns is not in pivot columns',0,1,@testruns)with
nowait
return (-3)
end
-- for now using @maxlen (see above).
-- set @runsfield='Runs varchar('+@runsfieldlen+')default~, '
if @rowrunslabel!='[Runs]'
set @rowrunslabel= '['+replace(replace(@rowrunslabel,['','']),['',''])+']'
set @runsfield=@rowrunslabel+' varchar('+@runsfieldlen+')default~, '
set @runsfieldselect=@rowrunslabel+', '
end
--
-- Check for @diffs
--
set @difffields=''
set @difffieldselect=''
if @diffs>'
begin
set @difffields='[Diff] varchar(15)default~, '
set @difffieldselect='[Diff], '
--set @difffield=~Diff~' /* for update */
--set @diffvalue=~+char(39)+@diff1+char(39) ' /* for update */
--set @diffdeclare=@diff1 varchar(15), '
--set @diff0=@diff1=~0~, '
end
-- Check @rowcnters
--
set @ctfields=''
set @ctfieldselect=''
set @wherecntersupdate1=' '
set @wherecntersupdate2=' '
if @rowcnters>' /* and @display='m' @rowcnters should work with display=s I think
*/

```



```

begin
--          Check for a single field @grpcol
if @mgrps=''
begin
raiserror('row counter field(s) (@rowcnters) not allowed for a one field
@grpcol',0,1)with nowait
return(-3)
end
-- Check that each field in @rowcnters is in @groupstable# table (@grpcol).
-- Query eliminates user input from @rowcnter field to compare to grpcol field.
/* old code
if exists(select pivotin
from racpovitin#(@rowcnters, '&') as a left join @groupstable# as b
on grp=pivotin
where grp is null)

if exists(select x1
from (select replace(pivotin, '{'+ltrim(case when charindex('{',pivotin)>0 and
charindex('}',pivotin)>0 then
substring(pivotin,charindex('{',pivotin)+1,
((charindex('}',pivotin)-1) - (charindex('{',pivotin)))) else '' end)+'}', '') as x1
from racpovitin#(@rowcnters, '&')) as a left join @groupstable# as b
on grp=x1 where grp is null )
*/
--
set @cntername=''
select top 1 @cntername=x1
from
(select rd, ltrim(rtrim(case when charindex('{',pivotin)>0 and charindex('}',pivotin)>0
then replace(replace(replace(pivotin, substring(pivotin, charindex('{',pivotin)+1,
((charindex('}',pivotin)-1) - (charindex('{',pivotin))))), '{', ''), '{', ''))
else pivotin end)) as x1
from racpovitin#(@rowcnters, '&')) as a
where x1 not in
(select ltrim(rtrim(grp)) from @groupstable# )
order by rd
if @cntername>''
begin
raiserror('@rowcnters field '%s' is not in multi row fields (@grpcol),review all
field(s)',0,1,@cntername)with nowait
return(-3)
end
--
--          Check that fastest varying (right most) @grpcol field is not in @rowcnters
--
/* if exists(select * from racpovitin#(@rowcnters, '&')
where replace(pivotin, '{'+ltrim(case when charindex('{',pivotin)>0 and
charindex('}',pivotin)>0 then
substring(pivotin,charindex('{',pivotin)+1,
((charindex('}',pivotin)-1) - (charindex('{',pivotin)))) else '' end)+'}', '')=
(select top 1 grp from @groupstable# order by rd desc))
*/
select @cntername=ltrim(rtrim(grp)) from @groupstable# where rd=@kcheck
if exists(select x1
from
(select rd, ltrim(rtrim(case when charindex('{',pivotin)>0 and charindex('}',pivotin)>0
then replace(replace(replace(pivotin, substring(pivotin, charindex('{',pivotin)+1,
((charindex('}',pivotin)-1) - (charindex('{',pivotin))))), '{', ''), '{', ''))
else pivotin end)) as x1
from racpovitin#(@rowcnters, '&')) as a

```

```

where x1=@cntername)
begin
raiserror('a row counter is not available for the last (fastest varying) row field
'%s'',0,1,@cntername)with nowait
return(-3)
end
--
--set @ctfield='Cnters varchar(15)default~, '
--set @ctfieldselect='Cnters,'
--
--      New code for separate columns for counters
--
/*      @ctfields='Cnter1 varchar(5),CnterN varchar(5),....' (no default)
      @ctfieldselect='Cnter1,Cnter2,CnterN,'
      @ctfield (for update)=~,Cnter1,Cnter2,CnterN~
      For update form @ctfield from @ctfieldselect ie. ~,Cnter1,Cnter2,Cnter3~
*/
-- Get modified @rowcnters term
-- Concatenate (1st transform alias) and 1st column field from @fieldselect
-- this will simulate @rowruns for racsp;4 (means less changes)
select @qyrstatement1=alias from @trans# where rd=1
if charindex(',',@fieldselect)>0
select @qyrstatement2=substring(@fieldselect,1,charindex(',',@fieldselect)-1)
else
-- This will cover situation where there's only 1 pivot field
set @qyrstatement2=@fieldselect
-- We're removing parenthesis from pivot field
set @qyrstatement2=replace(replace(replace(@qyrstatement2,'-
@@@','Totals'),'[',''),'','')
set @rowcnterswork=''
--
/* old code before user option to specify counter field name.
select @rowcnterswork=@rowcnterswork+case when @rowcnterswork>' then '&' else '' end+
@qyrstatement2+'{' +pivotin+'}'+(''+@qyrstatement1+'),'
@ctfields=@ctfields+case when @ctfields>' then ',' else '' end+
'Cnter'+cast(rd as varchar(2))+ ' varchar(5)', /* no default */
@ctfieldselect=@ctfieldselect+case when @ctfieldselect>' then ',' else '' end+
'Cnter'+cast(rd as varchar(2))
*/
--
--      New code takes into account user supplied counter field name.If no name supplied
--      ( in { } ) default of 'Cnter'+field# is used ie Cnter1.
--      Also includes code to replace counter fields in @wherecnters with ctN for using
--      counter logic in update
--
set @wherecntersupdate1=@wherecnters
select
@cntername=ltrim(rtrim(case when charindex('{',pivotin)>0 and charindex('}',pivotin)>0
then
substring(pivotin,charindex('{',pivotin)+1,
((charindex('}',pivotin)-1) - (charindex('{',pivotin)))) else '' end )),
--
@rowcnterswork=@rowcnterswork+case when @rowcnterswork>' then '&' else '' end+
@qyrstatement2+'{' +rtrim(ltrim(replace(replace(replace(pivotin,'{',''),'}',''),
@cntername,''),@cntername,'{' +@cntername+'}')))+'}'
+'(' +@qyrstatement1+')',
--
@ctfields=@ctfields+case when @ctfields>' then ',' else '' end+
case when @cntername='' then 'Cnter'+cast(rd as varchar(2)) else @cntername end+'
varchar(5)', /* no default */
--

```

```

@ctfieldselect=@ctfieldselect+case when @ctfieldselect>' ' then ',' else '' end+
case when @cntrname='' then 'Cntr'+cast(rd as varchar(2)) else @cntrname end,
--      Replace counter field with cast(ctN as int)
@wherectersupdate1=case when @cntrname>' ' then
replace( @wherectersupdate1,@cntrname,'cast(@ct'+cast(rd as varchar(2))+ ' as int)' )
else
replace( @wherectersupdate1,'Cntr'+cast(rd as varchar(2)), 'cast(@ct'+cast(rd as
varchar(2))+ ' as int)')
end
from racpovitin# ( @rowcnters, '&')
order by rd
set @ctfields=@ctfields+', '
--
--   In the case of a standard (non-rotated) xtab with multiple transforms and
--   rowbreaking we want an empty string ('') to display in counter fields so
--   we add a default value to cntr fields definition. For other types of reports
--   we don't need default''. Also @numvalues>1 @rotate='y' and displayrowfunctions='m'
--
if @printrowcnters='y'
begin
if ((@numvalues>1) and (@rowbreak='y') and (@rotate='n')) or
((@numvalues>1) and (@rotate='y') and (@rowfunctions>'') and
(@displayrowfunctions='m'))
set @ctfields=replace(@ctfields, ',', 'default''', ' )
--
set @ctfield='~,'+@ctfieldselect+'~' /* for update */
set @ctfieldselect=@ctfieldselect+', '
end
else
begin
set @ctfield='' /* for update */
set @ctfields=''
set @ctfieldselect=''
end
set @rowcnters=@rowcnterswork
--
--   Check for alias(s) in case of query emulation or cutpvt='y'. Alias can only
appear
--   in @wherecters when there is only 1 pvtcol for the alias, ie query emulation or
--   cutpvt='y'. Only for numbers. Using fixed format of decimal(15,6). Note alias could
be
--   used separately or in addition to row counters.
--
end /* ends @rowcnters>' ' */
--   Check @wherecters for modification. It's possible there is no @rowcnters
--   but there is @wherecters with just @transform logic.
if (@wherecters>' ') and ( (charindex('select_query',@pvtcol)>0) or (@cutpvt='y') )
begin
if @rowcnters=''
set @wherectersupdate1=@wherecters
--   Replacing each @transform alias with valueN ie. cnt with value1.
select @wherectersupdate1=
replace(@wherectersupdate1, alias, 'cast(value'+cast(rd as varchar(2))+ ' as
decimal(15,6))')
from @trans#
order by rd
end
--
if @wherecters>' '
begin

```

```

-- Check if a row field from @grpcol is being used.If true replace rac.field with
#rac.field.
-- User inputs rac.field then we get #rac.field
set @wherectersupdate1=replace(@wherectersupdate1,'rac.','rac.')
set @wherectersupdate1='case when '+@wherectersupdate1+' then '
set @wherectersupdate2=' else ~ end '
end
--
-- Process @fieldadd if present,adding @addfield
set @qrytrans='' /* This will hold field definition of fieldadd(n) */
set @valcreate='' /* This will hold field name of fieldadd(n) */
if @fieldadd1>''
begin
set
@qrytrans=['+replace(replace(@fieldadd1,['',''],'],['',''])+']varchar('+cast(len(@fieldva
lue1) as varchar(5))+')default'+'''+@fieldvalue1+''''
set @valcreate=['+replace(replace(@fieldadd1,['',''],'],['',''])+']'
if @fieldadd2>''
begin
set
@qrytrans=@qrytrans+', ['+replace(replace(@fieldadd2,['',''],'],['',''])+']varchar('+cast(l
en(@fieldvalue2) as varchar(5))+')default'+'''+@fieldvalue2+''''
set @valcreate=@valcreate+', ['+replace(replace(@fieldadd2,['',''],'],['',''])+']'
end
if @fieldadd3>''
begin
set
@qrytrans=@qrytrans+', ['+replace(replace(@fieldadd3,['',''],'],['',''])+']varchar('+cast(l
en(@fieldvalue3) as varchar(5))+')default'+'''+@fieldvalue3+''''
set @valcreate=@valcreate+', ['+replace(replace(@fieldadd3,['',''],'],['',''])+']'
end
if @fields>''
begin
set @qrytrans=@qrytrans+', '
set @valcreate=@valcreate+', '
end
--set @fields=@qrytrans+@fields
--set @fieldselect=@valcreate+@fieldselect
end
-- Final fields/fieldselect definition
--
--@mgrpsfieldsdef
--A varchar(50),B varchar(50),C varchar(50),
--@mgrpsfieldselect
--A,B,C
--
-- Check for Rotate
--
-- @rotatefields: fields definition up to pvt fields
-- @rotatefieldspvt: fields definition for pvt fields
-- @rotatefieldselect: fields select up to pvt fields
-- @rotatefieldspvtselect: fields select for pvt fields
--
if (@rotate in ('y','nest')) or (@convert>'' )
begin
set @rotatefieldspvt=@rowfunctfield + @runsfield+@diffields+@fields
set
@rotatefieldspvtselect=@rowfunctfieldselect+@runsfieldselect+@diffieldselect+@fieldsel
ect
end

```

```

--
if @mgrps='' /* no multiple grps */
begin
--
if (@rotate in ('y','nest')) or (@convert>')
begin
set @rotatefields=@grpcolinsert varchar(@mxlengrp)default~, '+@qrytrans
-- @qrytrans has fieldadd(n) definition, if none it is ''
set @rotatefieldselect=@grpcolinsert+', '+ @valcreate
-- @valcreate has fieldadd(n) name, if none it is ''
end
--
set @fields=@grpcolinsert varchar(@mxlengrp)default~, '+ @qrytrans + @str1 +
@rowfunctfield + @runsfield + @diffields+@fields
set @fieldselect=@grpcolinsert+', '+ @valcreate + @printaggfieldselect +
@rowfunctfieldselect+@runsfieldselect+@diffieldselect+@fieldselect
end
else
begin
-- Here multiple groups ----
if (@rotate in ('y','nest')) or (@convert>')
begin
set @rotatefields=@mgrpsfieldsdef + @ctfields + @qrytrans
set @rotatefieldselect=@mgrpsfieldselect+', '+ @ctfieldselect + @valcreate
end
--
set @fields=@mgrpsfieldsdef + @ctfields + @qrytrans + @str1 + @rowfunctfield +
@runsfield + @diffields+@fields
set @fieldselect=@mgrpsfieldselect +', '+ @ctfieldselect + @valcreate +
@printaggfieldselect +
@rowfunctfieldselect+@runsfieldselect+@diffieldselect+@fieldselect
end
set @fields=replace(@fields, '@grpcolinsert', @grpcolinsert)
set @fields=replace(@fields, '@mxlengrp', @mxlengrp)
set @fields=replace(@fields, '~', char(39))
set @fields=replace(@fields, '-@@@', 'Totals')
if (@rotate in ('y','nest')) or (@convert>')
begin
set @rotatefields=replace(@rotatefields, '@grpcolinsert', @grpcolinsert)
set @rotatefields=replace(@rotatefields, '@mxlengrp', @mxlengrp)
set @rotatefields=replace(@rotatefields, '~', char(39))
set @rotatefieldspvt=replace(@rotatefieldspvt, '@grpcolinsert', @grpcolinsert)
set @rotatefieldspvt=replace(@rotatefieldspvt, '@mxlengrp', @mxlengrp)
set @rotatefieldspvt=replace(@rotatefieldspvt, '~', char(39))
set @rotatefieldspvt=replace(@rotatefieldspvt, '-@@@', 'Totals')
-- Get convert strings
-- Examples
--set @convertfields='shipcountry varchar(11)default'', '
--set @convertfieldselect='shipcountry, '
--set @convertfieldspvtselect='[Totals], [1], [2], [3]'
set @convertfieldspvtselect=@rotatefieldspvtselect
set @convertfields=@rotatefields+replace(@str1, '~', char(39)) /* adding on
'Funct', (excluded in @rotatefields) */
set @convertfieldselect=@rotatefieldselect+ @printaggfieldselect
end
--
--set @fieldselect=@grpcolinsert+', '+
@printaggfieldselect+@rowfunctfieldselect+@runsfieldselect+@fieldselect
--
--This covers all conditions to return xtabfields and tabledef

```

```

-- Store table definition amd/or select string (also want tabledef for worktable3).
if (@tabledef is not null or @worktable3>'') and (@multicell!='y')
set @tabledef='rd int, '+@fields
if @xtabfields is not null and @multicell!='y'
set @xtabfields='rd, '+@fieldselect
-- Check for immediate return (do not create xtab table/compute xtab)
if @exec='n'
begin
if isnull(len(@tabledef),0)>=7950 or isnull(len(@xtabfields),0)>=7950
raiserror('your crosstab MAY be too big to create,recommend you check specs',0,1)with
nowait
return
end
--
if @rowfunctions>' ' and @displayrowfunctions='m'
set @fieldsdef=@fields /* create table fields definition */
if @forcerows='y'
set @fieldsforce=@fields
-- Check for local temp table #xtab
if @xtab='#xtab'
set @fields='alter table #xtab add '+@fields
else
if @xtab='#multicellzz1'
set @fields='alter table #multicellzz1 add '+@fields
else
set @fields='create table '+@xtab+'(rd int, '+@fields+')'
-- Check length of create xtab table statement
if len(@fields)>=7950
begin
raiserror('your crosstab is too big',16,1)with nowait
return(-3)
end
exec(@fields)
if @@error!=0
begin
raiserror('create table %s definition error',16,1,@xtab)with nowait
return(-3)
end
-- Delete overall totals for transposing or @grand_totals=n
--if charindex(' ',@transform)=0 or (@rank is null and @grand_totals='n')
if @rank is null and @grand_totals='n'
begin
set @str1='delete from #rac where '+@grpcolinsert+'='+'@@@''
exec(@str1)
if @@error!=0
begin
raiserror('error deleting grand totals',16,1)with nowait
return(-3)
end
end
-- Store #rac in @worktable if @worktable1 is active
-- @worktable1 has partial columns of #rac
if @worktable1>' '
begin
set @str1=
'select rd,@grpcolinsert,@pvtcolinsert,@valinsert into @worktable
from #rac '
set @str1=replace(@str1, '@grpcolinsert', @grpcolinsert)
set @str1=replace(@str1, '@pvtcolinsert', @pvtcolinsert)
set @str1=replace(@str1, '@valinsert', @valinsert)

```

```

set @str1=replace(@str1, '@worktable', @worktable)
exec(@str1)
if @@error!=0
begin
raiserror('error creating worktable %s', 16, 1, @worktable) with nowait
return(-3)
end
end /* ends worktable>' ' */

-- Loop thru value(s)
-- Check for ranking
if @rank is null
-- set @rank1=@rank=null, '
set @rank1=' '
else
set @rank1=@rank=case when #rac.@pvtcolinsert =~-@@@~ then 0
else case when @grp=#rac.@grpcolinsert then @rank1+1 else 1 end end, '
--
-- Check for @cutpvt
if (@cutpvt='n') or (@cutpvt='y' and @colretain>'')
set @updatecut=' '
-- ' (select top 100 percent * from #rac order by rd ) as a, #rac
-- where a.rd=#rac.rd '
else
set @updatecut=' where #rac.@pvtcolinsert=~-@@@~ '
-- ' (select top 100 percent * from #rac order by rd ) as a, #rac
-- where a.rd=#rac.rd and #rac.@pvtcolinsert=~-@@@~ '
-- Define @startpivotflds
set @startpivotflds='insert into '+@xtab+'(rd'
--Check for printing aggregates (funct)
if @printagg='n'
begin
set @aggfields='~'
set @aggvalues='~'
end
else
begin
-- set @aggfields= '~', funct~'
set @aggfields= '~', '@translabel'+ '~'
set @aggvalues='~', ~char(39)+replace(replace(~@funct~, ~[, ~), ~]~, ~)+char(39)'
end
set @k=1
while 1=1
begin
-- Check update statistics on #rac
/* set @str2=
'use tempdb
select @table=object_name(object_id('#rac'))
select i.name as [index name],
stats_date(i.id, i.indid) as [statistics date]
from sysobjects o, sysindexes i
where o.name =@table and i.indid>0 and o.id = i.id'
execute sp_executesql @str2, N'@table varchar(250)', @table=null */
--
-- Get current value (@grpcolvalue) and transform (@funct)
--
select @grpcolvalue='value'+cast(@k as varchar(5))
select @funct=funct, @alias=alias from @trans# where rd=@k
--
-- Check if @decimal from rdecimal is needed for rowruns, rowfunctions or colruns

```



```

    set @decimal=''
--select top 1 @existest=rd from @rowsums# where rtrans=@alias runcol
--select top 1 @existest=rd from @rowfuncts# where rtrans=@alias
-- if (@rowruns>'') or (exists(select * from @rowfuncts# where rowfunct
in('sum','avg')))) or
--      (@colruns>'')
    if (charindex(@alias,@colruns)>0)
    or (ltrim(@colruns)='')
    or (exists(select * from @rowfuncts# where rtrans=@alias and rowfunct
in('sum','avg'))))
    or (exists(select * from @rowsums# where rtrans=@alias ))
    or (charindex(@alias,@diffs)>0)
begin
    execute @check=racsp;3 /* rdecimal */
@grpcolvalue=@grpcolvalue,@lengthd1=@lengthd1 output,
    @decimal=@decimal output
    if @check!=0
    begin
        raiserror('sp racsp;3 (rdecimal) error for alias %s',16,1,@alias)with nowait
        return(-3)
    end
end /* ends calling racsp;3 rdecimal */
--
--      Check if row percents are needed.
--
    if charindex(@alias,@rpercents)=0 and charindex(@alias,@rpercentstotals)=0
        and ltrim(@rpercents)!='' and ltrim(@rpercentstotals)!=''
        set @rowpercents='@rowpercent=~,'
else
    if (charindex(@alias,@rpercentstotals)>0 or ltrim(@rpercentstotals)='') and
@display='m'
    begin
        -- get total for current @grpcolvalue
        set @str2='select @rtotal='+@grpcolvalue+' from #rac
            where '+@grpcolinsert+'='''+@@@''' and '+
                @pvtcolinsert+'='''-@@@'''
        execute sp_executesql @str2,N'@rtotal float(2) output',@rtotal=@rtotal output
        if abs(@rtotal)>0
            set @rowpercents=
                '@rowpercent=case when (#rac.@grpcolinsert not in (~@@@~)) and
                (#rac.@pvtcolinsert=~@@@~)
                then
~[%~+cast(round(100*( abs(cast(#rac.@grpcolvalue as float(2)))/cast('+cast(@rtotal as
varchar(15))+' as float(2))) ),1) as varchar(7))+~]~
                    else ~ end,'
            else
                set @rowpercents='@rowpercent=~[%]~,'
            end
    else
    if (charindex(@alias,@rpercents)>0 or ltrim(@rpercents)='') and @display='m'
    begin
        -- Here we want each column/row total percents
        -- First store overall total for the column @rtotal
        set @rowpercents=
            '@rtotal=(select m.@grpcolvalue from #rac as m where
            m.@pvtcolinsert=#rac.@pvtcolinsert and m.@grpcolinsert=~@@@~),
            @rowpercent=case when (#rac.@grpcolinsert not in (~@@@~)) then case
            when abs(cast(@rtotal as float(2)))>0 then
            ~[%~+cast(round(100*( abs(cast(#rac.@grpcolvalue as float(2)))/cast(@rtotal as
float(2))) ),1) as varchar(7))+~]~

```



```

        else ~[%0]~ end else ~ end, '
end
-- Check if column percents are needed.
if charindex(@alias,@cpercents)=0 and ltrim(@cpercents)!='*'
set @colpercents = '@total=~',
@colpercent=~', '
else
if (charindex(@alias,@cpercents)>0 or ltrim(@cpercents)!='*') and @display='m'
set @colpercents=
'@total=case when @grp=#rac.@grpcolinsert then @total else
(select n.@grpcolvalue from #rac as n where
n.@grpcolinsert=#rac.@grpcolinsert and n.@pvtcolinsert=~@@@~) end,
@colpercent=case when (#rac.@pvtcolinsert=~@@@~) then ~ else
case when abs(cast(@total as float(2)))>0
then ~(%+cast(round(100*( abs(cast(#rac.@grpcolvalue as float(2)))/cast(@total as
float(2))) ),1) as varchar(7))+~)~
else ~(%0)~ end end, '
-- Check for rowfunctions
if (exists(select * from @rowfuncts# where rtrans=@alias))
and @display='m'
-- if @rowfunctions>' and @display='m'
begin
if @displayrowfunctions='s'
set @rowp=', '
else
set @rowp='^'
execute @check=racsp;7 /* rowfunct */
@rowp=@rowp,@grpcolvalue=@grpcolvalue,@maxlen=@maxlen,
@rowfunctions=@rowfunctions,@decimal=@decimal,@funct=@funct,@alias=@alias,
@rowfunctfield=@rowfunctfield output,@rowfunctqrys=@rowfunctqrys output
if @check!=0
begin
raiserror('sp racsp;7 (rowfunct) error for value %s :-(',16,1,@grpcolvalue)with
nowait
return(-3)
end
end
else
begin
set @rowfunctfield=~'
set @rowfunctqrys=~'
end
--
-- Running sums
-- if @rowruns>' and @display='m'
if (exists(select * from @rowsums# where rtrans=@alias )) and (@display='m')
begin
-- Here @flag_runs_cnters is 'runs'
set @runsfield=''
if @flag1='y'
set @lengthd1=@runsfieldsingle
execute @check=racsp;4 /* rowruns */
@grpcolvalue=@grpcolvalue,@length=@lengthd1,@allrowfields=@mgrpsfieldselect,
@flag_runs_cnters='runs',@flag1=@flag1,@rowruns=@rowruns,@decimal=@decimal,@funct=@fun
ct,
@alias=@alias,@runs=@runs output,@runsfield=@runsfield output,@runsvalue=@runsvalue
output,
@runsdeclare=@runsdeclare output,@runs0=@runs0 output
if @check!=0
begin

```

```

        raiserror('sp racsp;4 (rowruns) error,value %s',16,1,@grpcolvalue)with nowait
        return(-3)
    end

    --
    set @runsfield=replace(@runsfield,'runs',@rowrunslabel)
    --
end
else
    begin
        set @runsfield='~~~'
        set @runs=' '
    --
        set @runs=' '
        set @runsvalue='~~~'
        set @runsdeclare=' '
        set @runs0=' '
    end

    --
    -- Multiple row field counters
    --
    if @rowcnters>' ' /* get counters for 1st alias only */
    begin
        if @k=1
        begin
            -- set @ctfield=''
            -- @length (@lengthd1) refers to length of each part of counter field for a counter.
            -- For now lets hard code it to 4.Note we can set it based on some row count.
            -- We'll define flag in sp indicating rowruns or counters.For cnters set
            -- @flag_runs_cnters='counters'
            -- Not need to return @runsfield for counters
            execute @check=racsp;4 /* rowruns */
            @grpcolvalue=@grpcolvalue,@length='4',@allowfields=@mgrpsfieldselect,
            @flag_runs_cnters='counters',@flag1='',@rowruns=@rowcnters,@decimal='',@funct=@funct,
            @alias=@alias,@runs=@ct output,/* @runsfield=@ctfield output,*/ @runsvalue=@ctvalue
            output,
            @runsdeclare=@ctdeclare output,@runs0=@ct0 output
            if @check!=0
            begin
                raiserror('sp racsp;4 (rowruns) error for row counter(s)',16,1)with nowait
                return(-3)
            end

            --
            -- set @runsfield=replace(@runsfield,'runs',@rowrunslabel)
            -- Rename 'runs' stuff to cnter stuff.
            -- set @ct=replace(@ct,'@run','@ct')
            -- set @ctfield=replace(@ctfield,'runs','Cnters')
            -- set @ctdeclare=replace(@ctdeclare,'@run','@ct')
            -- if @printrowcnters!='y'
            -- set @ctvalue='~~~'
            -- set @ctvalue=replace(@ctvalue,'@run','@ct')
            -- set @ct0=replace(@ct0,'@run','@ct')
            -- correctly returned from proc now
            -- has already by computed previously
            -- correctly returned from proc now
        end
    else
        if (@rowbreak='y') and (@k>1) and (@wherecnters=' ')
        begin
            set @ctfield='~~~'
            set @ct=' '
            set @ctvalue='~~~'
            set @ctdeclare=' '
            set @ct0=' '
        end
    end
end

```

```

else
  if (@rowbreak='y') and (@k>1) and (@wherecnters>' ')
    begin
      set @ctfield='~~'
      -- set @ct=' '
      set @ctvalue='~~'
      -- set @ctdeclare=''
      -- set @ct0=''
    end
  end /* end rowcnters>' ' */
  else
    begin
      -- Here no row counters
      set @ctfield='~~'
      set @ct=' '
      set @ctvalue='~~'
      set @ctdeclare=''
      set @ct0=''
    end
  end
--
-- Differences
--
  if (charindex(@alias,@diffs)>0) and (@display='m')
    begin
      -- Here @flag_runs_cnters is 'diff' @length has default now in sp,
      execute @check=racsp;4 /* rowruns */
      @grpcolvalue=@grpcolvalue,@allrowfields=@mgrpsfieldselect,
      @flag_runs_cnters='diff',@flag1=@flag1,@rowruns=@diffs,@decimal=@decimal,@funct=@funct
    ,@alias=@alias,@runs=@df output,@diffonly=@diffonly output
      if @check!=0
        begin
          raiserror('sp racsp;4 (rowruns) error,value %s',16,1,@grpcolvalue)with nowait
          return(-3)
        end
      -- Setting following values here instead of in sp rowruns
      set @difffield='~, [Diff]~' /* for update */
      set @diffvalue='~,~+char(39)+@diff1+char(39) ' /* for update */
      -- set @diffdeclare='@diff1 varchar(15),@lastd1 decimal(12,2),' /* for
update */
      set @diffdeclare='@diff1 varchar(15),@lastd1 '+@decimal+', ' /* for
update */
      set @diff0='@diff1=~0~,@lastd1=~0~, ' /* for update */
      --
    end
  else
    begin
      set @difffield='~~'
      set @diffonly=' '
      set @df=' '
      set @diffvalue='~~'
      set @diffdeclare=''
      set @diff0=''
    end
  end
--
-- Check for column running sums.
if @colruns>' ' and @display='m'
  begin
    -- if exists(select * from #cruns where run=@alias)
    if charindex(@alias,@colruns)>0 or ltrim(@colruns)='*'

```

```

begin
/*      execute @check=racsp;3 @grpcolvalue=@grpcolvalue,@decimal=@decimal output
if @check!=0
begin
raiserror('sp racsp;3 (rdecimal) error for colruns alias %',16,1,@alias)with
nowait
return(-3)
end
*/
set @cruns='@cruns=case when @grp=#rac.@grpcolinsert then
case when (#rac.@pvtcolinsert not in (~@000~))
and (#rac.@grpcolvalue is not null) then cast(cast(@cruns as '+@decimal+') +
cast(#rac.@grpcolvalue as '+@decimal+') as varchar(20)) else @cruns end
else case when (#rac.@pvtcolinsert not in (~@000~))
and #rac.@grpcolvalue is not null then cast(cast(#rac.@grpcolvalue as '+@decimal+') as
varchar(20))
else cast(0 as varchar(20)) end end,'
set @crunsvalue='case when #rac.@pvtcolinsert not in (~@000~)
then ~/~@cruns else ~ end'
end
else
begin
--set @cruns='@cruns=null,'
set @cruns=' '
set @crunsvalue='~'
end
end
else
begin
--set @cruns='@cruns=null,'
set @cruns=' '
set @crunsvalue='~'
end
end
-- Check for retaining/concatenating column values.
if @colretain>' ' and @display='m'
begin
if charindex(@alias,@colretain)>0 or ltrim(@colretain)='*'
begin
if @colretainpvt='n'
set @retain=
'@retain=case when @grp=#rac.@grpcolinsert then
case when (#rac.@grpcolinsert not in (~@000~))
and (#rac.@pvtcolinsert not in (~@000~))
and (#rac.@grpcolvalue is not null) then
@retain + case when @retain~ then ~@separator~ else ~ end
+ #rac.@grpcolvalue else @retain end
else case when (#rac.@pvtcolinsert not in (~@000~))
and (#rac.@grpcolvalue is not null)
and (#rac.@grpcolinsert not in (~@000~))
then #rac.@grpcolvalue
else ~ end end,'
else
-- 10/10/00 added ', ' to #rac.@pvtcolinsert in first substring below,also to @retain.
set @retain=
'@retain=case when @grp=#rac.@grpcolinsert then
case when #rac.@grpcolvalue is not null
then @retain +
case when
charindex(substring(~,~#rac.@pvtcolinsert,1,charindex(~^~,~,~#rac.@pvtcolinsert)),~,
~@retain)=0 then

```

```

case when @retain>~ then ~,~ else ~ end + #rac.@grpcolvalue else
~!~+substring(#rac.@grpcolvalue,charindex(~^~,#rac.@grpcolvalue)+1,len(#rac.@grpcolval
ue)) end
else @retain end
else case when #rac.@grpcolvalue is not null
then #rac.@grpcolvalue else ~ end end,'
        set @retainfld=~,'+@all_cols+~'
        set @retainvalall=~~,~+char(39)+@retain+char(39)'
    end
    else
    begin
        -- set @retain=@retain=null,'
        set @retain=' '
        set @retainfld=~~'
        set @retainvalall=~~'
    end
end
else
begin
    --set @retain=@retain=null,'
    set @retain=' '
    set @retainfld=~~'
    set @retainvalall=~~'
end
end
if @printagg='y'
begin
    if @display='s'
    set @func=@printaggs /* here display='s' */
    end
    else
    set @func=''
-- @str2 using sp_executesql taken out in favor of exec(@str1)
-- Set rowbreak variable
if @rowbreak='y'
    set @rbreak='1'
    else
    set @rbreak='2'
-- Check blocking type
if @blocktype='seq' /* sequence,row1+funcnt1+funcnt2+funcntn..row2+funcnt1+funcntn*/
begin
    set @rdstart='null'
    set @block=' isnull(@rd+@numvalues,@k) '
    end
    else
    /* stack,rows+funcnt1+repeat rows+funcnt2+repeat rows+funcnt3..*/
    begin
        if @k=1
        set @rdstart='0'
        else
        begin
            set @str2='select @rdstart=max(rd) from '+@xtab
            execute @check=sp_executesql @str2,N'@rdstart int output',
            @rdstart=@rdstart output
            if @check!=0
            begin
                raiserror('error obtaining max(rd) from #rac for ''stack'',16,1)with nowait
                return(-2)
            end
            end
            set @block=' @rd+1 '
        end
    end

```

```

--
set @str1=
'update #rac
set
@cnt=#rac.key1=case when @grp=#rac.@grpcolinsert then @cnt else @cnt+1 end,
@rd=case when @grp=#rac.@grpcolinsert then @rd
      else @block end,
@rank1?
@commonpivotflds=
~,~+[-+ case when #rac.@pvtcolinsert=~-@@@~
      then ~Totals~ else #rac.@pvtcolinsert end + ~]~ ,
@rowpercents?
@colpercents?
@runs?
@diffs?
@ct?
@cruns?
@retain?
@commonvalues=
case when #rac.@grpcolvalue is not null then
~,~char(39)+#rac.@grpcolvalue+@crunsvalue+@colpercent+@rowpercent+char(39)
else ~,~char(39)+~@nullcell~ + char(39) end ,
@pivotflds=case when @grp=#rac.@grpcolinsert then @pivotflds + ?commonpivotflds
      else @startpivotflds + case when @k=1 then
~,~ + @u5 else
case when @rbreak=1 then ~ else ~,~ + @u3 end
end +
@ctfield + @aggfields + @rowfunctfield + @runsfield + @difffield + @commonpivotflds
end,
@values=case when @grp=#rac.@grpcolinsert then @values + ?commonvalues
      else @startvalues + cast(@rd as varchar(10)) +
      case when @k=1 then ~,~ +
      case when #rac.@grpcolinsert=~-@@@~ then char(39)+Totals+char(39) @break?
      else @u6 end
      else
      case when @rbreak=1 then ~ else
~,~ + case when #rac.@grpcolinsert=~-@@@~ then char(39)+Totals+char(39)
else @u4 end end
end + @ctvalue +
@aggvalues + @rowfunctqrys + @runsvalue + @diffvalue
+ @commonvalues end,
#rac.col= @wherecntersupdate1 @pivotflds + @retainfld + ~)~ + @values + @retainvalall
+ ~)~ @wherecntersupdate2,
@grp=#rac.@grpcolinsert @u2 @diffonly
@updatecut '
--
if (@rank is not null) and (@ranklimit>'0')
begin
set @str1=replace(@str1,'?commonpivotflds',
'case when @rank1<='+@ranklimit+' then ?commonpivotflds else ~ end ')
set @str1=replace(@str1,'?commonvalues',
'case when @rank1<='+@ranklimit+' then ?commonvalues else ~ end ')
end
--
if (@cutpvt='n') or (@cutpvt='y' and @colretain='')
begin
set @str1=replace(@str1,'?commonpivotflds','@commonpivotflds' )
set @str1=replace(@str1,'?commonvalues','@commonvalues' )
end
else

```

```

begin
-- Here @cutpvt='y' and @colretain>' so we want to accumulate totals
-- and [all_cols] but not individual cols.
if @cutpvt='y' and @colretain>' and @row_totals='y'
begin
set @str1=replace(@str1,'?commonpivotflds','~')
set @str1=replace(@str1,'?commonvalues','~')
end
else
if @cutpvt='y' and @colretain>' and @row_totals='n'
begin
set @str1=replace(@str1,'?commonpivotflds','~')
set @str1=replace(@str1,'?commonvalues','~')
set @str1=replace(@str1,'+ @commonpivotflds','+ ~')
set @str1=replace(@str1,'+ @commonvalues','+ ~')
end
end
if @rank is not null
set @str1=replace(@str1,'else #rac.@pvtcolinsert',
'else ~'+@rank+'~'+cast(@rank1 as varchar(4)))
--
-- Check for multiple groups
--
if @mgrps>'
begin
if @rowbreak='y'
begin
set @str1=replace(@str1,'@u5',@mgrpsupdate5)
set @str1=replace(@str1,'@u6',@mgrpsupdate6)
set @str1=replace(@str1,'@u2',''+@mgrpsupdate2)
end
else
begin
-- No rowbreak, but have to check for rowruns.rowruns needs @f1,@f2 etc stuff
if @rowruns='' and @rowcnters=''
begin
set @str1=replace(@str1,'@u5',@mgrpsupdate3)
set @str1=replace(@str1,'@u6',@mgrpsupdate4)
set @str1=replace(@str1,'@u2',' ')
set @mgrpsupdate1=''
set @mgrpsupdateinit=''
end
else
begin
-- Here rowbreak=n but there are rowruns need @mgrpsupdate1,2 and init
set @str1=replace(@str1,'@u5',@mgrpsupdate3)
set @str1=replace(@str1,'@u6',@mgrpsupdate4)
set @str1=replace(@str1,'@u2',''+@mgrpsupdate2)
-- @mgrpsupdate1 and @mgrpsupdateinit stay defined
end
end
set @str1=replace(@str1,'@u3',@mgrpsupdate3)
set @str1=replace(@str1,'@u4',@mgrpsupdate4)
--set @str1=replace(@str1,'@char(39)','~')
--set @str1=replace(@str1,'@u2',''+@mgrpsupdate2)
end /* ends multiple groups */
else
begin
-- Here no multiple groups
set @str1=replace(@str1,'@u5','~@grpcolinsert~')

```

```

set @str1=replace(@str1,'@u3','~@grpcolinsert~')
set @str1=replace(@str1,'@u6',' char(39)+#rac.@grpcolinsert+char(39) ')
set @str1=replace(@str1,'@u4',' char(39)+#rac.@grpcolinsert+char(39) ')
set @str1=replace(@str1,'@u2',' ')
--set @str1=replace(@str1,'@char(39)','char(39)')
set @mgrpsupdate1=' '
end /* end no multiple groups */
--
set @str1=replace(@str1,'@wherecntersupdate1',@wherecntersupdate1)
set @str1=replace(@str1,'@wherecntersupdate2',@wherecntersupdate2)
set @str1=replace(@str1,'@break?',@mgrpsupdate4breakvalues)
set @str1=replace(@str1,'@rank1?',@rank1)
set @str1=replace(@str1,'@block',@block)
set @str1=replace(@str1,'@aggfields',@aggfields)
set @str1=replace(@str1,'@aggvalues',@aggvalues)
set @str1=replace(@str1,'@runs?',@runs)
set @str1=replace(@str1,'@diffs?',@df)
set @str1=replace(@str1,'@diffonly',@diffonly)
set @str1=replace(@str1,'@ct?',@ct)
set @str1=replace(@str1,'@runsfield',@runsfield)
set @str1=replace(@str1,'@ctfield',@ctfield)
set @str1=replace(@str1,'@runsvalue',@runsvalue)
set @str1=replace(@str1,'@ctvalue',@ctvalue)
set @str1=replace(@str1,'@difffield',@difffield)
set @str1=replace(@str1,'@diffvalue',@diffvalue)
set @str1=replace(@str1,'@cruns?',@cruns)
set @str1=replace(@str1,'@crunsvalue',@crunsvalue)
set @str1=replace(@str1,'@retain?',@retain)
set @str1=replace(@str1,'@separator',@separator)
set @str1=replace(@str1,'@retainfld',@retainfld)
set @str1=replace(@str1,'@retainvalall',@retainvalall)
set @str1=replace(@str1,'@funct',@alias)
set @str1=replace(@str1,'@rowpercents?',@rowpercents)
set @str1=replace(@str1,'@colpercents?',@colpercents)
set @str1=replace(@str1,'@rowfunctfield',@rowfunctfield)
set @str1=replace(@str1,'@rowfunctqrys',@rowfunctqrys)
set @str1=replace(@str1,'@updatecut',@updatecut)
set @str1=replace(@str1,'@pvtcolinsert',@pvtcolinsert)
set @str1=replace(@str1,'@grpcolinsert',@grpcolinsert)
set @str1=replace(@str1,'@grpcolvalue',@grpcolvalue)
set @str1=replace(@str1,'@maxlen',@maxlen)
set @str1=replace(@str1,'@nullcell',@nullcell)
set @str1=
'declare
@grp varchar('+@mxlengrp+'),' +@mgrpsupdate1+
'@commonpivotflds varchar(100),@commonvalues varchar(1000),@pivotflds varchar(8000),
@startpivotflds varchar(100),@values varchar(8000),@startvalues varchar(10),@cnt int,
@rd int,@numvalues int,@k int,@rtotal varchar(15),@total varchar(15),@colpercent
varchar(10),
@rowpercent varchar(10),'+@ctdeclare+@runsdeclare+@diffdeclare+'@cruns
varchar(20),@retain varchar(7000),@rbreak int,@rank1 int
select @grp=~,@commonpivotflds=~,@commonvalues=~,@pivotflds=~, '
+ @mgrpsupdateinit+' @startpivotflds=~insert into '+@xtab+'(rd~'+',@values=~,
@numvalues=~values(~,@cnt=0,@rank1=0,@rd='+@rdstart+',
@numvalues='+cast(@numvalues as varchar(4))+',@total=~,@retain=~,@rbreak='+@rbreak+
',@colpercent=~,@k='+cast(@k as varchar(4))+','+@ct0+@runs0+@diff0+'@cruns=~0~ ' +
@str1
set @str1=replace(@str1,'~',char(39))
if len(@str1)>=7975
begin

```



```

--                                     Insert records in @xtab
--     Get parameters (@limit1,@limit2) for records to insert.
--     Racsp;19 gets above values,if no @limit (@limit='') all records are inserted.
--     In this case @limit1=1 and @limit2=max(key1) from #rac.
--
execute @check=racsp;19 /* limit */ @limit=@limit,@limit1=@limit1
output,@limit2=@limit2 output
if @check!=0
begin
    if @check!=-44
        raiserror('error for record return parameters for @limit='%s'' from racsp;19 (sp
limit)',16,1,@limit)with nowait
        return(-3)
    end
-- Check for blasting.
--
if @burst='y' /* using bursting here (racsp;13) */
begin
    exec @check=racsp;13
    @burstlen=@burstlen,@limit1=@limit1,@limit2=@limit2,@wherecnters=@wherecnters
    if @check!=0
        begin
            raiserror('error in sp racsp;13 (burst)',16,1)with nowait
            return(-3)
        end
    end /* ends burst='y' */
else
begin
    --'set @rd=(select max(key1) from #rac)
    set @str2=
    'while @key1<=@limit2
    begin
    select @fields=col
    from
    (select max(rd) as maxrd from #rac where key1=@key1) as a,
    #rac as b
    where maxrd=b.rd
    exec(@fields)
    @worktable3?
    set @key1=@key1+1
    end '
    if @worktable3>'
    set @str2=replace(@str2,'@worktable3?','insert '+@worktable3+'(store_xtab)
values(@fields)')
    else
    set @str2=replace(@str2,'@worktable3?',' ')
    --execute @check=sp_executesql @str2,N'@key1 int,@rd int,@fields varchar(8000)',
    --@key1=1,@rd=@rd,@fields=''
    execute @check=sp_executesql @str2,N'@key1 int,@limit2 int,@fields varchar(8000)',
    @key1=@limit1,@limit2=@limit2,@fields=''
    if @check!=0
        begin
            raiserror('insert error for %s xtab/worktable3',16,1,@xtab)with nowait
            return(-3)
        end
    end /* ends no burst */
-- Check to continue looping
--xx:
set @k=@k+1
if @k>@numvalues

```

```

        break
    if @display='m'
        begin
            -- Index on key1:drop it then recreate it
            drop index #rac.grpcol_key1
        end
    end /* end of update loop */
    -- Store #rac in @worktable if @worktable2 is active
    -- @worktable2 has all columns of #rac
    if @worktable2>' '
        begin
            set @str1=
            'select * into @worktable from #rac '
            set @str1=replace(@str1, '@worktable', @worktable)
            exec(@str1)
            if @@error!=0
                begin
                    raiserror('error creating worktable %s',16,1,@worktable)with nowait
                    return(-3)
                end
            end /* ends worktable2>' ' */
            --
            -- Insert rows from @from excluded by @force
            --
            if @forcerows='y'
                begin
                    execute @check=racsp;5 /* forcerows */ @grpcol=@grpcol,@grpsorttype=@grpsorttype,
                    @grpcolinsert=@grpcolinsert,@grpfldtype=@grpfldtype,@grptable=@grptable,
                    @functionlen=@functionlen,@fields=@fieldselect,@fieldsdef=@fieldsforce,
                    @aliasck=@aliasck,@numvalues=@numvalues,@grpart=@grpart,@xtab=@xtab,@aliastable=@alias
                    table,
                    @grandtotalsposition=@grandtotalsposition,@datelen=@datelen,@style=@style
                    if @check!=0
                        begin
                            raiserror('sp racsp;5 (forcerows) error',16,1)with nowait
                            return(-3)
                        end
                    end /* end of @forcerows='y' */
                    --
                    -- Call displayrowfunctions to break on rowfunctions
                    --
                    if @rowfunctions>' ' and @displayrowfunctions='m'
                        begin
                            set @fieldsempy=''
                            if @emptycell>' '
                                begin
                                    set @check=len(@fieldspvt)-len(replace(@fieldspvt, ',', ''))
                                    while @check>=0
                                        begin
                                            set @fieldsempy=@fieldsempy+case when @fieldsempy>' ' then ',' else '' end
                                            +','
                                        end
                                    --',' ',' ',' ',' ',' ',' ',' ',' ',' ' (this is what fieldsempy looks like)
                                    set @check=@check-1
                                end
                            end
                            else
                                -- set @fieldspvt='' for emptycell='' (fieldspvt was set=pvt fields in @str2
                                executesql
                                set @fieldspvt=''
                            end
                        end
                    end
                end
            end

```

```

--      Perform check for rotate. For rotate the number of transforms should be same
--      for number of transforms in rowfunctions. Also the number of rowfunctions for
--      each transform should be same ie. 2 for all transforms, 3 for all transforms
etc.
--      This check is performed right after populating rowfunctions table.
--
execute @check=racsp;8 /* displayrowfunctions */ @xtab=@xtab,@translabel=@translabel,
@rotate=@rotate,@fields=@fieldselect,@fieldsdef=@fieldsdef,@rowfunctlen=@rowfunctlen,
@rowfunctlen1=@rowfunctlen1,@fieldspvt=@fieldspvt,@fieldsemt=@fieldsemt
if @check!=0
begin
    raiserror('sp racsp;8 (displayrowfunctions) error',16,1)with nowait
    return(-3)
end
end
--
--      Get rotated xtab
--
if @rotate='y' or @rotate='nest'
begin
    execute @check=racsp;17 /* rotate */
    @rotate=@rotate,@transform=@transform,@translabel=@translabel,@pformat=@pformat,
    @rowfmax=@rowfmax,@rotatefieldselect=@rotatefieldselect,@rotatefieldspvtselect=@rotate
    fieldspvtselect,
    @rotatefields=@rotatefields,@rotatefieldspvt=@rotatefieldspvt,@rotatextab=@rotatextab
    output,
    @rotatequery=@rotatequery output,@fieldselect=@fieldselect output,
    @xtabfields=@xtabfields output,@tabledef=@tabledef output
    if @check!=0
    begin
        raiserror('sp racsp;17 (rotate) error',16,1)with nowait
        return(-3)
    end
    set @xtab=@rotatextab
end
end
--
--      Format @xtab based on exact max length of each field
--      Additionally modify @tabledef if it is being returned
--
if @pformat='y'
begin
    --      Populate @tabledef# if it is to be returned
    if @tabledef is not null
    begin
        set @charx=','
        set @tabledef = ltrim(@tabledef) + @charx
        while len(@tabledef)>0
        begin
            insert @tabledef#
            select rtrim(substring (@tabledef,1,charindex(@charx,@tabledef)-1))
            select @tabledef= ltrim(right(@tabledef, len(@tabledef)-charindex(@charx,@tabledef)))
        end
    end
end
--
--      Loop thru each field (after rd)
set @charx=','
set @valcreate=@fieldselect
set @valcreate = ltrim(@valcreate) + @charx
set @test=2
while len(@valcreate)>0
begin
    select @qrytrans=rtrim(substring (@valcreate,1,charindex(@charx,@valcreate)-1))

```

```

select @valcreate= ltrim(right(@valcreate, len(@valcreate)-
charindex(@charx,@valcreate)))
-- Find max for each column
set @str2=
'select @max=max(len('+@qrytrans+')) from '+@xtab
execute @check=sp_executesql @str2,N'@max int output',@max=@maxlen output
if @check!=0
begin
raiserror('pformat error max length column: %s',16,1,@qrytrans)with nowait
return(-3)
end
-- If default is '' and no data in column maxlen will be 0 (illegal),make it '1'
if @maxlen='0'
set @maxlen='1'
set @str2=
'alter table '+@xtab+' alter column '+@qrytrans+' varchar('+@maxlen+')'
execute @check=sp_executesql @str2
if @check!=0
begin
raiserror('pformat error alter table column: %s',16,1,@qrytrans)with nowait
return(-3)
end
if @tabledef is not null
update @tabledef#
set field=replace(field,substring(field,charindex('varchar(',field)+7,
charindex(')',field,charindex('varchar(',field)+8)-
(charindex('varchar(',field)+6)),
'('+@maxlen+')')
where rd=@test
set @test=@test+1
end /* ends while(valcreate) */
-- Rebuild @tabledef with new varchar length if it is to be returned
if @tabledef is not null
begin
set @tabledef=''
select @tabledef=@tabledef+case when @tabledef>' ' then ',' else '' end +field
from @tabledef#
order by rd
end
-- If worktable3 is active modify create table statement to reflect change.
if @worktable3>' '
begin
-- set @str1=~create table
@xtab(+-+replace(@tabledef,char(39),'~')+~+-~)~'
-- set @str2='update '+@worktable3+' set store_xtab='+@str1+' where rd=2 '
set @str1= 'create table '+@xtab+'('+@tabledef+')'
set @str2='update '+@worktable3+' set store_xtab=@str1 where rd=2 '
execute sp_executesql @str2,N'@str1 varchar(8000)',@str1=@str1
end
end /* end pformat='y' */
--
-- Modify length of worktable3 field store_xtab if active
if @worktable3>' '
begin
set @str2=
'select @max=max(len(store_xtab)) from '+ @worktable3
execute @check=sp_executesql @str2,N'@max int output',@max=@maxlen output
if @check!=0
begin

```

```

        raiserror('error modifying length of store_xtab field in worktable3:
%s',16,1,@worktable3)with nowait
        return(-3)
    end
    set @str2=
    'alter table '+@worktable3+' alter column store_xtab varchar('+@maxlen+')'
    execute @check=sp_executesql @str2
    if @check!=0
    begin
        raiserror('alter worktable3: %s error for store_xtab field',16,1,@worktable3)with
nowait
        return(-3)
    end
end
--
--      Check for convert
--
if @convert>' '
begin
/*      Not using this.All results from rotate select from table.
        Take care of special case of a rotated select_query and no save table
        in which case @rotatequery will be modified
if charindex('select_query',@pvtcol)>0 and @convert>' ' and @pformat='n' and
@rotate='y'
and @converttable=''
begin
select
@rotatequery=replace(@rotatequery,pivotin+'.[dummyz1a]', 'cast('+pivotin+'.[dummyz1a]
as '+@convert+')')
from racpovitin#(replace(@fieldselect,@rotatefieldselect,''),default)
order by rd
print @x
*/
--      Check for rotate and make adjustments for rotate fields
if @rotate in ('y','nest')
begin
-- Get select fields for rotate (non-pvt fields)
set @convertfieldselect=@rotatefieldselect
-- Get new pvtselect rotate fields
set @convertfieldspvtselect=replace(@fieldselect,@convertfieldselect,'')
end
execute @check=racsp;20 /*converttype */ @convert=@convert,@xtab=@xtab output,
@converttable=@converttable,@convertfieldselect=@convertfieldselect,
@convertfieldspvtselect=@convertfieldspvtselect,@convertquery=@convertquery output
if @check!=0
begin
    if @check!=-44
        raiserror('error in racsp;20 (converttype) for datatype %s',16,1,@convert)with nowait
        return(-3)
    end
end
--
--      Check for returning xtab and check for returning rd (@tablecater).
--
if @return='y'
begin
    set @checktable=len(@xtab)-len(replace(@xtab,'#',''))
    if @tablecater='y'
        set @fieldselect='rd,'+@fieldselect
    --if (@convert>' ') and (@xtab='#xtab')

```

```

if (@convert>'') and (@checktable=1)
-- set @str1=@convertquery
select @str1=case when @tablecater='y' then replace(@convertquery,'select','select
rd,') else @convertquery end
-- else
-- if (@rotate in ('y','nest')) and (@rotatetab='#')
-- set @str1=@rotatequery
else
set @str1='select '+@fieldselect+' from '+@xtab+' order by rd'
exec(@str1)
if @@error!=0
begin
raiserror('error selecting from @xtab table',16,1)with nowait
return(-3)
end
end
--Multicell report is special case for returning xtabfields/tabledef.Because of
recursive call
--the recursive run (2nd) must store its xtabfields/tabledef in global cursors.
--Below,if true,stores the xtabfields/tabledef from the recursive run (2nd) so fields
can be
--returned by first run output field (if @xtabfields/@tabledef is not null) when 1st
run
--reenters RAC (at end).
if @multicellrun2='y' and @multicell='n'
begin
if @xtabfields is not null
declare xfieldszz1 cursor global for select @xtabfields as xfields
if @tabledef is not null
declare tabdefzz1 cursor global for select @tabledef as tfields
end
--
-- Check for multicell
--
if @multicell='y'
begin
-- Prepare for recursive call to RAC to process table #multicell
-- str1 holds the @from for the recursive call which parses #multicell table
-- The columns/concatenated string has blanks right trimmed only.Leading blanks
-- will NOT be trimmed.
set @char1='~,~'
set @str1=
'(select @grpcol, substring(expr,1,charindex(~~,expr)-1) as [column],
replace(substring(expr,charindex(~~,expr)+1,len(expr)),~!,~@separator~) as cellvalue
from
(select @grpcol,
rtrim(substring(@char1+all_cols+@char1,pos+1,
charindex(@char1,@char1+all_cols+@char1,pos+1)-
(charindex(@char1,@char1+all_cols+@char1,pos)+1)))
as expr
from @inputable,racforparsestring#()
where pos=charindex(@char1,@char1+all_cols+@char1,pos)
and pos<len(@char1+all_cols+@char1) ) as a) as b'
-- To insure 2nd run references correct row field we're using @grpcolinsert.If 1st run
uses
-- expression for @grpcol,@grpcolinsert will correctly reference it while just @grpcol
will
-- not.If 1st run uses only a field for row then @grpcol and @grpcolinsert are the
same.Get it?
set @grpcol=@grpcolinsert

```

```

set @str1=replace(@str1,'@grpcol',@grpcol)
set @str1=replace(@str1,'@inputable','#multicellzz1')
set @str1=replace(@str1,'@char1',@char1)
set @str1=replace(@str1,'@separator',@separator)
set @str1=replace(@str1,'~',char(39))
-- Need digits table for recursive call
-- Table #forparsestring replaced by function racforparsestring#().
--
--
-- Make recursive call to RAC
--
-- @multicell='n' so recursive call does not execute code for @multicell='y'
-- @transform and @pvtcol are based on fields created from @from (@str1 above)
-- @xtab is set to @multicelltable which is @xtab for 1st RAC run. Table ##multicell is
-- always xtab table on 1st run and @xtab is used as output table in recursive (2nd
run) call.
-- Other option which are hardcoded below are needed for proper output. For @return and
-- @printqry, turning them on in 1st run leaves them on in second.
-- Recursive notes:
-- Recursive call that creates a #table generates a different #table with same name. Non
-- dynamically selecting from a #table seems to show correct data but column labels
are
-- from first #table. Selecting dynamically shows correct column names. There also seems
to be
-- problem with @where on recursive call. When fields are supplied or @where is
generated
-- (passing in no @where or @where='') RAC crashes. I think this is bug due to
substring in a
-- derived table. Workaround is setting @where to constant (1=1). If a ##table is used
instead
-- of the derived query the @where works ok. All filtering logic should therefore be
done
-- in first RAC run as no where is applied in recursive call.
-- When RAC is called recursively (2nd RAC run below) the active parameter values are
now
-- those defined below and the recursive call runs thru and completes the entire RAC
-- procedure. But the 1st run is STILL ACTIVE and comes back to RAC with all the values
from
-- the first run. The place it comes back is the 'if @check!=0' statement below and
then
-- RAC is existed. At that point all #tables are dropped. In other words #tables from
1st
-- would still exist for recursive run if not explicitly dropped. Also note that when
adding
-- procedure to a db there is warning message that it can find RAC. This is because it
finds
-- the recursive call to RAC and RAC has not yet added to system tables.
-- For dates use same @datelen and @style values. 2nd RAC run effectively ignores
@grpsortype
-- in favor of subselect query in order by for @grpcol. Only sorting parameters for
@pvtcol
-- is relevant for 2nd run.
exec @check=rac @transform='max(cellvalue) as cellvalue',@grpcol=@grpcol,
@pvtcol='[column]',@printagg='n',@emptycell=@emptycell,
@xtab=@multicelltable, @where='1=1', @return=@returnmulticell,@grpsortype=@grpsortype,
@pvtstype=@pvtstype,@getmxlenagg='y',@from=@str1,@multicell='n',@force=@multicellf
orce,
@row_totals='n',@grand_totals='n',@printqry=@printqry,@xtabfields=@xtabfields,@forcete
st='n',
@grpsortnum='n',@pvtstype=@pvtstype,@tabledef=@tabledef,@worktable1=@multiwor
ktable1,

```



```

@worktable2=@multiworktable2,@worktable3=@multiworktable3,@datelen=@datelen,@style=@style,
@fieldadd1=@multifieldadd1,@fieldvalue1=@multifieldvalue1,@fieldadd2=@multifieldadd2,
@fieldvalue2=@multifieldvalue2,@fieldadd3=@multifieldadd3,@fieldvalue3=@multifieldvalue3,
@racheck=@racheck,@multicellrun2='y',@burst=@burst,@burstlen=@burstlen,
@forcerange=@multicellforcerange,@tablecater=@tablecater
-- Here is where 1st multicell returns after 2nd recursive run.Return value in @check
(from
-- 2nd run) is being checked by 1st run.Get it?
if @check!=0
begin
raiserror('error for 2nd rac (recursive) multicell report run',16,1)with nowait
return(-3)
end
end /* end multicell='y' */
-- Get xtabfields/tabledef for recursive run.Here is 1st run returning and we test for
1st run
-- value of 'y' for @multicell.
if @multicell='y'
begin
if @xtabfields is not null
begin
open xfieldszz1
fetch xfieldszz1 into @xtabfields
close xfieldszz1
deallocate xfieldszz1
end
if @tabledef is not null
begin
open tabdefzz1
fetch tabdefzz1 into @tabledef
close tabdefzz1
deallocate tabdefzz1
end
end
set nocount off
return
-- Start of inserting into table variables
-- Insert into @trans#.Idea is to use table variable local to procedure
-- and if table has to be passed to another procedure pass it as a function.
-- Using table variable in procedure should cut down on contention of repeatedly
-- call function.
trans_table:
set @str=@transform
set @charx='&'
set @str = ltrim(@str) + @charx
while len(@str)>0
begin
insert @trans#
select rtrim(substring (@str,1,charindex(@charx,@str)-1))
select @str= ltrim(right(@str,len(@str)-charindex(@charx,@str)))
end
goto resume1
--
rowfunctions_table:
-- rowfuncterm is rowfunction term ie. count(sum(freight))
-- rowfunct is count
-- rtrans is #trans(funcnt) term ie. sum(freight) rowfuncterm is rowfunction term ie.
count(sum(freight))

```

```

set @str = @rowfunctions
set @charx='&'
set @str = ltrim(@str) + @charx
while len(@str)>0
begin
insert @rowfuncts#
select rtrim(substring (@str,1,charindex(@charx,@str)-1))
select @str= ltrim(right(@str, len(@str)-charindex(@charx,@str)))
end
goto resume2
--
rowruns_table:
-- Same structure as @rowfuncts table
set @str = @rowruns
set @charx='&'
set @str = ltrim(@str) + @charx
while len(@str)>0
begin
insert @rowsums#
select rtrim(substring (@str,1,charindex(@charx,@str)-1))
select @str= ltrim(right(@str, len(@str)-charindex(@charx,@str)))
end
goto resume3
--
user_table:
-- User supplied code snippets
set @str1=@str
set @charx='^'
set @str = ltrim(@str) + @charx
while len(@str)>0
begin
insert @user#
select rtrim(substring (@str,1,charindex(@charx,@str)-1))
select @str= ltrim(right(@str, len(@str)-charindex(@charx,@str)))
end
set @str=@str1
goto resume4
--
aliasfillin:
-- Check if @from has brackets on any tables ie [order details] and if
-- yes fill spaces with '@' character ie [order@details]
set @check=len(@aliastable)
set @k=1
while @k<=@check
begin
if substring(@aliastable,@k,1)='['
begin
while substring(@aliastable,@k,1)!=']'
begin
set @k=@k+1
if substring(@aliastable,@k,1)= ' '
set @aliastable=stuff(@aliastable, @k, 1, '@')
end
end
set @k=@k+1
end
goto resume5
--
multigrpcols:
-- Fill table @groupstable# with all grpcol (row) fields

```

```

set @str = @grpcol
set @charx='&'
set @str = ltrim(@str) + @charx
while len(@str)>0
begin
insert @groupstable#
select rtrim(substring (@str,1,charindex(@charx,@str)-1))
select @str= ltrim(right(@str, len(@str)-charindex(@charx,@str)))
end
-- Store number of multi row fields
select @k=max(rd) from @groupstable#
--Fill in @groupstablesort# if present
if @grpsortsub>' '
begin
set @str = @grpsortsub
set @charx='&'
set @str = ltrim(@str) + @charx
while len(@str)>0
begin
insert @groupstablesort#
select rtrim(substring (@str,1,charindex(@charx,@str)-1))
select @str= ltrim(right(@str, len(@str)-charindex(@charx,@str)))
end
-- Check that there is same number of grpsortsub records as in @grpcol
if @k!=(select max(rd) from @groupstablesort#)
begin
raiserror('number of multi row sort fields must equal number of multi row
fields',0,1)with nowait
return(-3)
end
end /* end grpsortsub>' ' */
goto resume6
--
return

```

B. SQL SERVER 2000 VIEWS

vwGraphs

```
SELECT TOP 100 PERCENT
dbo.tblMishaps.MishapID,
YEAR DATEADD(month, 3, dbo.tblMishaps.MishapDate))
AS [Year], dbo.tblMishaps.Aircraft_FK,
dbo.tblMishaps.Class_FK,
dbo.tblMishapClass.MishapClassDefinition,
dbo.tblMishaps.Type_FK,
dbo.tblMishapType.MishapTypeDefinition,
dbo.tblMishaps.LocationID_FK,
dbo.tblMishapLocation.MishapLocation,
dbo.tblMishaps.OrgID_FK, dbo.tblOrganization.OrgName,
dbo.tblDatabaseType.DatabaseType

FROM dbo.tblMishaps
INNER JOIN dbo.tblDatabaseType
ON dbo.tblMishaps.DatabaseType= dbo.tblDatabaseType.DatabaseType
AND dbo.tblMishaps.DatabaseType= dbo.tblDatabaseType.DatabaseType
INNER JOIN dbo.tblMishapClass
ON dbo.tblMishaps.Class_FK = dbo.tblMishapClass.MishapClassCode
INNER JOIN dbo.tblMishapLocation
ON dbo.tblMishaps.LocationID_FK= dbo.tblMishapLocation.MishapLocationID
INNER JOIN dbo.tblMishapType
ON dbo.tblMishaps.Type_FK = dbo.tblMishapType.MishapTypeCode
INNER JOIN dbo.tblOrganization
ON dbo.tblMishaps.OrgID_FK = dbo.tblOrganization.OrgID

ORDER BY dbo.tblMishaps.MishapID
```

vwReports

```
SELECT  vwGraphs.MishapID, vwGraphs.Year,
        vwGraphs.Aircraft_FK, vwGraphs.Class_FK,
        vwGraphs.MishapClassDefinition, vwGraphs.Type_FK,
        vwGraphs.MishapTypeDefinition, vwGraphs.LocationID_FK,
        vwGraphs.MishapLocation, vwGraphs.OrgID_FK,
        vwGraphs.OrgName, vwGraphs.DatabaseType,
        tblMishapFactors.FactorID, tblFactors.[3rdLevelCode],
        tblFactors.[3rdLevelDesc], tblFactors.[2ndLevelCode],
        tblFactors.[2ndLevelDesc], tblFactors.[1stLevelCode],
        tblFactors.[1stLevelDesc]

FROM    tblFactors INNER JOIN tblMishapFactors
ON      tblFactors.[3rdLevelCode] = tblMishapFactors.[3rdLevelCode_FK]
        INNER JOIN vwGraphs
ON      tblMishapFactors.MishapID_FK = vwGraphs.MishapID
```

vwReport By Aircraft1

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[1stLevelCode],
    vwReports.Aircraft_FK

FROM    vwReports RIGHT OUTER JOIN tblFactors
ON      vwReports.[1stLevelCode] = tblFactors.[1stLevelCode]
```

vwReport By Aircraft2

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[2ndLevelCode],
    vwReports.Aircraft_FK

FROM    vwReports
        RIGHT OUTER JOIN tblFactors
        ON      vwReports.[2ndLevelCode] = tblFactors.[2ndLevelCode]
```

vwReport By Aircraft3

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[3rdLevelCode],
    vwReports.Aircraft_FK

FROM    vwReports RIGHT
        OUTER JOIN tblFactors
        ON      vwReports.[3rdLevelCode] = tblFactors.[3rdLevelCode]
```

vwReport By Class1

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[1stLevelCode],
    vwReports.Class_FK

FROM    vwReports RIGHT OUTER JOIN tblFactors
ON      vwReports.[1stLevelCode] = tblFactors.[1stLevelCode]
```

vwReport By Class2

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[2ndLevelCode],
    vwReports.Class_FK

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[2ndLevelCode] = tblFactors.[2ndLevelCode]
```

vwReport By Class3

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[3rdLevelCode],
    vwReports.Class_FK

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[3rdLevelCode] = tblFactors.[3rdLevelCode]
```

vwReport By Type1

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[1stLevelCode],
    vwReports.Type_FK

FROM    vwReports RIGHT OUTER JOIN tblFactors
ON      vwReports.[1stLevelCode] = tblFactors.[1stLevelCode]
```

vwReport By Type2

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[2ndLevelCode],
    vwReports.Type_FK

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[2ndLevelCode] = tblFactors.[2ndLevelCode]
```

vwReport By Type3

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[3rdLevelCode],
    vwReports.Type_FK

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[3rdLevelCode] = tblFactors.[3rdLevelCode]
```


vwReport By FiscalYear 1

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[1stLevelCode],
    vwReports.Year

FROM    vwReports RIGHT OUTER JOIN tblFactors
ON      vwReports.[1stLevelCode] = tblFactors.[1stLevelCode]
```

vwReport By FiscalYear 2

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[2ndLevelCode],
    vwReports.Year

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[2ndLevelCode] = tblFactors.[2ndLevelCode]
```

vwReport By FiscalYear 3

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[3rdLevelCode],
    vwReports.Year

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[3rdLevelCode] = tblFactors.[3rdLevelCode]
```

vwReport By Location 1

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[1stLevelCode],
    vwReports.LocationID_FK

FROM    vwReports RIGHT OUTER JOIN tblFactors
ON      vwReports.[1stLevelCode] = tblFactors.[1stLevelCode]
```

vwReport By Location 2

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[2ndLevelCode],
    vwReports.LocationID_FK

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[2ndLevelCode] = tblFactors.[2ndLevelCode]
```

vwReport By Location 3

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[3rdLevelCode],
    vwReports.LocationID_FK

FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[3rdLevelCode] = tblFactors.[3rdLevelCode]
```

vwReport By Organization 1

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[1stLevelCode],
    vwReports.OrgID_FK
FROM    vwReports RIGHT OUTER JOIN tblFactors
ON      vwReports.[1stLevelCode] = tblFactors.[1stLevelCode]
```

vwReport By Organization 2

```
SELECT DISTINCT
    vwReports.MishapID, tblFactors.[2ndLevelCode],
    vwReports.OrgID_FK
FROM    vwReports RIGHT OUTER JOIN
    tblFactors
ON      vwReports.[2ndLevelCode] = tblFactors.[2ndLevelCode]
```

vwReport By Organization 3

```
SELECT DISTINCT vwReports.MishapID,
    tblFactors.[3rdLevelCode],
    vwReports.OrgID_FK
FROM    vwReports
    RIGHT OUTER JOIN tblFactors
    ON      vwReports.[3rdLevelCode] = tblFactors.[3rdLevelCode]
```

APPENDIX E. HFACS–ME WEBSITE USABILITY EVALUATION

Background. Thank you for participating in a usability study (evaluation) of a prototype for HFACS–ME Web, a browser-based data retrieval and analysis tool. This tool was developed by CDR Tony Boex, USN as part of a project for his Master of Science program in Information Technology Management at the Naval Postgraduate School. The purpose of this research is to define and develop a prototype web-based aviation safety information management system that will facilitate data collection, organization, query, analysis, and reporting of maintenance errors that contribute to Naval Aviation mishaps, equipment damage, and personnel injury using Appendix O, OPNAVINST 3750.6R, Human Factors Analysis and Classification System Maintenance Extension (HFACS–ME) taxonomy. The HFACS–ME taxonomy is an effective method for classifying and analyzing the presence of human error in maintenance operations leading to major mishaps, accidents of lesser severity, incidents and maintenance related personal injury cases. However, working with a large database (approximately 600 Naval Aviation maintenance-related mishaps in Fiscal Years 1990-1999) is very labor intensive. Given the capability of current browser and database tools, an Internet based information management system will bring HFACS–ME to the next level.

The HFACS–ME Web application is designed to facilitate the retrieval, analysis and identification of common maintenance errors and associated trends. The target audience for this tool includes safety, maintenance and management personnel, mishap investigators and analysts.

Usability Evaluation. You will be given a packet of instructions to guide you through HFACS–ME Web. You will be asked to make comments on the effectiveness and usability of the prototype system during your testing phase. Additionally, you will be asked to complete an “exit survey” after completion of your testing. Questions will include demographic information, objective questions about website usability, and subjective questions and comments for areas not covered in the objective section. The evaluation should take no more than 15-20 minutes.

Completion of Evaluation. Upon completion of your evaluation and exit survey please return your packet to Professor Robert Figlock's Office (E-305, East Wing Herrmann Hall).

Thank you again,
Tony Boex

Instructions for the prototype Human Factors Analysis and Classification System–Maintenance Extension (HFACS–ME) Web Evaluation

1. This Evaluation can be completed from any computer that has access to the Internet.

Please complete the following information (if known):

Computer used: ☐ Government ☐ Personal ☐ Public

Connection Type: ☐ Dial-Up Connection Speed: _____
 ☐ LAN

Screen Resolution: ☐ 640 x 480 ☐ 800 x 600
 ☐ 1024 x 768 ☐ Greater than 1024 x 768
 ☐ Don't Know

Browser Used: ☐ Internet Explorer Version: _____
 ☐ Netscape Version: _____
 ☐ Other _____

2. Once connected to the Internet, open your browser and enter the following URL in the address bar and press Enter: <http://131.120.51.194/safety/>

Home Page

3. The HFACS–ME homepage should now be displayed on your browser. Take a moment to examine the HFACS–ME Home Page.

4. There are five hyperlinks on the blue menu bar located under the HFACS–ME graphic (**Mishap Data**, **Factors Analysis**, **Graph Data**, **Reports**)..

5. Move the cursor over each menu item to display more detailed information about the purpose of that link.

6. Select (click on) the **Mishap Data** menu item.

7. When prompted, enter the following information to gain access to the HFACS website:

UserID: hfacs
Password: hfacs

☐ I was not prompted for a UserID and Password

Question 1: Were you able to access the website? If not, what error message was displayed?

Mishap Details

8. Read the Data Selection instructions.

Question 2: Are the directions understandable as written? How could they be made clearer?

9. Select a single Aircraft Type and click on the Submit button.

Aircraft 1 selected: _____
Number of matching records: _____

Question 3: How would you find out details about a specific mishap?

10. Scroll through the data returned to verify that only data containing your selected Aircraft Type is present.

11. Now, keeping the same Aircraft previously selected, select an additional constraint from one of the other criteria. Click on the Submit button.

Additional criteria selected: _____
Number of matching records: _____

12. Again scroll through the data returned to verify that only data containing **both** your selected criteria is present. (Note: if additional criteria selected was a 1st, 2nd or 3rd Level Factor, then verification must be accomplished on the Mishap Detail page by clicking on each Mishap ID in the main table).

13. Click on the Mishap ID of any record listed in the table and record the following:

MishapID Number: _____
3rd Level Factor: _____ (first one listed)

14. Click the Back button on the bottom of the page or your browser's Back button.

15. Finally, keeping both previous criteria selected, select an additional Aircraft Type (follow the directions listed on the page). Click on the Submit button.

Aircraft 2 selected: _____
Number of matching records: _____

16. Again scroll through the data returned to verify that only data containing either Aircraft AND your additional criteria is present. *Please describe any difficulty encountered with the multiple select options.*

17. Select very restrictive criteria such that a match is extremely unlikely (i.e., P3 mishap while Embarked) and click Submit.

18. Select (click on) the **Factor Analysis** menu item. Record the following information:

Total number of mishaps in HFACS–ME database: _____
Percentage of mishaps involving Management Conditions: _____

Number of mishaps in 1991 that listed a Maintainer **Infraction** as a factor

19. Click on the **Graph Data** menu item and enter the following query. Click Submit:

Aircraft: EA6, F14, F18, H60, P3, S3
Class: C
All other criteria: (All)

20. On the **Grouping Selection** page, select Aircraft Type as primary and Fiscal Year as secondary. Click Show Graph. Record the following information:

Total number of factors involving F18 Aircraft in 1991: _____
Total number of factors involving S3 Aircraft in 1999: _____

21. Now click on the Back button and without changing any criteria, click on Show Graph and Data Table. View the Data Table that opens in a new window.

Question 4: Is the Data Table opening in a new window a distraction? If so, which is preferable, having the Data Table appear below the Graph and scrolling to view it, or having the Data Table open in a new window?

22. Click on the **Reports** menu item. On the Report Selection page, click on Mishap Factors by Aircraft Type Report. When the page loads, use the hyperlinks under the report title to move between reports. Use the Back button to select other reports.

23. Select HOME from the menu bar.

*Question 12: Having used the four main area of the HFACS–ME website, are the menu item names: **Mishap Details**, **Factors Analysis**, **Graphs**, and **Reports**, sufficiently descriptive to guide a new user to the appropriate area of the website? If not, what changes would you make?*

Question 13. Please comment on the following with regard to the HFACS–ME Website: (Use additional sheets if necessary)

a. Visual appeal

b. Layout

c. Appropriate use of colors and graphics

d. Design consistency

e. Functionality of menu items and hyperlinks

24. Please fill out the Exit Survey Questionnaire.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. HFACS–ME WEBSITE USABILITY EVALUATION EXIT SURVEY

Purpose: This survey evaluates a user’s overall satisfaction of the Human Factors Analysis and Classification System–Maintenance Extension (HFACS–ME) Web Prototype. It consists of three parts.

Part I: *Demographic Information.* Part I captures the user’s maintenance affiliation, computer experience, and availability of software and hardware systems used in the Navy and Marine Corps.

Part II: *User Satisfaction with the Four Sections of the HFACS–ME Web Prototype.* Part II deals directly with user feedback as they use the prototype.

Part III: *User Overall Satisfaction with the HFACS–ME Web Prototype.* Part III allows users to give general feedback about the prototype.

Part I. Demographic Information

Follow the instructions after each numbered question or statement.

1. I am attached to a command that **primarily performs maintenance** (military and/or civilian) at the:

(Select one from the list and check the box)

- ☐ Organizational Level (Squadron)
- ☐ Intermediate Level (AIMD)
- ☐ Depot Level (NADEP)
- ☐ Command does not perform aircraft maintenance
- ☐ Other (describe if other) _____

2. On average, how many hours during a typical day do you use each of the following computer applications? (include both on-duty and off-duty use)

Web Browser	_____	Email	_____
Word processor	_____	Spreadsheet	_____
Database	_____	Other	_____

3. What **browser** do you normally use?

(Check all boxes that apply)

	<u>Work</u>	<u>Home</u>
Internet Explorer 6.x	☐	☐
Internet Explorer 5.x	☐	☐
Internet Explorer 4.x	☐	☐
Netscape 6.x	☐	☐
Netscape 5.x	☐	☐
Netscape 4.x	☐	☐
Opera	☐	☐
not sure of version	☐	☐
Not Applicable	☐	☐
Other: _____	☐	☐

4. What computer **operating systems** do you use?

(Check all boxes that apply)

	<u>Work</u>	<u>Home</u>
Windows 9X (95, 98, ME)	☐	☐
Windows NT (4.0, 2000, XP)	☐	☐
Macintosh	☐	☐
UNIX	☐	☐
Linux	☐	☐
Other (describe if other)	☐	☐

Part II. User Satisfaction with the Four Sections of the HFACS–ME Web Interface Prototype

Select the category that best matches your impression of each of the below categories (and check the box).

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I feel the information on the HFACS–ME Web was in a logical form	☐	☐	☐	☐	☐

(comments) _____

I found the HFACS Web easy to navigate	☐	☐	☐	☐	☐

(comments) _____

My tour of the HFACS Web was very interesting	☐	☐	☐	☐	☐

(comments) _____

The information presented on the HFACS–ME Web is relevant to maintenance operations	☐	☐	☐	☐	☐

(comments) _____

The concept of the HFACS Web is a good one.	☐	☐	☐	☐	☐

(comments) _____

Part III. User Overall Satisfaction with the HFACS–ME Web Prototype

Please make any comments on the HFACS–ME Web Prototype not reflected in your comments in sections I and II.

The most **positive** aspects of the HFACS–ME Web Prototype were:

The most **negative** aspects of the HFACS–ME Web Prototype were:

I would make these changes (if any) to the HFACS–ME Web Prototype:

Thank you! Your participation is greatly appreciated!

LIST OF REFERENCES

- Castro, E. (2000). *HTML For the World Wide Web, Fourth Edition: Visual QuickStart Guide*, Peachpit Press, 2000.
- CINCPACFLT Pearl Harbor HI Naval Message, Subject: Information Technology for the 21st Century, 300944Z Mar 97.
- Davenport, T. (1998). *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press.
- Department of the Navy, (1999). *Information Technology Standards Guidance, (Version 99-1)*. [<http://www.doncio.navy.mil/training/oools/itsg/>]. 5 April 99.
- Department of the Navy (2001). *Naval Aviation Safety Program, (OPNAVINST 3750.6R, Appendix O)*. Washington, DC: Office of the Chief of Naval Operations. March 2001.
- Flanders, P.& Tufts, S. (2001). *Software Re-engineering of the Human Factors Analysis and Classification System – (Maintenance Extension) Using Object Oriented Methods in a Microsoft Environment*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 2001.
- Fry, A. (2000). *Modeling and Analysis of Human Error in Naval Aviation Maintenance Mishaps*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 2000.
- Graeber, R. (1999). *Tools Developed by Aircraft Builder Designed to Help Airlines Manage Human Error*, ICAO Journal, 54 (6), 11-14.
- Hibit & Marx (1994). *Reducing Human Error in Aircraft Maintenance Operations with the Maintenance Error Decision Aid*. Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting. Nashville, TN, 111-114.
- Internet Engineering Task Force (IETF) (1999). *RFC 2616: Hypertext Transfer Protocol - HTTP/1.1*. June 1999.
- Marx, D. (1998). *Learning from Our Mistakes: A Review of Maintenance Error Investigation and Analysis Systems* (FAA TR). Atlanta, GA: Galaxy Scientific Corp.
- Naval Safety Center (1999). Aviation Safety Data.
http://www.safetycenter.navy.mil/aviation/Aviation_Data/Aviation_Data.html.

Nelson, D. (2001). *Information Management System Development for the Investigation, Reporting and Analysis of Human Error in Naval Aviation Maintenance*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 2001.

Nutwell & Sherman (1997). *Safety: Changing the Way We Operate*. Naval Aviation News. March-April, 79 (3), 12-15. Washington, DC: Naval Air Systems Command.

QMB (1997). "Human Factors Quality Management Board Charter". [http://safetycenter.navy.mil/Aviation/HFAC/hfac_qmb_charter.htm]. May 1997.

Reuters (2000). *FAA Rule Would Make Airlines Share Safety Data*. www.yahoo.com. June 30, 2000.

Schmidt, J., Figlock, R., & Teeters, C. (1999). *Human Factors Analysis of Naval Transport Aircraft Maintenance and Flight Line Related Incidents (1999-01-2981)*, Society of Automotive Engineers, Inc.

Schmidt, J., Schmorow, D., & Figlock, R. (2000). *Forecasting Return on Investment for Naval Aviation Maintenance Safety Initiatives*. SAE AEMR Conference Proceedings, Long Beach, CA.

Schmidt, J., Schmorow, D., and Hardee, M. (1997). *A Preliminary Human Factors Analysis of Naval Aviation Maintenance Related Mishaps (983111)*, Society of Automotive Engineers, Inc.

Schmorow, D. (1998), *A Human Error Analysis and Model of Naval Aviation Maintenance Related Mishaps*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1998.

School of Aviation Safety (1999). From *Aviation Safety History* presentation. Monterey, CA: Naval Postgraduate School.

SYBEX (2001), *ASP, ADO, and XML Complete*, SYBEX Inc.

Wood, B. (2000), *Information Management System development for the Characterization and Analysis of Human Error in Naval Aviation Maintenance Related Mishaps*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 2000.

World Wide Web Consortium (W3C) (1999), *HTML 4.01 Specification: W3C Recommendation 24 December 1999*, <http://www.w3.org/TR/1999/REC-html401-19991224/>.

www.faa.gov (2000). Federal Aviation Administration website.

www.galaxyscientific.com (2000). Galaxy Scientific Corp. website.

<http://www.microsoft.com/data/archb.htm> (2001). Microsoft Universal data
Access- A Comparison of the ADO and DAO/RDO Architectures.

www.hfskyway.gov (1999). Federal Administration website for Maintenance
Related Mishap training.

www.webstandards.org (2001). The Web Standards Project.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
8725 John J. Kingman Rd., STE 0944
Fort Belvoir, Virginia 22060-6218
2. Dudley Knox Library
Naval Postgraduate School, Code 52
411 Dyer Rd.
Monterey, California 93943-5101
3. John J. Goglia, Honorable
Board Member
National Transportation Safety Board
490 L'Enfant Plaza East, S.W.
Washington, DC 20594-2000

4. Jean Watson
FAA-Office of Aviation Medicine
Medical Specialties Division
AAM-240
800 Independence Avenue, SW
Washington, DC 20591

5. Barbara Kanki, PhD
Crew Factors, Flight Management & Human Factors Division
Mail Stop 262-4
NASA Ames Research Center
Moffett Field, California 94035-1000

6. Commander
Naval Safety Center (Code 10)
375 A Street
Norfolk, Virginia 23511-4399

7. CAPT (sel) John K. Schmidt, USN
Naval Safety Center (Code 145)
Norfolk, VA 23511

8. Professor George A. Zolla
Naval Postgraduate School
Monterey, California 93943

9. Professor Robert C. Figlock
School of Aviation Safety
Naval Postgraduate School, Code 10
1588 Cunningham Rd, Room 303
Monterey, California 93943-5202

10. Don Lawson
PO Box 641
Jeffersonville, VT 05464

11. CDR Anthony R. Boex
656 San Jacinto Place
Chula Vista, California 91914
