

SANDIA REPORT

SAND97-1788 • UC-900

Unlimited Release

Printed September 1997

Technical Manual for the Augmented Computer Exercise for Inspection Training (ACE-IT) Software

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ph

Pauline R. Dobranich, Karl E. Horak, David Hagan, Deborah Evanko, Jon Nelson,
Christine Ryder, Dana Hedlund

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

SAND97-1788
Unlimited Release
Printed September 1997

Distribution
Category UC-900

TECHNICAL MANUAL FOR THE AUGMENTED COMPUTER EXERCISE FOR INSPECTION TRAINING (*ACE-IT*) SOFTWARE

Pauline R. Dobranich
Cooperative Monitoring Center and Regional Security
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1373

Karl E. Horak, David Hagan, Deborah Evanko, Jon Nelson, Christine Ryder, and Dana Hedlund
Ogden Environmental and Energy Services, Inc.
Albuquerque, New Mexico 87110

ABSTRACT

The on-site inspection provisions in many current and proposed arms control agreements require extensive preparation and training on the part of both the Inspection Teams (inspectors) and Inspected Parties (host). Current training techniques include table-top inspections and practice inspections. The Augmented Computer Exercise for Inspection Training (*ACE-IT*), an interactive computer training tool, increases the utility of table-top inspections. *ACE-IT* has been designed to provide training for “challenge inspections” under the Chemical Weapons Convention (CWC); however, this training tool can be modified for other inspection regimes. Although *ACE-IT* provides training from notification of an inspection through post-inspection activities, the primary emphasis of *ACE-IT* is in the inspection itself — particularly with the concept of “managed access.” *ACE-IT* also demonstrates how inspection provisions impact compliance determination and the protection of sensitive information. This Technical Manual describes many of the technical aspects of the *ACE-IT* training software.

Technical Manual for the ACE-IT Software

This page intentionally left blank.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

CONTENTS

EXECUTIVE SUMMARY7
1.0 SYSTEM REQUIREMENTS11
2.0 ACE-IT DIRECTORY STRUCTURE12
3.0 DESCRIPTION OF COMPONENTS14
4.0 SPECIAL SYSTEM FILES17
 4.1 AutoCAD®/ArcCAD®17
 4.2 Visual BASIC®17
 4.3 Access®18
 4.4 Text Files18
 4.5 ZyIndex®19
5.0 VISUAL BASIC® TO ArcCAD®20
 5.1 The FAX Viewer Executable21
 5.2 The Negotiation Executables (NEG_H.EXE, NEG_I.EXE, NEG_M.EXE) 21
 5.2.1 The Inspected Party executable (NEG_H.EXE)21
 5.2.2 The Inspection Team executable (NEG_I.EXE)37
 5.2.3 The Moderator executable (NEG_M.EXE)54
 5.3 OSI.EXE (The ACE-IT Control Module)58
 5.3.1 OSI.EXE Visual Basic® Forms60
 5.4 RECALL.EXE62
 5.4.1 RECALL.EXE Visual Basic® Form62
 5.5 SLEEPER.EXE63
 5.5.1 SLEEPER.EXE Visual Basic® Form64
6.0 THE ACE-IT GIS INTERFACE64
 6.1 Menu Structure64
 6.2 Site Maps64
 6.3 Layers Used in CMC.DWG65
 6.3.1 Layer Control67
7.0 VERSION CONTROL68
8.0 COMPUTER SECURITY69

APPENDIX A-Installation of ACE-IT70

FIGURES

Figure E-1. ACE-IT Schedule of Events7
Figure E-2. Interactive Menu with Digital Images of a Room to be Inspected8
Figure E-3. Inspection Team Negotiations9

Technical Manual for the ACE-IT Software

This page intentionally left blank.

EXECUTIVE SUMMARY

Many current and proposed arms control agreements have on-site inspection requirements. Either government facilities or private firms may be subject to these inspections. The Augmented Computer Exercise for Inspection Training (ACE-IT) is a tool for training both inspectors and hosts.

ACE-IT is an interactive computer program. It teaches users (1) how to conduct an inspection and (2) how the provisions of an on-site inspection affect the ability to determine compliance and protect sensitive information. ACE-IT has been developed for practicing a “challenge inspection” under the Chemical Weapons Convention (CWC).

“Challenge Inspections” may occur at any time and anywhere—without the right of refusal and with little advance notice. Therefore, advance training is important.

ACE-IT training is conducted with the trainees divided into two teams: an “Inspection Team” (inspectors) and an “Inspected Party” (hosts). An exercise moderator controls the exercise. The training includes all of the events allowed by the CWC, from notification of an inspection through post-inspection procedures.

This technical manual provides a detailed description of the components used by the ACE-IT system to conduct a simulated inspection. The manual is written in sufficient detail to allow complete description of the system operation, and complete installation and operation of the system.

Overview of Technical Information

Control Module

The control module shows the schedule for the “challenge inspection” as shown in Figure E-1. The time that would be required for an actual inspection is compressed for the training exercise. All events are automatically recorded in an inspection log, and users may also add their own notes. The moderator may pause the exercise or jump ahead to the next scheduled event.

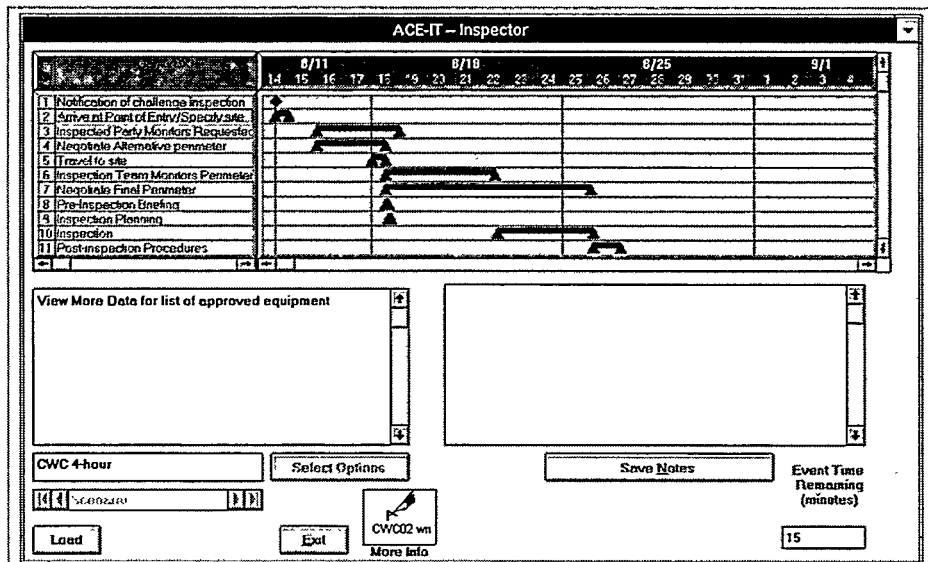


Figure E-1. ACE-IT Schedule of Events

Technical Manual for the ACE-IT Software

Geographic Information System (GIS) Building Maps

There are three buildings available for inspection: a “single small-scale facility,” an office building, and a demonstration building. At the “single small-scale facility,” a country produces Schedule 1 chemicals for research, medical, pharmaceutical, and protective purposes.

Interactive Menus

Interactive menus allow the Inspection Team and the Inspected Party to “conduct” the inspection. These menus teach the important concept of *managed access* — i.e., how the Inspected Party can protect sensitive information that is unrelated to the CWC. This includes controlling physical access to a room, negotiating permission to look beneath shrouds, and controlling requests for alternate information.

Digital Images

Digital images provide users with multiple views of each room. The images may be zoomed to full screen. (See Figure E-2.)

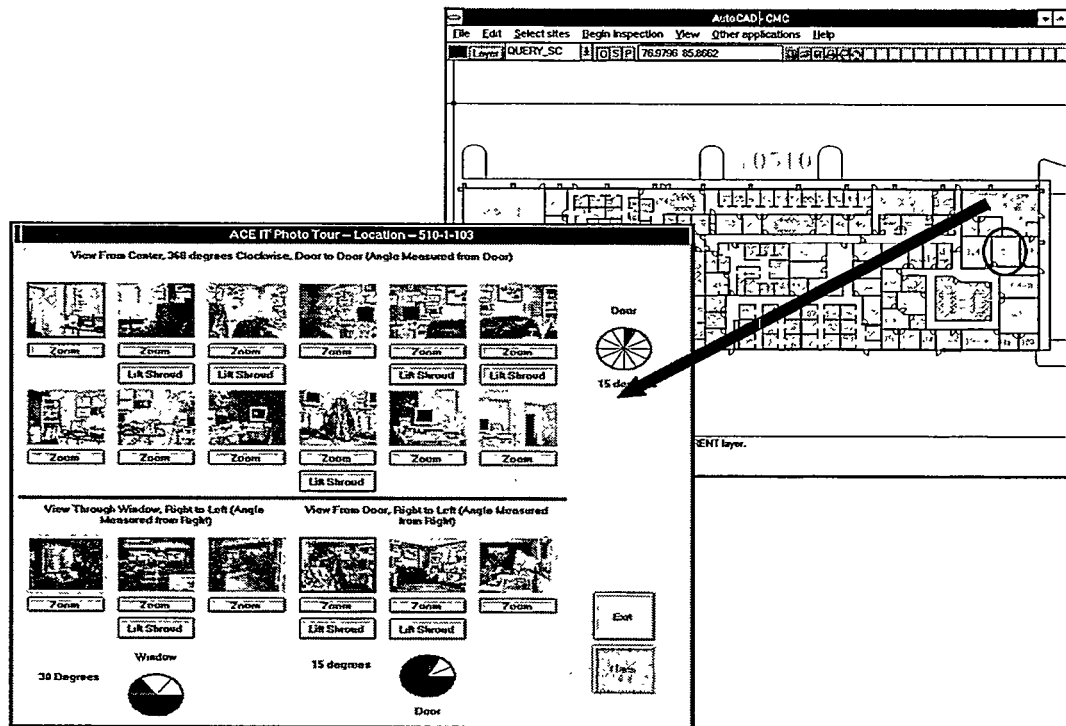
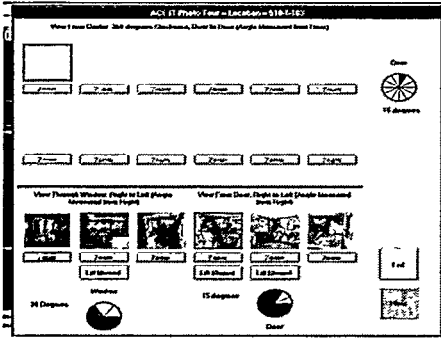


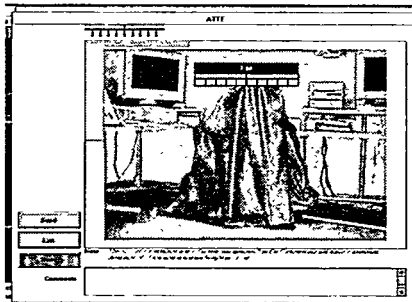
Figure E-2. Interactive Menu with Digital Images of a Room to be Inspected

Tabular Data

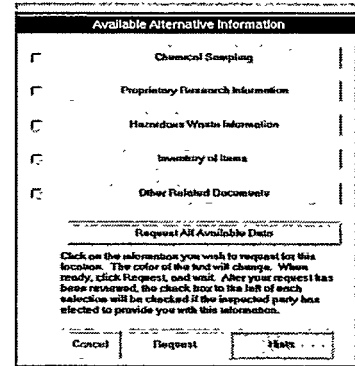
The user can access data associated with each room, if permission to access this information is granted by the Inspected Party. The data include inventory of items in each room, chemical sampling results, proprietary research information, hazardous waste information, and other related documents.



Partial Access (View through window and door only)

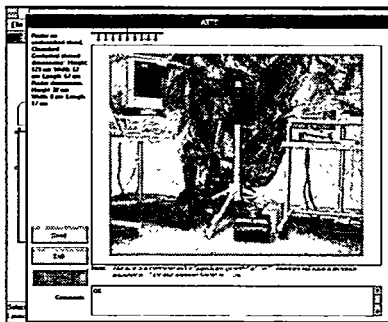
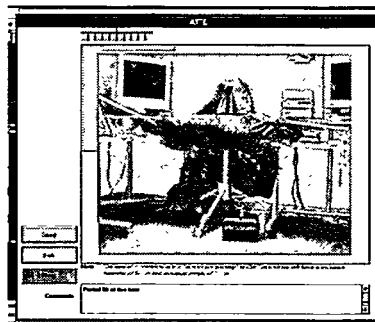


Shrouded Item (Shroud not lifted)



Alternate Information Available

Shrouded Item (Shroud partially lifted)



Shrouded Item (Shroud completely lifted)

Figure E-3. Inspection Team Negotiations

Technical Manual for the ACE-IT Software

Text Retrieval

Some documents are not associated with a particular location (CWC text and safety plans). These documents are displayed as hypertext documents or as electronic fax documents.

Additional Features

In addition to the interactive windows of the **ACE-IT** system, users are provided with a “Chat” feature, allowing them to hold discussions across the computer network. Each message is captured by the system for inclusion into the log.

Hardware and Software Requirements

ACE-IT consists of a suite of commercial software that runs on a local area network with three personal computers. Each computer is at least 486/66-class, with 32 MB RAM and 2 GB hard drive, capable of video display at a resolution of at least 1024 by 768. The operating system is Windows 3.11.

Custom applications have been developed to integrate the system components using several commercial products:

- AutoCAD[®] by Autodesk—graphics engine
- ArcCAD[®] by ESRI—geographical information system
- Access[®] by Microsoft—database
- ZyIndex[®] by ZyLAB—text retrieval
- Visual BASIC[®] by Microsoft—program development
- Windows for Workgroups by Microsoft—local area network

Other Applications

In addition to providing training in conducting “challenge inspections” under the CWC, **ACE-IT** can be used to study managed access techniques for other inspection regimes and manage time during an inspection. Other agreements, inspection regimes, and buildings could be added to **ACE-IT**.

Availability of ACE-IT

All of the equipment and most of the software is commercially available. The customized code is protected by copyright, and available from Sandia National Laboratories and Ogden Environmental and Energy Services.

Technical Manual for the ACE-IT Software

1.0 SYSTEM REQUIREMENTS

The following commercial software is required for the systems running the **ACE-IT** program. These packages should be installed in the listed order.

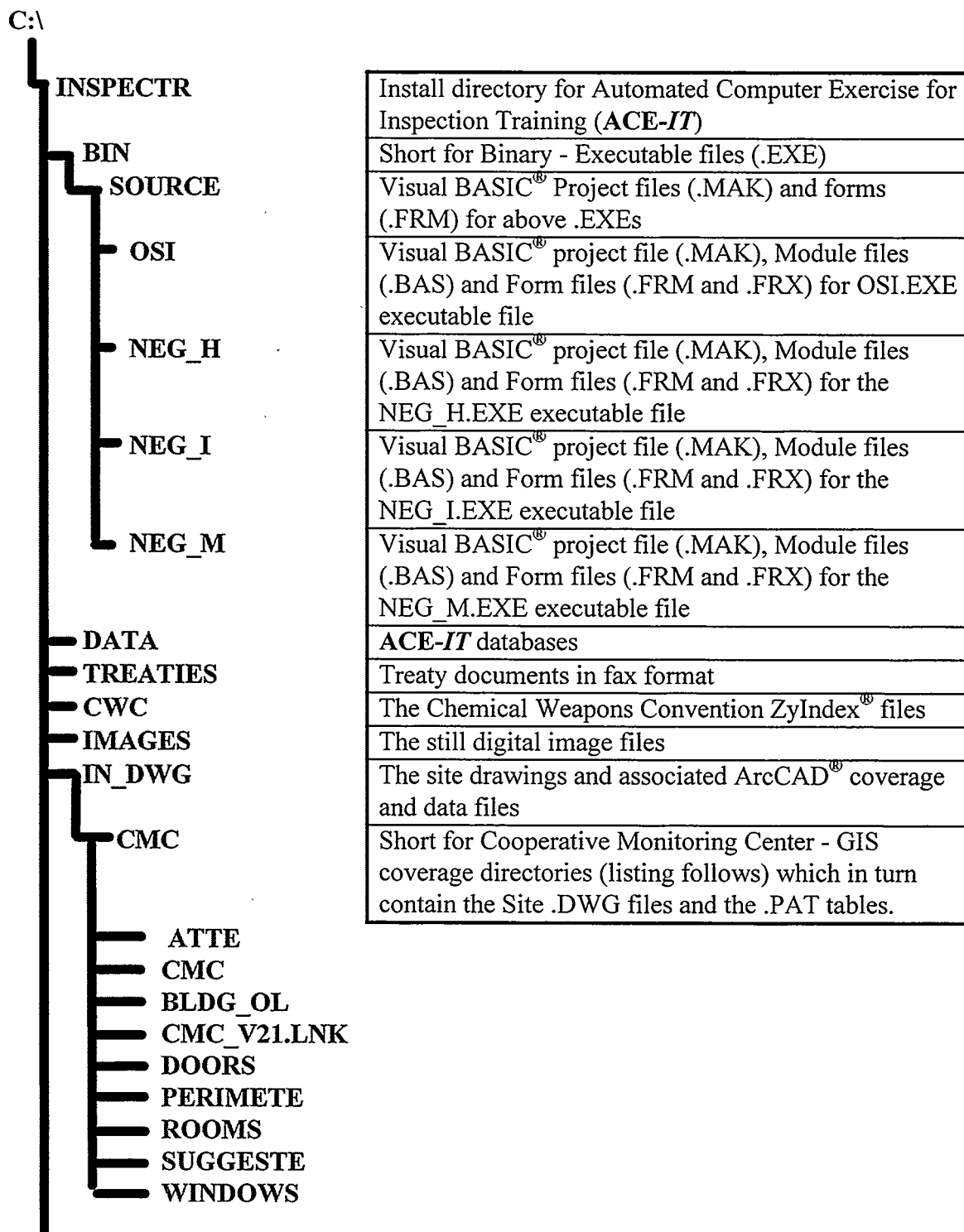
1. DOS 6.0 or higher
2. Windows[®] 3.11
3. AutoCAD[®] release 12c (for Windows)
4. ArcCAD[®] 11.3 (for Windows)
5. Access[®] 2.0
6. ZyIndex[®] 5.15 (for Windows)
7. Visual BASIC[®] 3.0 (optional)
8. Access[®] 2.0 Compatibility layer for Visual BASIC[®]

The following hardware is required for systems running **ACE-IT**.

1. Three 486/66-class, PC-based computers each with at least 32 MB RAM and 500MB of free hard drive space, and a local area network.
2. ArcCAD[®] Hardware Key

2.0 ACE-IT DIRECTORY STRUCTURE

The directory structure used in the ACE-IT System is integral to system operation. This directory structure should not be modified under any circumstances, as such modifications will result in a nonfunctional system. Each directory or subdirectory contains files that if moved to other locations will cause errors and malfunctions.



C:\INSPECTR (cont)

- INI**
- MENU**
- NOTES**
- VIDEO**

| |
|---|
| The initialization (.INI) files |
| The various menus (.MNU and .MNX) needed for different aspects of ACE-IT and the various LISP Programs (.LSP) that drive ACE-IT |
| Windows [®] Write (.WRI) files for the Notes Box in the Control Module. |
| Video (.AVI) and QTVR (.MOV) files for shroud lifting videos and QTVR Scenes |

Technical Manual for the ACE-IT Software

3.0 DESCRIPTION OF COMPONENTS

The **ACE-IT** System uses a combination of commercially available software packages and custom-developed executable files. These components include MicroSoft Access[®], AutoCAD[®], ArcCAD[®], ZyIndex[®], and Modules developed using MicroSoft Visual BASIC[®]. Because ArcCAD[®] stores GIS coverage information according to the original location of a drawing file, the **ACE-IT** System must be installed to the directory **C:\INSPECTR**. This directory can be changed by using ArcCAD[®] either to copy or move the coverage to another directory, and to update the OSI.INI file contents. Commercial software, such as AutoCAD[®], ArcCAD[®], and ZyIndex should be installed according to the manufacturers' instructions.

NOTE: *ACE-IT must reside on drive and directory C:\INSPECTR because of how the ArcCAD[®] handles file tracking; it is not possible to simply use the file manager to copy the GIS coverages and their associated .PAT files without destroying the integrity of the system. You must use ArcCAD's[®] COPY COVERAGE command to do this.*

Applications software (C:\INSPECTR and subdirectories)

The directory **C:\INSPECTR\BIN** stores all Visual BASIC[®] executable files.

| | |
|-----------------------|---|
| FAX.EXE | View treaty documents |
| NEG_H.EXE | Building/Room access negotiation for Inspected Party role |
| NEG_I.EXE | Building/Room access negotiation for Inspection Team role |
| NEG_M.EXE | Building/Room access negotiation for Moderator role |
| OSI.EXE | Exercise clock/Gantt controller |
| RECALL.EXE | Tool for recall of last location visited for Inspected Party role |
| SLEEPER.EXE | GIS communication tool for Inspection Team role |
| SOURCE (subdirectory) | Visual BASIC [®] Project files (.MAK) and forms (.FRM) for listed EXEs |

The directory **C:\INSPECTR\DATA** stores all MicroSoft Access[®] databases.

| | |
|--------------|---|
| COMM.MDB | Communications and event recording database |
| SCENARIO.MDB | Inspection scenario database |

Technical Manual for the ACE-IT Software

The directory **C:\INSPECTR\IMAGES** stores all the still digital images (.BMP files) that can be accessed by the software. *NOTE: All still digital images must be stored in this **single** directory.*

The directory **C:\INSPECTR\IN_DWG** stores all the AutoCAD® site drawings (.DWG) and the ArcCAD® GIS coverage files (.LNK) in the subdirectory structure created by ArcCAD®.

CMC (subdirectory) AutoCAD® site drawings and ArcCAD® GIS interface files for Cooperative Monitoring Center

The directory **C:\INSPECTR\INI** stores the initialization (.INI) files created and/or used by the Visual BASIC® executables. The name of each initialization file matches the name of the executable file that it supports.

| | | |
|-------------|-----------|-----------|
| Fat.INI | Neg_i.INI | Neg_m.INI |
| Sleeper.INI | Neg_h.INI | Osi.INI |
| Recall.INI | | |

The directory **C:\INSPECTR\MENU** stores all the menus needed in the **ACE-IT** suite of software tools plus the custom AutoLISP® routines written for **ACE-IT**. AutoLISP® is a special subset of the Lisp programming language, which is specific to AutoCAD® products.

| | |
|-------------|---|
| 500.LSP | AutoLISP file that recalls the saved AutoCAD® view of building 10500 interior |
| 500EVAC.LSP | AutoLISP file that recalls the saved AutoCAD® view of building 10500 evacuation routes |
| 510.LSP | AutoLISP file that recalls the saved AutoCAD® view of building 10510 interior |
| 510EVAC.LSP | AutoLISP file that recalls the saved AutoCAD® view of building 10510 evacuation routes |
| 520.LSP | AutoLISP file that recalls the saved AutoCAD® view of building 10520 interior |
| 520EVAC.LSP | AutoLISP file that recalls the saved AutoCAD® view of building 10520 evacuation routes |
| ALL_EXT.LSP | AutoLISP file that recalls the saved AutoCAD® view of all CMC Site building exteriors |
| ALL_INT.LSP | AutoLISP file that recalls the saved AutoCAD® view of all CMC site building interiors |
| CMC.LSP | AutoLISP file that recalls the saved AutoCAD® view of the CMC site |
| PERIM.LSP | AutoLISP file that recalls the saved AutoCAD® view of the CMC site requested and alternative perimeters |
| SENSORS.LSP | AutoLISP file that recalls the saved AutoCAD® view of the CMC site sensors |
| ARCAD12.MNU | The original customized ACE-IT menu is automatically loaded every time ArcCAD® is loaded. |

Technical Manual for the ACE-IT Software

| | |
|-------------|---|
| ARCADOR.MNU | Short for ArcCAD [®] Original, this is a renamed copy of the original, unaltered version of the ArcCAD [®] menu |
| ARCAD12.MNX | Compiled version of ARCAD12.MNU file |
| ARCADOR.MNX | Compiled version of ARCADOR.MNU file |

The directory **C:\INSPECTR\NOTES** stores the Write[®] files needed by the OLE function on the Gantt game control form. The contents of these files can be changed or updated using the Windows[®] Write program. The specific files and their functions are listed below.

| | |
|-----------|--|
| CWC01.WRI | Additional information for Step 1 of the CWC Challenge Exercise |
| CWC02.WRI | Additional information for Step 2 of the CWC Challenge Exercise |
| CWC03.WRI | Additional information for Step 3 of the CWC Challenge Exercise |
| CWC04.WRI | Additional information for Step 4 of the CWC Challenge Exercise |
| CWC05.WRI | Additional information for Step 5 of the CWC Challenge Exercise |
| CWC06.WRI | Additional information for Step 6 of the CWC Challenge Exercise |
| CWC07.WRI | Additional information for Step 7 of the CWC Challenge Exercise |
| CWC08.WRI | Additional information for Step 8 of the CWC Challenge Exercise |
| CWC09.WRI | Additional information for Step 9 of the CWC Challenge Exercise |
| CWC10.WRI | Additional information for Step 10 of the CWC Challenge Exercise |
| CWC11.WRI | Additional information for Step 11 of the CWC Challenge Exercise |

The directory **C:\INSPECTR\VIDEO** stores video files (.AVI) used by **ACE-IT** to lift shrouds from the photo tour function.

The directory **C:\ARCADWIN\BIN** is integral to the installation of ArcCAD[®]. In this directory, the file ACAD.LSP determines the system behavior on AutoCAD[®] load. The default file created by ArcCAD[®] on installation prompts the user by asking permission to load ArcCAD[®]. The customized LISP file automatically loads ArcCAD[®] when AutoCAD[®] is loaded, regardless of what drawing file has been opened.

| | |
|----------|--|
| ACAD.LSP | Startup AutoLISP [®] file. This file activates an automatic ArcCAD [®] load on the Inspection Team computer. |
|----------|--|

4.0 SPECIAL SYSTEM FILES

There are various file formats used by the **ACE-IT** system. The following sections discuss the basic types and their generic functions. More detailed descriptions of the customized portions of the **ACE-IT** system follow in Sections 5.0 and 6.0.

4.1 AUTOCAD®/ARCCAD®

The **ACE-IT** system uses the AutoCAD® drafting package as its graphical display module. The AutoCAD® menu (.MNU and .MNX) files are specific to each computer in the network and are not interchangeable. A detailed discussion of ArcCAD® functionality is provided in Section 6.0 “The **ACE-IT** GIS Interface”. The ArcCAD® GIS engine is used to create GIS coverages and databases. The primary coverage created for the use of the **ACE-IT** system is the ROOMS coverage, built on the CMC drawing. Other data stored in this coverage is the unique room number stored as the ACAD_TEXT field in the optional properties table. The ArcCAD® function xLoad is used to activate the SLEEPER.EXE file on the Inspection Team Computer with the command parameter specific to the location selected.

4.1.1 AUTOLISP®

AutoLISP® is specialized subset of the LISP programming language, designed specifically for AutoDesk® products. The AutoLISP® files called by the menus of the **ACE-IT** system are interchangeable between computers. These menus provide the users with the ability to manipulate the views presented in the AutoCAD® window. A listing of these files appears in Section 5.0. Backup copies of all files in the C:\INSPECTRMENU\ directory should be kept.

4.2 VISUAL BASIC®

Seven separate customized applications developed in Visual BASIC® provide most of the **ACE-IT** system functionality.

1. FAX.EXE, which is a stand-alone executable called by various components of the **ACE-IT** system to view graphical FAX versions of selected documents.
2. NEG_H.EXE, the inspected party portion of the **ACE-IT** negotiation module.
3. NEG_I.EXE, the inspection team portion of the **ACE-IT** negotiation module.
4. NEG_M.EXE, the moderator portion of the **ACE-IT** negotiation module.
5. OSI.EXE, the primary control module for the **ACE-IT** system.
6. RECALL.EXE, native only to the Inspected Party computer, where it provides the Inspected Party with the ability to recall the last requested room.
7. SLEEPER.EXE, native to the Inspection Team Computer, where it provides the Inspection Team with the ability to point and click on maps to activate the negotiation module.

The FAX.EXE code is identical for each computer and is interchangeable. The FAX.EXE file requires that the C:\INSPECTR\DATA\TREATIES\ subdirectory exists on each computer.

Technical Manual for the ACE-IT Software

The negotiation modules of the code are not interchangeable and must reside on the proper computer in the **ACE-IT** network to function properly. Which executable is used by the system depends on the conditions provided to OSI.EXE as initial variables. NEG_I.EXE is used for the Inspection Team functionality, NEG_H.EXE for the Inspected Party functionality, and NEG_M.EXE for the Moderator functionality.

The OSI.EXE code is interchangeable for each computer and is manipulated by altering the content of the OSI.INI file that resides in the C:\INSPECTR\INI subdirectory. The system will not run without OSI.INI on each computer.

4.3 ACCESS®

There are two Access databases associated with the **ACE-IT** system. Both of these database files must reside on the moderator computer for the system to function properly. The first database used by the system is SCENARIO.MDB, which resides in the C:\INSPECTR\DATA\ subdirectory. This file is used by only the OSI.EXE program. The second database is COMM.MDB, residing in the same directory. This file is used by OSI.EXE as well as all three negotiation executable files.

The SCENARIO.MDB contains all of the data required to activate and run the OSI.EXE control module, including Gantt chart parameters, scenario timelines, and the associated queries necessary to activate this module. The COMM.MDB contains all the tables, data, and queries necessary to activate and run the negotiation modules (NEG_H.EXE, NEG_I.EXE, and NEG_M.EXE).

4.4 TEXT FILES

The text files involved in the operation of the **ACE-IT** system are created, as required, by either the OSI.EXE executable file or the negotiation executable files. By default, these files always reside on the root directory (C:\) of each computer on the network. If the system is running properly, users will not need to manipulate these files. In the event of an abnormal interruption of **ACE-IT**, for example, a power failure, use the DOS command "delete C:*.ACE" to erase any text files before restarting. The following is a list of all possible text files that may exist at any given time. Because the **ACE-IT** system manipulates these files during run-time, creating and deleting them to achieve certain functionality, all of the listed files may or may not exist on any given computer.

| | | | | |
|-------------|--------------|-------------|--------------|--------------|
| Access.ACE | DataReq.ACE | Jump.ACE | ChatAns.ACE | OSIXTime.ACE |
| Calling.ACE | DataResp.ACE | Moddata.ACE | HostData.ACE | Remain.ACE |
| Chat.ACE | Hangup.ACE | OSIDate.ACE | OSITime.ACE | Sleeper.ACE |
| Status.ACE | | | | |

4.5 ZYINDEX[®]

ZyIndex[®], a product of ZyLabs, Inc, is used as a hypertext search engine in the **ACE-IT** System. ZyIndex should be installed according to the manufacturer's instructions and is best accessed via the Windows[®] 3.11 Program Manager.

The **C:\INSPECTR\DATA\TREATIES** subdirectory contains a version of the CWC text which has been indexed using ZyIndex[®]. Once ZyIndex[®] has been installed, this file can be accessed using the "Open" command in the ZyIndex[®] window.

Technical Manual for the ACE-IT Software

5.0 VISUAL BASIC[®] TO ARCCAD[®]

ArcCAD[®] is the GIS engine that provides the Visual BASIC[®] executable files with the unique location identifiers from the CMC maps. Only the Inspection Team computer uses the GIS to identify specific locations. The following sections discuss the interactions between the various executable files of the ACE-IT system. AutoCAD[®] and ArcCAD[®] are automatically loaded through the OLE capabilities built into the OSI.EXE executable.

Initialization Files

The ACE-IT system software uses two local initialization (INI) files that provide information on the current installation setup, database files, and network configuration. The first INI file used by the system is called INSTALL.INI, and can be found in the C:\INSPECTR\BIN\ subdirectory. This file provides only one piece of information about the system, the installation directory. If the user has modified the ArcCAD[®] coverages and renamed the installation directory, the user can change the contents of this file to reflect the new directory, and then can start the system. Example contents are as follows:

Install Directory = C:\inspectr\

If at any time the system cannot find this file, the software will prompt the user to create this local INI file and continue running the system.

The second INI file, OSI.INI, can be found in the C:\INSPECTR\INI\ subdirectory. This INI file contains database and network configuration information. If this file is deleted, then the system will prompt the user to create a new file. Example contents are as follows:

ACE-IT Role = Moderator
Inspection Team Drive = I:\
Inspected Party Drive = H:\
Moderator Drive = M:\
Scenario Database = SCENARIO.MDB
Communications Database = COMM.MDB

This file indicates that the current role for this computer is the Moderator, the network drive connection to the Inspection Team is I:\, and the connection to the Inspected Party is H:\. The other Inspection Team and Inspected Party computers reference the moderator as network drive M:\. The database containing the CWC challenge inspection scenario is SCENARIO.MDB and the database used for storing and running the exercise is COMM.MDB. If the system is operating properly, and if the user decides to modify one or more of the parameters set in the INI file, it is highly recommended that a backup copy of the current INI file be made prior to file modification.

5.1 THE FAX VIEWER EXECUTABLE

The FAX.EXE file is a stand-alone executable file that uses the Visual BASIC® “Shell” command to activate the Microsoft® FAX viewer included with the Windows® 3.11. While FAX.EXE can be activated independently of the ACE-IT System, it is also called by the system as an external executable file to provide access to various types of information. The documents called by FAX.EXE are stored in a graphical FAX format, and cannot be searched, formatted, or otherwise modified by the user. These fax documents may be considered the equivalent of hardcopy (print) documents.

5.2 THE NEGOTIATION EXECUTABLES (NEG_H.EXE, NEG_I.EXE, NEG_M.EXE)

Three separate modules of code, one on each computer, interact with each other to create the ACE-IT system Negotiation Module. The Negotiation Module is automatically activated by OSI.EXE when the scenario timeline reaches the Inspection event. Once the Inspection event has begun, all date/time values are provided to all three computers from the Moderator’s OSI.EXE ACE-IT Control Module.

5.2.1 The Inspected Party Executable (NEG_H.EXE)

The AutoCAD® map for the CMC is automatically activated by OSI.EXE when the scenario timeline reaches the Inspection event.

NEG_H.EXE runs in the background from the time it is started by the Control Module until it is shut down by the user manually at the end of the exercise. ACE-IT assumes that all requests originating from the Inspection Team are for the maximum available access to a requested location. NEG_H.EXE monitors the Moderator’s computer (using Visual BASIC® timer controls) to determine if a request has been received, logged and passed to the Inspected Party. NEG_H.EXE interfaces with the COMM.MDB database file residing on the Moderator computer to access the required data for the requested location. On receipt of a request, NEG_H.EXE determines the type of request: physical access, alternate data, or lift a shroud. Corresponding to the type of request, there are three possible responses.

For a physical access request, NEG_H.EXE reads a text file residing on the Moderator computer that provides the requested location. This location is a parameter in stored queries in the COMM.MDB database. The system provides the Inspected Party with the option to grant three levels of physical access: full access, partial access, or denied access. The Inspected Party can preview the physical layout, the photo tour, the shrouds, and the available alternate data for the requested location. Based on this preview, the Inspected Party determines the level of access to be granted, and passes it to the Moderator’s computer (NEG_M.EXE), along with any justification for granting that level of access. This information is written to a text file on the Moderator’s computer, and the NEG_H.EXE file returns to background. The moderator’s computer (NEG_M.EXE) receives the response, logs it, and forwards it to the Inspection Team’s computer (NEG_I.EXE).

Technical Manual for the ACE-IT Software

For an alternate information request, the system provides the Inspected Party with the option to grant or deny access to five topics of alternate data. NEG_H.EXE reads a text file residing on the Moderator computer to determine the topics of alternate data requested by the Inspection Team and the location associated with the request. The NEG_I.EXE executable file creates a unique text file for each possible combination of requested topics. This file is parsed by the NEG_H.EXE file to determine which topics are being requested by the Inspection Team. NEG_H.EXE provides the capability to preview all the available alternate information for the requested location. The Inspected Party determines which topics of alternate data will have access granted and responds to the Moderator's computer (NEG_M.EXE). The moderator's computer (NEG_M.EXE) receives the response, logs it, and forwards it to the Inspection Team's computer (NEG_I.EXE).

For a lift shroud request, the system allows the Inspected Party to preview video and the description/dimensions of the requested shrouded object. The Inspected Party can grant three levels of access: full access, partial access (0% to 100% shroud lift), or deny access. The Inspected Party determines the level of access to grant, and responds to the Moderator's computer (NEG_M.EXE), along with any justification for granting that level of access. The moderator's computer (NEG_M.EXE) receives the response, logs it, and forwards it to the Inspection Team's computer (NEG_I.EXE).

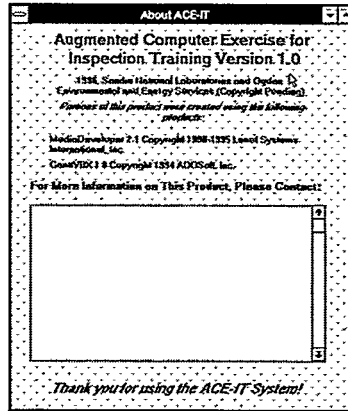
The NEG_H.EXE also has an internal chat feature, which is activated by clicking on the CALL ANOTHER PLAYER button located on all HINTS forms. This chat feature utilizes internal timer controls to detect calls for connections, display/send text, and disconnect messages. All text that is sent using the chat feature is saved to the exercise log. Text that is not sent through the chat feature is not saved, and the system does not mirror keystrokes like the Windows® chat function. NEG_H.EXE, NEG_I.EXE, and NEG_M.EXE must all be running in order for the chat feature to operate.

Once a response to a request has been sent to the Moderator, NEG_H.EXE returns to background and waits for the next request.

Technical Manual for the ACE-IT Software

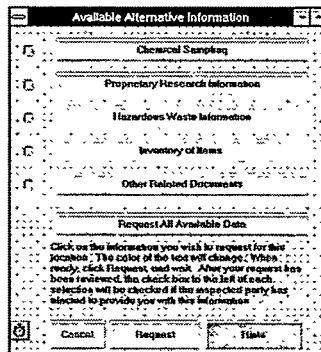
5.2.1.1 NEG_H.EXE Visual BASIC® Forms

The following forms comprise the source code for NEG_H.EXE. They are provided in alphabetical order, with a table listing the major functions available on each form. When used in conjunction with Visual BASIC® in the development, the specific controls referenced for each form can be identified, and the source code for each function accessed.



Form About.Frm

| Control | Function |
|---------|---|
| Text1 | Text Box that displays contact points for ACE-IT system. Multiline and scroll bar properties have been set. |

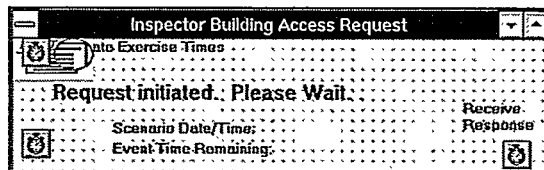


Form AltData.Frm

| Control | Function |
|-------------|---|
| Description | Control array that either indicates the topics of alternate data being requested, by a change in text color, or accesses the appropriate database queries and forms to load the available information for the selected topic. |
| Include | Control array of check boxes that indicates the topics of alternate information to which access will be granted. Not enabled, state is modified by code only. |
| Timer1 | Timer control that puts system into wait mode, and checks for the creation of the text file dataresp.ACE on the Moderator computer to continue the operation of the system. |
| Cancel | Control that unloads the form, without sending a request. |
| Return | Control that determines which topics of alternate information are being requested, creates a request, and then puts the system into wait mode. |
| Hints | Loads the Hint2.Frm form. |

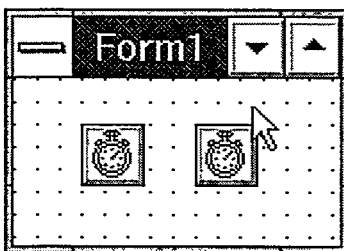
Form Altdata2.frm

| Control | Function |
|-------------|---|
| Description | Control array that activates the proper database queries and forms for each of the functions. |
| Include | Control array of check boxes that indicates whether or not to allow access to the selected topic. |
| PassTo | Returns the Inspected Party's response to a request for alternate data by the Inspection Team. |
| Hints | Loads the Hint5.Frm Form. |



Form BldgNegI.Frm

| Control | Function |
|---------|--|
| Timer1 | Timer control that provides the animation to the Request Initiated label. |
| Timer2 | Timer control that checks for the creation of the response in the form of a database file created on the Moderator computer. When response is received, loads Response.Frm form. |
| Timer3 | Timer control that gets the current exercise times from the Moderator Computer. |



Form Chatchek.Frm

| Control | Function |
|--------------|--|
| ChatCheker | Timer Control that checks for activation of the Chat feature. |
| AnswerCheker | Timer Control that checks for the other user to answer the Chat ring before activating the Chat Feature. |

The screenshot shows a software form titled "Sample Type and Analysis". It is divided into several sections:

- Sample ID:** A group of six text input fields arranged in two columns. The left column contains "Sample ID", "Time", and "Date". The right column contains "Site", "Building", and "Room".
- Sample Type:** A section with the instruction "(Select only one type)". It contains two rows of radio buttons. The first row has "Soil Sample", "Wipe/Other Solid", and "Air Filter". The second row has "Water Sample" and "Process Sample".
- Analysis Type:** A section with the instruction "(Select only one type)". It contains a single row of three radio buttons: "Chemical", "Biological", and "Radiological".
- Buttons:** At the bottom of the form are three buttons: "OK", "Cancel", and a Help icon (a question mark inside a square).

Form ChemSam.Frm

| Control | Function |
|--------------|--|
| Option3d1 | Control array of radio buttons that indicate the type of analysis to be performed on the sample recorded. |
| Option3d2 | Control array of radio buttons that indicate what type of sample will be recorded. |
| Timer1 | Timer control that forces a delay between the recording of the sample and display of results. (1 minute) |
| SampleOK | Control that records the sample and activates the Timer1 timer control. Hides the current form, but leaves it running. |
| SampleCancel | Control that unloads the form without recording a sample. |

Technical Manual for the ACE-IT Software

Form Crada_1.Frm

| Control | Function |
|--------------------|---|
| MoveFirstButton | Control that moves selected recordset to the first available record. |
| MoveNextButton | Control that moves the selected recordset to the next available record. |
| MovePreviousButton | Control that moves the selected recordset to the previous available record. |
| MoveLastButton | Control that moves the selected recordset to the last available record. |
| Command1 | Control that unloads the form. |

Form Haz_W.Frm

| Control | Function |
|--------------------|---|
| MoveFirstButton | Control that moves selected recordset to the first available record. |
| MoveNextButton | Control that moves the selected recordset to the next available record. |
| MovePreviousButton | Control that moves the selected recordset to the previous available record. |
| MoveLastButton | Control that moves the selected recordset to the last available record. |
| Command2 | Control that unloads the form. |

Form Hint1.Frm

Form Hint2.Frm

| Control | Function |
|--------------|--|
| AltInfo | Loads the Alternate Information Request Form. |
| CancelButton | Unloads the form without taking any other actions. |
| Panel3d1 | Control that loads the OurChat.frm to activate the Chat feature. |

Form Hint3.Frm

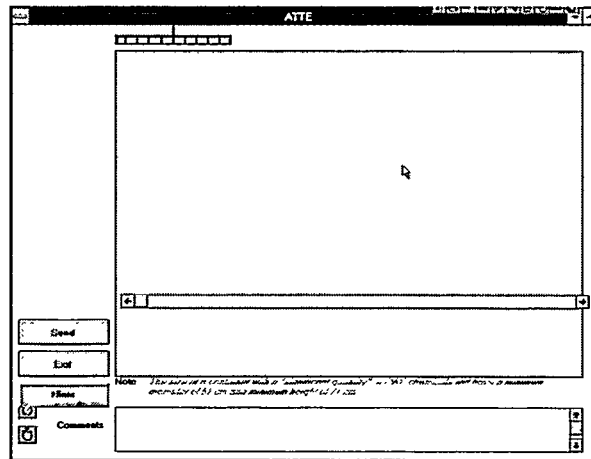
Form Hint4.Frm

| Control | Function |
|--------------|--|
| AltInfo | Loads the Alternate Information Request Form. |
| CancelButton | Unloads the form without taking any other actions. |
| Panel3d1 | Control that loads the OurChat.frm to activate the Chat feature. |

Form Hint5.Frm

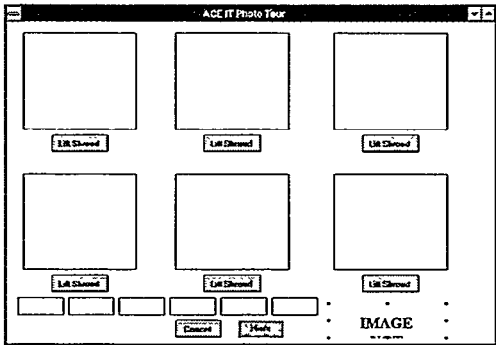
Form Hint6.Frm

| Control | Function |
|--------------|--|
| AltInfo | Loads the Alternate Information Request Form. |
| CancelButton | Unloads the form without taking any other actions. |
| Panel3d1 | Control that loads the OurChat.frm to activate the Chat feature. |



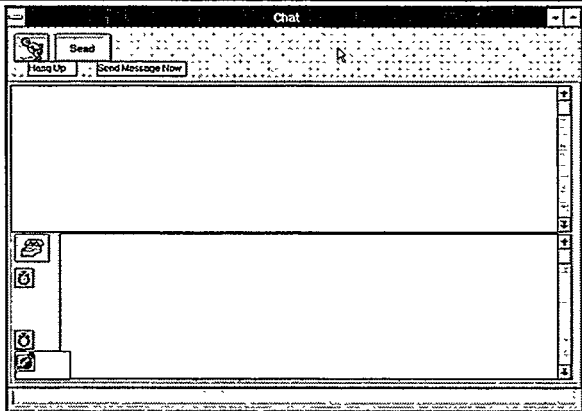
Form Lift.Frm

| Control | Function |
|------------|---|
| Text1 | Text box control that displays the available data for the shrouded item only if lift of the shroud is granted to the 100% level. |
| Command2 | Sends the request to lift the shroud from an item and activates the Timer1 timer control, putting the system into a wait mode. |
| Command1 | Unloads the form without taking any other actions. |
| Hintbutton | Loads the Hint6.Frm form. |
| Text2 | Displays and stores comments associated with the current request. |
| measure | Image box used to activate the on-screen meter stick. |
| MediaDev1 | Displays the currently selected shrouded item. |
| Timer1 | Timer control that puts the system into a wait mode, and checks for the creation of a response on the Moderator Computer. Once a response is received, activates the Timer2 timer control to introduce a delay. |
| Timer2 | Timer control that introduces a delay between the receipt of a response, and the carrying out of appropriate actions automatically. |



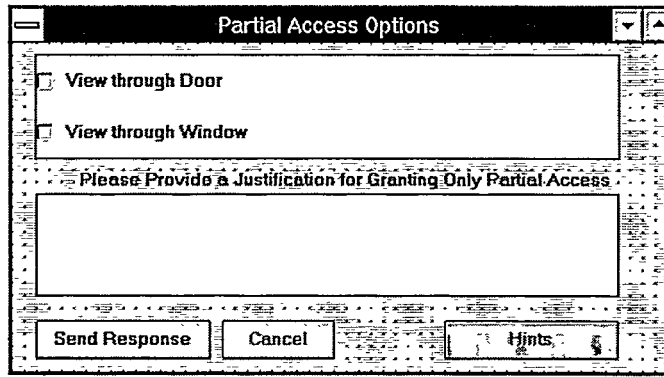
Form MultForm.Frm

| Control | Function |
|----------|---|
| VidImage | Control array that displays the starting frame of each video file. |
| Lifter | Control array that loads the Lift.Frm form with the selected video file and associated information. |
| Command1 | Control that unloads the form. |



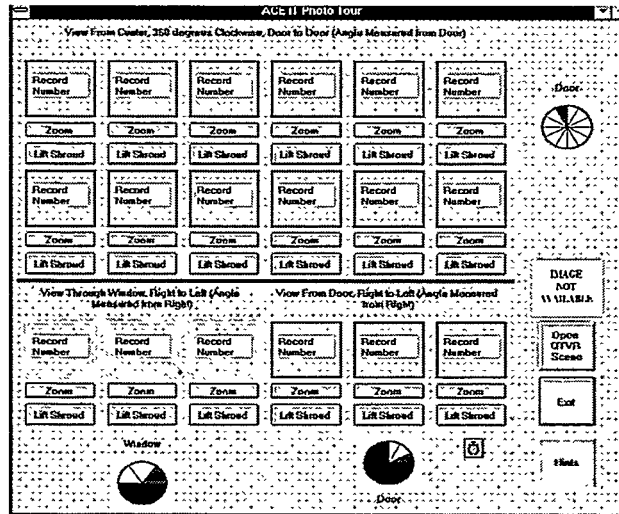
Form OurChat.Frm

| Control | Function |
|-------------|---|
| bhangup | Control that disconnects a running Chat Session. |
| HangupTimer | Timer control that checks to see if other user has disconnected. |
| SendButton | Control that sends the contents of the user's text box to the other user. |
| TmrRead | Timer control that checks for text from other user. |
| tmrRing | Timer control that makes the Chat feature ring when activated. |



Form PartAcc.Frm

| Control | Function |
|------------------|---|
| Check3d1 | Control array of check boxes that displays the selected levels of partial access. |
| JustificationBox | Text box control that displays and stores the associated justification for providing only partial access. |
| ResponseButton | Control array that either sends the partial access response to the Inspection Team, unloads the form without sending a response, or loads the Hint3.Frm Form. |



Form Photour.Frm

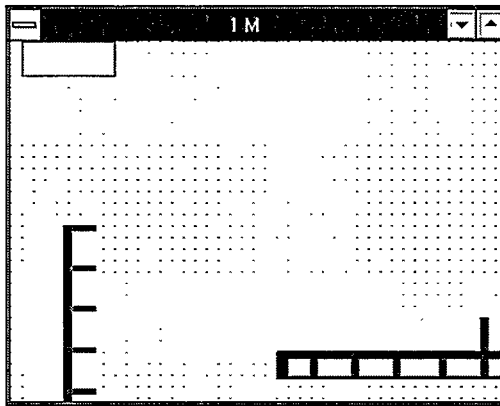
| Control | Function |
|---------------|---|
| CenterOutLine | Shape control that indicates the image currently in focus for the view from the center of the room. |
| CenterImage | Image control that displays the images for the view from the center of the room. |
| LiftShroud1 | Control that loads either the MultForm.frm form or the Lift.Frm form, enabling the Inspection Team to request that the shroud be lifted from a shrouded item. |
| CenterZoom | Control that loads the ZoomForm with the selected image and the images to each side of the selected image to full screen display. |
| CenterView | Pie chart that indicates the direction of view relative to the doorway from the center of the room. |
| WindowOutLine | Shape control that indicates the image currently in focus for the view through the window, if available. |
| WindowImage | Image control that displays the images for the view through the window of the room. |
| LiftShroud2 | Control that loads either the MultForm.frm form or the Lift.Frm form, enabling the Inspection Team to request that the shroud be lifted from a shrouded item. |
| WindowZoom | Control that loads the ZoomForm with the selected image and the images to each side of the selected image to full screen display. |
| WindowView | Pie chart that indicates the direction of view relative to the right-hand side of the available window view. |
| DoorOutLine | Shape control that indicates the image currently in focus for the view through the door of the room. |
| DoorImage | Image control that displays the images for the view through the door. |
| LiftShroud3 | Control that loads either the MultForm.frm form or the Lift.Frm form, enabling the Inspection Team to request that the shroud be lifted from a shrouded item. |
| DoorZoom | Control that loads the ZoomForm with the selected image and the images on each side of the selected image to full screen display. |
| DoorRight | Pie chart that displays the angle of view through the door if the door is located on the right-hand side of the room. |
| DoorLeft | Pie chart that displays the angle of view through the door if the door is located on the left-hand side of the room. |

Form Result.Frm

| Control | Function |
|----------|---|
| Grid1 | Grid Control that displays sampling results for the last recorded sample. |
| Command2 | Control that unloads the form. |

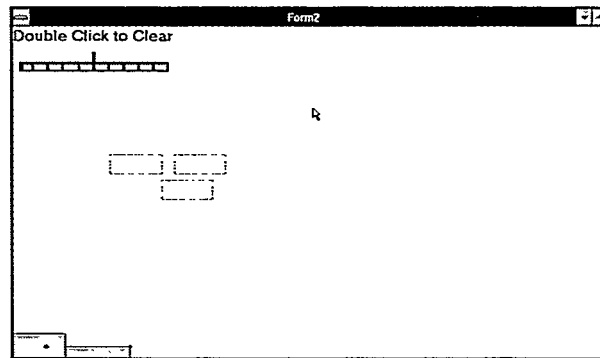
Form Rooms.Frm

| Control | Function |
|------------|---|
| Grid1 | Grid control that displays the results of the query for the selected location. |
| Command3d1 | Control array that allows the user to filter a selected field, return to the original record set, sort by record number or by a selected field, or that unloads the form. |



Form YrdStk.Frm

| Control | Function |
|------------|--|
| meterstick | Multimedia control that displays the image of the on-screen meter stick. |
| stickhorz | image box control that stores the image of the horizontal meter stick. |
| stickvert | image box control that stores the image of the vertical meter stick. |



Form ZoomForm.Frm

| Control | Function |
|------------|---|
| measure | Image control used to activate the on-screen meter stick. |
| ZoomImage | Image control that displays selected image. (Size and position is controlled by code at run time.) |
| ZoomImage2 | Image control that displays image to the left of the selected image. (Size and position is controlled by code at run time.) |
| ZoomImage3 | Image control that displays image to the right of the selected image. (Size and position are controlled by code at run time.) |

5.2.2 The Inspection Team Executable (NEG_I.EXE)

The AutoCAD®/ArcCAD® GIS system map for the CMC is automatically activated by OSI.EXE when the scenario timeline reaches the Inspection event.

The NEG_I.EXE runs in the background from the time it is started by the OSI.EXE until it is shut down by the user manually at the end of the exercise. It uses a combination of text files and database records to communicate with the moderator executable file (NEG_M.EXE).

AutoCAD® is used in this system to provide visual representations of locations. ArcCAD® is used to construct GIS coverages of rooms identified by room number. The Inspection Team selects BEGIN INSPECTION from the AutoCAD® dropdown menu and then clicks on a room on the map. This initiates a request for access to that room. SLEEPER.EXE receives the unique identification for this room, and writes it to a text file on the local hard drive. SLEEPER.EXE terminates on completion of this action.

NEG_I.EXE uses a Visual BASIC® timer control to check at regular intervals for the creation of a text file by the SLEEPER.EXE executable file. Once this file is created, the timer control activates a subroutine that indicates that a request is being processed. The location information provided by the GIS via SLEEPER.EXE is extracted by the NEG_I.EXE, and the text file is deleted. The location information then generates an “Access Request”, which uses a combination of database records and text files to pass on the details of this request, including the following:

1. The scenario time and date from the moderator’s OSI.EXE.
2. The exercise time remaining in minutes.
3. The location requested.

By default, all requests are assumed to be for “Full Access” to the specified location. Each time a request is initiated, NEG_I.EXE compares the requested location to the contents of a database table (residing in Comm.mdb) to see if a request for access to this location has already been processed. If so, then the user is warned that they have already made at least one request for access to this location. The user then has the option of canceling this request, or continuing with a repeat request. **The system does not prevent the Inspection Team from requesting access to a given location as many times as they feel it is necessary; it only provides them with an indication that a specific location may have already been visited.** If the location has not been previously requested, the system adds this location to the list of previously accessed locations so that subsequent requests can be screened.

Once a physical access request has been screened, the system goes into a “wait” mode. The Moderator Computer receives the request, logs it, and then forwards it on to the Inspected Party. NEG_I.EXE uses Visual BASIC® timers to check for the creation of appropriate “response” files or database records on the Moderator’s computer to

Technical Manual for the ACE-IT Software

determine whether a response has been received. Once a response has been detected, the system screens it, and, based on the type of response, behaves in one of three possible modes. These modes are as follows:

1. Physical Access Request “Denied” - The Inspection Team is denied physical access to the location. Requests for alternate information can be initiated by the Inspection Team, or the request for physical access can be repeated, with further justification for why the Inspection Team requires physical access to this location.
2. Physical Access Request “Partial” - The Inspection Team is granted some degree of access to this location. For the purposes of this exercise, Partial Access is defined as allowing the Inspection Team to view the location, through the window and/or door, without physically entering. Again, the Inspection Team can repeat the access request, request alternate information, or accept partial access. Permission to lift the shrouds from shrouded items, take physical samples, or view alternate data is not implied. Access to these items is negotiated separately for each location and/or item. If partial access is insufficient for the needs of the Inspection Team, the access request should be repeated, providing further justification for requiring access to the location, and the level of access can then be renegotiated.
3. Physical Access Request “Granted” - The Inspection Team is granted full physical access to the location. For the purposes of this exercise, Full Physical Access is defined as permission for the Inspection Team to enter the location, and to view it from all available angles. Permission to lift the shrouds from shrouded items, take physical samples, or view alternate data is not implied. Access to these items is negotiated separately for each location and/or item. Once an access request has been “granted,” the Inspection Team can either accept this level of access or request alternate information. There is no repeat request capability. If a QuickTime® Virtual Reality Scene is available for the location, it will be accessible from the full access screen.

Requests for access to alternate data and requests to lift a shroud from a shrouded item are processed only after a request for physical access has been processed and a response has been received. Alternate data can always be requested by the Inspection Team in inspection mode. Once a response has been received, the different alternate data options will vary according to the location, and the available topics will vary according to the actions of the Inspected Party. The Alternate Data Request Form first screens the form to determine which topics of Alternate Data are being requested, then writes a text file to the Moderator computer that consists of an ASCII string made of unique string values for each topic of Alternate Data included in the request. Therefore, every possible combination of requests creates a unique string, which is parsed to determine which topics are being requested. Once a request has been sent, the system goes into a wait mode. The request is logged on the Moderator’s computer and forwards it to the Inspected Party. NEG_I.EXE then uses a timer control to detect the creation of a text file

on the Moderator computer, which indicates that a response has been received. This text file is parsed into the unique values indicating which topics of Alternate Data the Inspected Party is allowing access to.

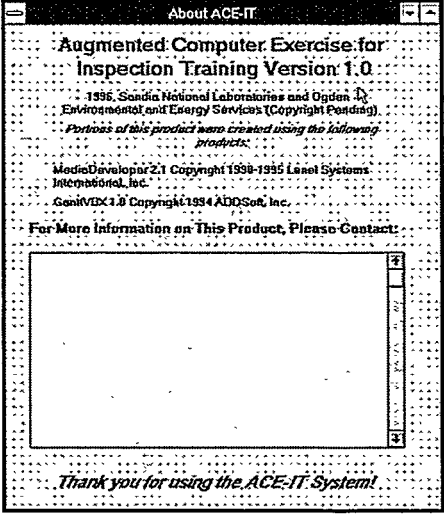
A request to lift the shroud from a shrouded item can only be processed if that item is visible in one or more of the available physical access views. Each item must be requested separately. When a request is sent, NEG_I.EXE writes a combination of text files and database records to send the complete request to the Moderator computer and goes into a wait mode. The request is logged by the Moderator's Computer, and forwarded to the Inspected Party. NEG_I.EXE uses a timer control to check for the creation of a combination of text files and database records on the Moderator's computer that indicate a response. The information included in these files includes the following: the percentage of shroud which will be lifted for a specific item; any comments from the Inspected Party; and additional data on the item if the percentage of shroud lift is equal to 100%.

The NEG_I.EXE also has an internal Chat feature, which is activated by clicking on the CALL ANOTHER PLAYER button located on all HINTS forms. This chat feature utilizes internal timer controls to detect calls for connections, display/send text, and disconnect messages. All text that is sent using the chat feature is saved to the exercise log. Text that is not sent through the chat feature is not saved, and the system does not mirror keystrokes like the Windows® chat function. NEG_H.EXE, NEG_I.EXE, and NEG_M.EXE must all be running in order for the chat feature to operate.

Technical Manual for the ACE-IT Software

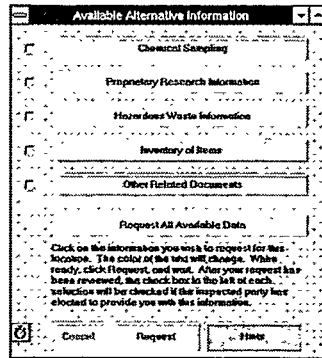
5.2.2.1 NEG_I.EXE Visual Basic® Forms

The following forms make up the source code for NEG_I.EXE. They are provided in alphabetical order, with a table listing the major functions available on each form. When used in conjunction with Visual BASIC® in the development, the specific controls referenced for each form can be identified and the source code for each function accessed.



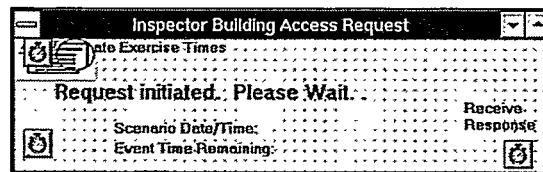
Form About.Frm

| Control | Function |
|---------|---|
| Text1 | Text Box that displays contact points for ACE-IT system. Multiline and scroll bar properties have been set. |



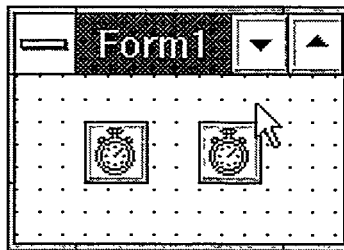
Form AltData.Frm

| Control | Function |
|-------------|---|
| Description | Control array that either indicates the topics of alternate data being requested, by a change in text color, or accesses the appropriate database queries and forms to load the available information for the selected topic. |
| Include | Control array of check boxes that indicate the topics of alternate information to which access will be granted. Not enabled, state is modified by code only. |
| Timer1 | Timer control that puts system into wait mode, and checks for the creation of the text file dataresp.ACE on the Moderator computer to continue the operation of the system. |
| Cancel | Control that unloads the form, without sending a request. |
| Return | Control that determines which topics of alternate information are being requested, creates a request, and then puts the system into wait mode. |
| Hints | Loads the Hint2.Frm form. |



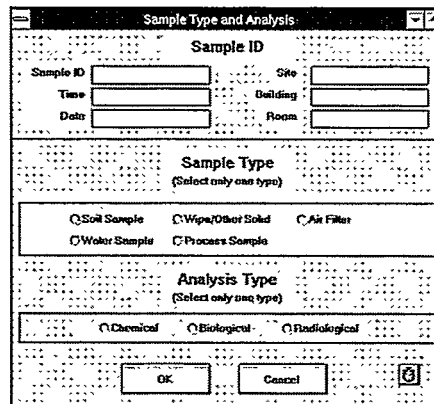
Form BldgNegI.Frm

| Control | Function |
|---------|--|
| Timer3 | Timer control that gets the current exercise times from the Moderator Computer. |
| Timer1 | Timer control that provides the animation to the Request Initiated label. |
| Timer2 | Timer control that checks for the creation of the response in the form of a database file created on the Moderator computer. When response is received, loads Response.Frm form. |



Form Chatchek.Frm

| Control | Function |
|--------------|--|
| ChatCheker | Timer Control that checks for activation of the Chat feature. |
| AnswerCheker | Timer Control that checks for the other user to answer the Chat ring before activating the Chat Feature. |



Form ChemSam.Frm

| Control | Function |
|--------------|--|
| Option3d2 | Control array of radio buttons that indicate what type of sample will be recorded. |
| Option3d1 | Control array of radio buttons that indicate the type of analysis to be performed on the sample recorded. |
| Timer1 | Timer control that forces a delay between the recording of the sample and display of results. (1 minute) |
| SampleOK | Control that records the sample and activates the Timer1 timer control. Hides the current form, but leaves it running. |
| SampleCancel | Control that unloads the form without recording a sample. |

Form Crada_1.Frm

| Control | Function |
|--------------------|---|
| MoveFirstButton | Control that moves selected recordset to the first available record. |
| MoveNextButton | Control that moves the selected recordset to the next available record. |
| MovePreviousButton | Control that moves the selected recordset to the previous available record. |
| MoveLastButton | Control that moves the selected recordset to the last available record. |
| Command1 | Control that unloads the form. |

Form Haz_W.Frm

| Control | Function |
|--------------------|---|
| MoveFirstButton | Control that moves selected recordset to the first available record. |
| MoveNextButton | Control that moves the selected recordset to the next available record. |
| MovePreviousButton | Control that moves the selected recordset to the previous available record. |
| MoveLastButton | Control that moves the selected recordset to the last available record. |
| Command2 | Control that unloads the form. |

Form Hint1.Frm

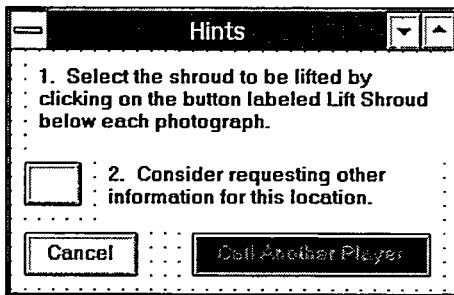
Form Hint2.Frm

| Control | Function |
|--------------|--|
| AltInfo | Loads the Alternate Information Request Form. |
| CancelButton | Unloads the form without taking any other actions. |
| Panel3d1 | Control that loads the OurChat.frm to activate the Chat feature. |

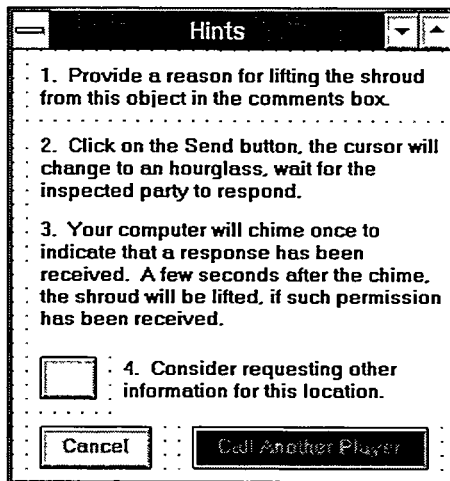
Form Hint3.Frm

Form Hint4.Frm

| Control | Function |
|--------------|--|
| AltInfo | Loads the Alternate Information Request Form. |
| CancelButton | Unloads the form without taking any other actions. |
| Panel3d1 | Control that loads the OurChat.frm to activate the Chat feature. |

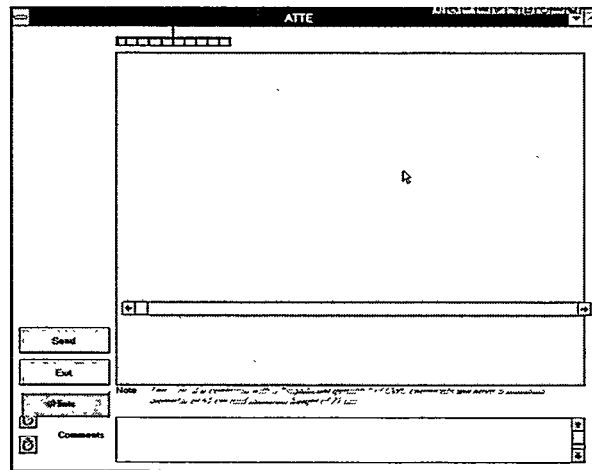


Form Hint5.Frm



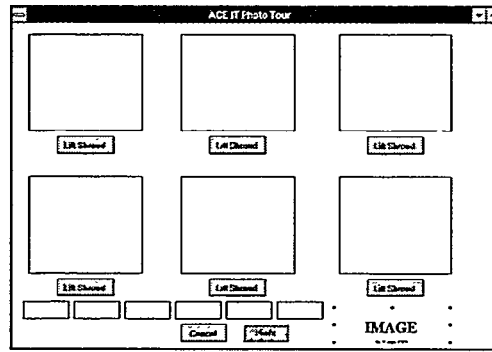
Form Hint6.Frm

| Control | Function |
|--------------|--|
| AltInfo | Loads the Alternate Information Request Form. |
| CancelButton | Unloads the form without taking any other actions. |
| Panel3d1 | Control that loads the OurChat.frm to activate the Chat feature. |



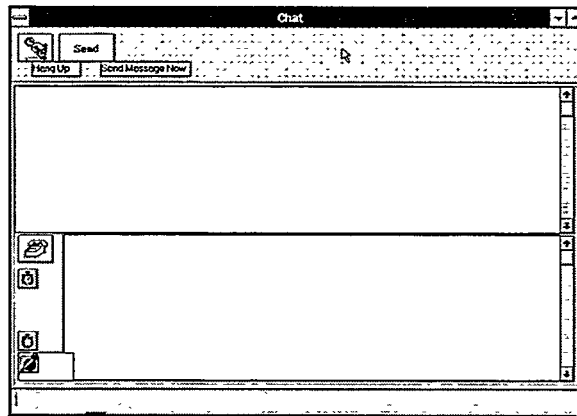
Form Lift.Frm

| Control | Function |
|------------|--|
| Text1 | Text box control that displays the available data for the shrouded item only if lift of the shroud is granted to the 100% level. |
| Command2 | Sends the request to lift the shroud from an item and activates the Timer1 timer control, putting the system into a wait mode. |
| Command1 | Unloads the form without taking any other actions. |
| Hintbutton | Loads the Hint6.Frm form. |
| Text2 | Displays and stores comments associated with the current request. |
| measure | Image box used to activate the on-screen meter stick. |
| MediaDev1 | Displays the currently selected shrouded item. |
| Timer1 | Timer control that puts the system into a wait mode and checks for the creation of a response on the Moderator Computer. Once a response is received, activates the Timer2 timer control to introduce a delay. |
| Timer2 | Timer control that introduces a delay between the receipt of a response and the carrying out of appropriate actions automatically. |



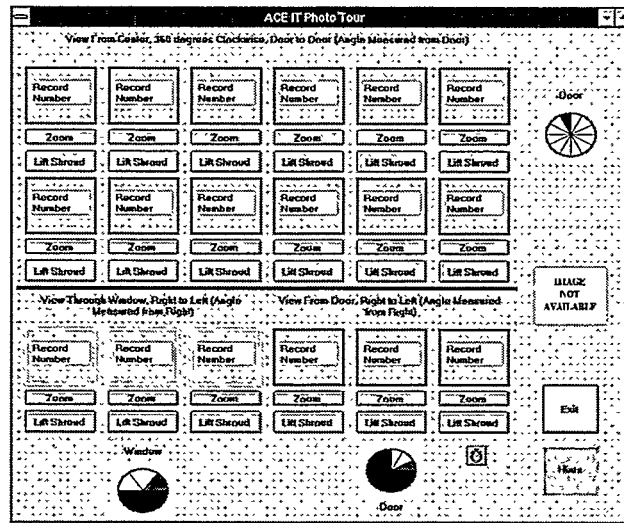
Form MultForm.Frm

| Control | Function |
|----------|---|
| VidImage | Control array that displays the starting frame of each video file. |
| Lifter | Control array that loads the Lift.Frm form with the selected video file and associated information. |
| Command1 | Control that unloads the form. |



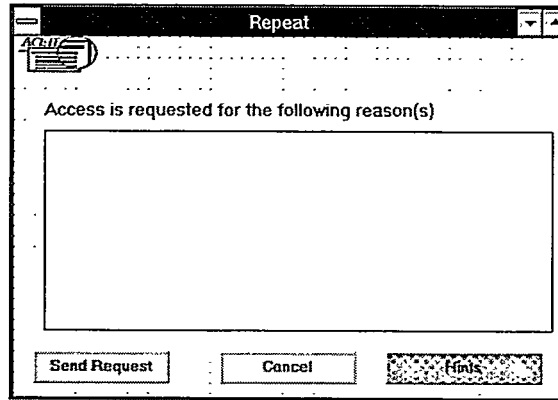
Form OurChat.Frm

| Control | Function |
|-------------|---|
| bhangup | Control that disconnects a running Chat Session. |
| HangupTimer | Timer control that checks to see other user has disconnected. |
| SendButton | Control that sends the contents of the user's text box to the other user. |
| TmrRead | Timer control that checks for text from other user. |
| tmrRing | Timer control that makes the Chat feature ring when activated. |



Form Photour.Frm

| Control | Function |
|---------------|---|
| CenterOutline | Shape control that indicates the image currently in focus for the view from the center of the room. |
| CenterImage | Image control that displays the images for the view from the center of the room. |
| LiftShroud1 | Control that loads either the MultForm.frm form or the Lift.Frm form, enabling the Inspection Team to request that the shroud be lifted from a shrouded item. |
| CenterZoom | Control that loads the ZoomForm with the selected image and the images to each side of the selected image to full screen display. |
| CenterView | Pie chart that indicates the direction of view relative to the doorway from the center of the room. |
| WindowOutline | Shape control that indicates the image currently in focus for the view through the window if available. |
| WindowImage | Image control that displays the images for the view through the window of the room. |
| LiftShroud2 | Control that loads either the MultForm.frm form or the Lift.Frm form, enabling the Inspection Team to request that the shroud be lifted from a shrouded item. |
| WindowZoom | Control that loads the ZoomForm with the selected image and the images to each side of the selected image to full screen display. |
| WindowView | Pie chart that indicates the direction of view relative to the right-hand side of the available window view. |
| DoorOutline | Shape control that indicates the image currently in focus for the view through the door of the room. |
| DoorImage | Image control that displays the images for the view through the door. |
| LiftShroud3 | Control that loads either the MultForm.frm form or the Lift.Frm form, enabling the Inspection Team to request that the shroud be lifted from a shrouded item. |
| DoorZoom | Control that loads the ZoomForm with the selected image and the images to each side of the selected image to full screen display. |
| DoorRight | Pie chart that displays the angle of view through the door if the door is located on the right-hand side of the room. |
| DoorLeft | Pie chart that displays the angle of view through the door, if the door is located on the left-hand side of the room. |



Form Repeat.Frm

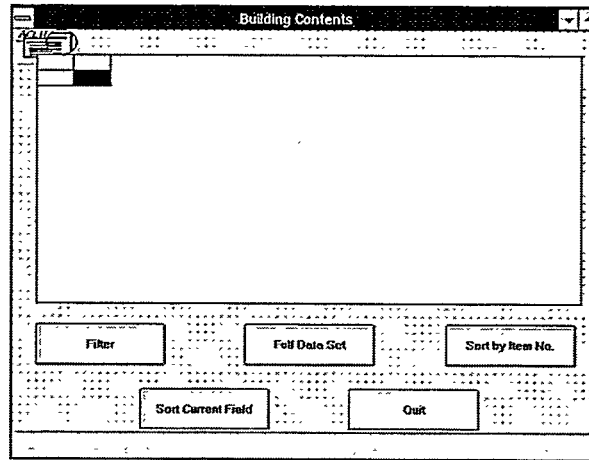
| Control | Function |
|------------|---|
| Text1 | Text box control that displays and stores the Inspection Team's justification for requesting greater access to a specific location. |
| Continue | Control array of two command buttons: Send Request and Cancel. Send Request loads the BldgNeg.I.Frm form. Cancel aborts the request for access, unloads the form, and returns to the Response.Frm Form. |
| Hintbutton | Loads the Hint3.Frm form. |

Form Response.Frm

| Control | Function |
|----------|--|
| Label2 | Label control that displays the status of the current response. Color is controlled by code: red for access denied, yellow for some level of partial access, and green for access granted. |
| Text2 | Text box control that displays the Inspected Party's justification for either denying access or providing only partial access. This control is not visible if access is granted. |
| Continue | Control array that either loads the photour for the current level of access, loads the Repeat.frm form so that a request may be repeated, or loads the Hint1.Frm form. |

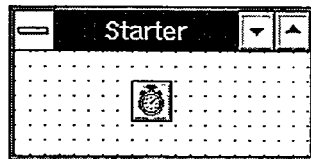
Form Result.Frm

| Control | Function |
|----------|---|
| Grid1 | Grid control that displays sampling results for the last recorded sample. |
| Command2 | Control that unloads the form. |



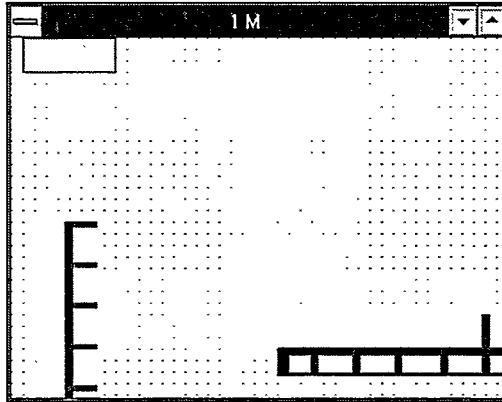
Form Rooms.Frm

| Control | Function |
|------------|---|
| Grid1 | Grid control that displays the results of the query for the selected location. |
| Command3d1 | Control array that allows the user to filter a selected field, return to the original record set, sort by record number or by a selected field, or unload the form. |



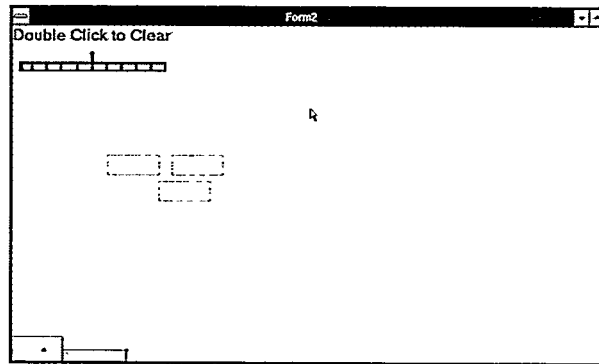
Form Starter.Frm

| Control | Function |
|---------|--|
| Timer1 | Timer control that checks for the creation of the text file sleeper.ACE on the Inspection Team computer. When this file is created, it parses the contents and initiates a request for the specific location. It then deletes the file sleeper.ACE so that the next request can be properly handled. |



Form YrdStk.Frm

| Control | Function |
|------------|--|
| meterstick | Multimedia control that displays the image of the on-screen meter stick. |
| stickhorz | Image box control that stores the image of the horizontal meter stick. |
| stickvert | Image box control that stores the image of the vertical meter stick. |



Form ZoomForm.Frm

| Control | Function |
|----------------|---|
| measure | Image control used to activate the on-screen meter stick. |
| ZoomImage | Image control that displays selected image. (Size and position is controlled by code at run time.) |
| ZoomImage2 | Image control that displays image to the left of the selected image. (Size and position is controlled by code at run time.) |
| ZoomImage3 | Image control that displays image to the right of the selected image. (Size and position are controlled by code at run time.) |

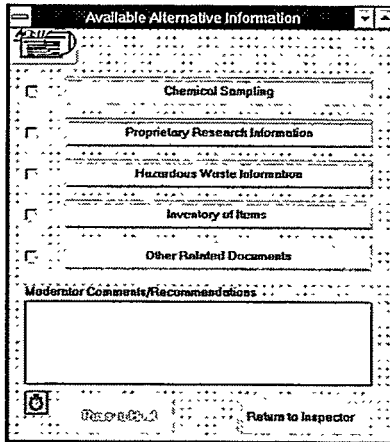
5.2.3 The Moderator Executable (NEG_M.EXE)

The Moderator serves as the communications hub of the *ACE-IT* system. It interacts with each of the other two computers through the creation and deletion of text files and database records. Visual BASIC® controls, known as timers, provide this functionality. All requests and responses pass through the Moderator executable, where they are logged before being automatically passed on to the appropriate player. NEG_M.EXE is a completely automated module; no interaction is required or expected from the Moderator computer. All requests and corresponding responses are recorded by the NEG_M.EXE file as it processes each in sequence. To pause, stop, or “jump” the exercise, the moderator must return to the OSI.EXE clock form.

A pair of timer controls functions with NEG_M.EXE. Prior to receiving the first request from the Inspection Team during the Inspection event, NEG_M.EXE runs in background. Once the initial request is received, NEG_M.EXE will show the status of the current request/response pairing. The two timer controls are turned ON and OFF to allow the system to wait for either a request or a response. They are never both ON or both OFF at the same time. NEG_M.EXE is always waiting for either a request or a response. The Moderator is provided with a visual indication of current system processing by the behavior and captions of each form. For example, on the initiation of a request from the Inspection Team for access to Room X, the Moderator will see the caption “**Request for Access to Location X**”. Once a request has been initiated by the Inspection Team, no other requests can be initiated until a response has been received from the Inspected Party. Similarly, once a response has been received, no other response can be initiated until the next request has been received by the Inspected Party. For each request/response, the Moderator is provided with the OSI.EXE Scenario Time, along with the exercise time remaining in the Inspection Event. This gives the Moderator a sense of “time pressure.”

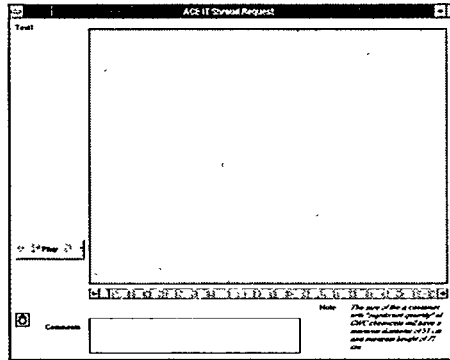
5.2.3.1 NEG_M.EXE Visual Basic® Forms

The following forms make up the source code for NEG_M.EXE. They are provided in alphabetical order, with a table listing the major functions available on each form. When used in conjunction with Visual BASIC® in the development, the specific controls referenced for each form can be identified, and the source code for each function accessed.



Form AltData.Frm

| Control | Function |
|---------|--|
| Timer1 | Checks for the creation of the text file HostData.ACE on the Moderator computer. When this file is detected, the timer control parses the file contents, deletes the file, then logs the response, and passes to the Inspection Team in the form of the text file dataresp.ACE created on the Moderator computer. It then returns control of the system to the Timer1 timer control on the Mod1frm form. |



Form Lift.Frm

| Control | Function |
|---------|---|
| Timer1 | Checks for a combination of text files and database records on the Moderator computer indicating an Inspected Party response to a request to lift a shroud from an item. When this information is detected, it is parsed and then deleted. The response is passed back to the Inspection Team and control of the system is returned to the Timer1 timer control on the Mod1.Frm form. |

Form Mod1.Frm

| Control | Function |
|---------|--|
| Timer1 | Checks for the creation of various text files or database records by the Inspection Team executable file, and then parses the request to determine the type of request. The request is then deleted and passed on to the Inspected Party. Control of the system is passed to the wait timers, Timer1 on the AltData.Frm, Timer1 on the Lift.Frm, and Timer2 on the Mod1.Frm, as appropriate. |
| Timer2 | In the case of a request for physical access, this timer control checks for the creation of a database record on the Moderator computer. When this record is detected, the timer parses the data, deletes the record, creates a new record for use by the Inspection Team executable file, and returns control of the system to the Timer1 timer control on the Mod1.Frm form. |

5.3 OSI.EXE (THE ACE-IT CONTROL MODULE)

The *ACE-IT* Control Module is actually three separate executables, one on each computer. The executables all contain the same single module of code. However, the Inspected Party and the Inspection Team computers do not access all of the functions built into the module of code. All three executables provide the user with the following:

1. A visual representation (in the form of a Gantt chart) of the inspection timeline outlined in the CWC challenge inspection regime;
2. The identification of the current event and the current exercise time remaining;
3. The capability to enter notes and view external documents (OLE) in the form of Windows® Write files.

The Moderator executable provides the user with the following additional functionality:

1. The ability to start, stop, pause, and jump the exercise forward on the inspection timeline on all three computers;
2. Display boxes that indicate the current scenario date/time, and the total exercise time remaining;
3. A tracking module that follows all the exercise events (notes, event starts, and request) as they occur.

The OSI.EXE file is completely “data driven,” that is, all of its behavior is determined by the contents of an underlying database. OSI.EXE uses the Visual Basic® add-on component, Gantt.VBX, to display a Gantt chart timeline of the CWC challenge inspection schedule. This Gantt chart is loaded from a query stored in the SCENARIO.MDB database file that is specific to the selected scenario. The exercise schedule, which allots blocks of time for training on each step of a CWC challenge inspection, is also loaded as a part of this same query. Timer controls are used to provide most of the OSI.EXE functionality. A separate timer control with a constant one-second interval is used to keep track of the total time elapsed as the exercise progresses. At whole minute intervals, the system checks to see if a new event is scheduled to start. When this occurs, the OSI.EXE moves the system forward one step in the selected schedule and uses an Object Linked or Embedded (OLE) control on the form to load a specific file and make it available to the user. For all but the Inspection step of the exercise, this file is a Windows® Write document stored in the Notes subdirectory of the **C:\inspectr** install directory. The Inspection step loads the AutoCAD® map of the CMC site, and it also loads ArcCAD® on the Inspection Team computer.

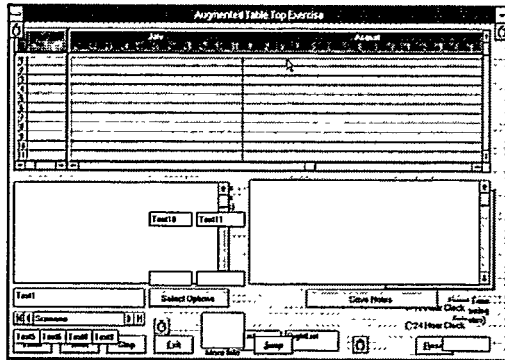
The Moderator computer can use the OSI.EXE file to Start the exercise, Pause/Resume the exercise, or Jump the exercise forward to the next step, if appropriate. This is accomplished through the creation of text files on both the Inspected Party computer and the Inspection Team computer. Timer controls on both of these computers check for the creation of these text files and, depending on their contents, either Start, Pause, Resume or Jump the exercise.

OSI.EXE has two initialization files associated with it. These files are unique to each computer in the *ACE-IT* setup, and are not interchangeable. The first file is *install.ini*, and resides in the *C:\INSPECTR\BIN* subdirectory. The contents of this file indicate only the installation directory of *ACE-IT* on each computer. Although it is not practical to move the GIS coverages created by ArcCAD® without extensive system manipulation, it is possible that the *ACE-IT* system could be installed in a different directory. The second initialization file resides in the *C:\INSPECTR\INI* subdirectory, and allows modification of the following parameters:

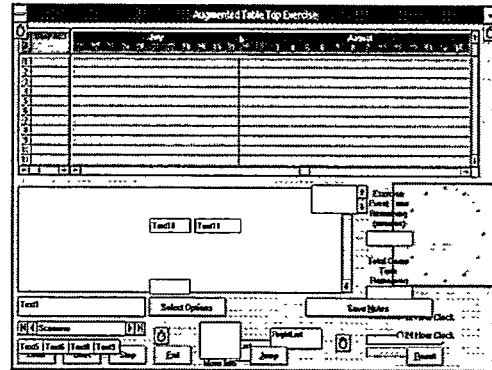
1. The name of scenario database file that drives the OSI.EXE file, currently *SCENARIO.MDB*;
2. The name of the communications database that drives the negotiation modules of *ACE-IT*, currently *COMM.MDB*; and
3. The network drive connections between all three computers. Currently, the Moderator is *M:*, the Inspected Party is *H:*, and the Inspection Team Computer is *I:*.

5.3.1 OSI.EXE Visual Basic® Forms

OSI.EXE consists of two forms, presented here in alphabetical order, with a table outlining the major controls and functions of each. There are two separate versions of the OSI executable file, but they differ only in the layout of the form. The form OSI.FRM is used to create the executable files for the Inspection Team and Inspected Party, and the form file MOD_OSI.FRM is used to create the executable for the Moderator.

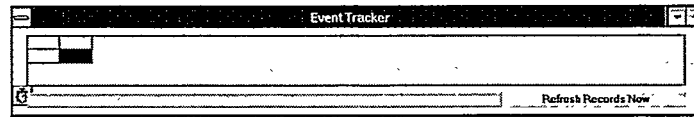


Form OSI.Frm



Form Mod_Osi.frm

| Control | Function |
|--------------|---|
| StartChecker | Timer control that checks for creation of a text file on the local drive that indicates the status of the exercise, start, run or pause. When this file is detected, its contents are read by the system and appropriate actions carried out. |
| GanttVBX1 | GanttVBX control that displays the Gantt Chart loaded from the database Scenario.Mdb. |
| Data1 | Data control that determines the record set that will identify the selected scenario in Scenario.mdb. |
| Text1 | Displays the scenario selected by the Data1 data control. |
| Text2 | Text box control that displays the detailed description of the current event from the GanttVBX1 GanttVBX control. |
| Text3 | Text box control that displays notes taken by the user during run time. |
| Text4 | Text box control that displays the elapsed time in the form hh:mm:ss for the current exercise. |
| Text5 | Text box control, not visible to the user at run time, that is used to determine the total number of minutes to be displayed in the Text4 text box control. |
| Text6 | Text box control, not visible to the user at run time, that is used to determine the total number of seconds to be displayed in the Text4 text box control. |
| Text7 | Text box control used to display the number of minutes remaining in the current exercise step. |
| Text10 | Text box control used to display the number of hours required by the scenario in real time for the current exercise step. |
| Text11 | Text box control used to display the number of minutes allotted to the trainees for the current exercise step. |
| LeftList | List box control used to display multiple choices available during a specific exercise step. |
| RightList | List box control used to display the selected choices made during a specific exercise step. |
| OLE1 | OLE control used to link objects to a specific exercise step. These objects include Windows Write (.WRI) format documents, and AutoCAD drawings. |
| StartStop | Control array that is used to Load, Start, Stop, Exit, and Jump the system. Only the Moderator computer has access to the Start, Stop, and Jump functions. |
| Timer1 | Timer control that is used to control the behavior of the analog clock display. The interval is varied as a function of real time required and exercise time allotted so that real time displayed will pass in the allotted exercise time. |
| TimerKeeper | Timer control that keeps track of elapsed time, and checks to see if the system schedule requires moving to the next event. |
| Image1 | Image control used to contain the various parts that make up the analog clock. |
| Command3d1 | Control that allows the user to reset the start time of the scheduled exercise to simulate any desired time. |



Form Tracker.Frm (Visible only to Moderator computer)

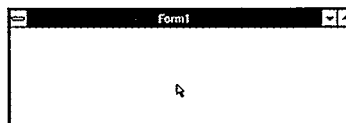
| Control | Function |
|---------------|---|
| Grid1 | Grid control that displays the current exercise record. |
| Timer1 | Timer control that runs a query at 30 second intervals to refresh the contents of the grid1 grid control. |
| RefreshButton | Control that can be used at any time to refresh the contents of the grid1 grid control. |

5.4 RECALL.EXE

The RECALL.EXE file resides on the Inspected Party computer. It creates a text file that is detected by NEG_H.EXE file, which activates the recall function.

5.4.1 RECALL.EXE Visual Basic® Form

RECALL.EXE creates a text file (RECALL.ACE). RECALL.EXE then automatically terminates. The RECALL.ACE file is read by NEG_H.EXE and then deleted. The user will never see this form during system operation.



Form Recall.FRM

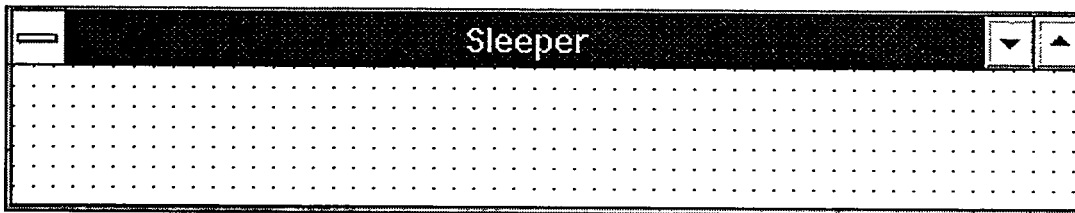
| Control | Function |
|---------|---|
| None | On the load event, RECALL.EXE creates the text file RECALL.ACE on the Inspected Party computer and then automatically terminates. This file is detected by the NEG_H.EXE. |

5.5 SLEEPER.EXE

The SLEEPER.EXE file is the only file in the ACE-IT system that is used by the GIS interface, and it is only used by the Inspection Team computer. SLEEPER.EXE is called by the ArcCAD® xLoad function when the user clicks on a room, with the command line specific to the location selected. When SLEEPER.EXE is activated, it immediately writes a text file with the contents of the AcadText field of the PAT.dbf file for the map currently displayed. SLEEPER.EXE then terminates. If multiple requests are initiated before the system can respond, then this text file is automatically overwritten, and only the last request is processed.

5.5.1 SLEEPER.EXE Visual Basic® Form

SLEEPER.EXE creates a text file (SLEEPER.ACE). SLEEPER.EXE then automatically terminates. The SLEEPER.ACE file is read by NEG_I.EXE then deleted.



Form Sleeper.Frm

| Control | Function |
|---------|--|
| None | On the load event, SLEEPER.EXE creates the text file sleeper.ACE on the Inspection Team computer. This file is detected by the NEG_I.EXE Starter.Frm form and then automatically terminates. |

6.0 THE ACE-IT GIS INTERFACE

The GIS interface is specific to only the Inspection Team computer. ArcCAD[®] is automatically loaded when the AutoCAD[®] map of the CMC site is loaded. The user first selects the desired view of the CMC Complex using the VIEW menu in the AutoCAD[®] Window. The user next selects the BEGIN NEGOTIATED INSPECTION selection from the pull-down Inspection menu. The user then clicks on a location to begin a negotiation for access to that location. The GIS returns the unique identification of the location stored in the AcadText field of the PAT.dbf file for the location displayed and activates SLEEPER.EXE.

6.1 MENU STRUCTURE

When the ACE-IT system is first activated using the Windows[®] icon in the ACE-IT Group within the Windows[®] Program Manager, the home directory is set by the group properties to be C:\INSPECTRMENU. AutoCAD[®] is then loaded, followed by ArcCAD[®]. When ArcCAD[®] loads, it automatically calls and loads its own customized menu, ARCAD12.MNU. This procedure is hardcoded into the ArcCAD[®] software and happens automatically every time ArcCAD[®] loads. The menu that the ACE-IT system uses is heavily customized and therefore must be loaded automatically upon startup.

The pull-down menu in the AutoCAD[®] window is modifiable according to the rules stated in the AutoCAD[®] documentation. Customizations are effected either through a call out of a direct AutoCAD[®] function via AutoLISP[®], a direct AutoCAD[®] function, or a call to a custom AutoLISP[®] function defined in GIS.LSP (which is also loaded upon startup and is designed to work hand-in-hand with the custom menu).

6.2 SITE MAPS

The CMC site map included in ACE-IT was provided by the Sandia National Laboratories Facilities Organization. The site map used in the training tool is an AutoCAD[®] drawing file, CMC.DWG. This drawing is linked to the Geographic Information System (GIS) software ArcCAD[®] via the GIS software. This coverage is made through only two of the AutoCAD[®] drawing layers. These layers are ACAD_TEXT and ROOMS. The ACAD_TEXT layer contains a unique text string identifying the room number in the building. The ROOMS layer contains the location information for the closed polygons that make up the room outline on the map. Essentially, the GIS returns the following information: "The area enclosed in coordinates X1 = 776, Y1 = 445, X2 = 998, Y2 = 576 is room number 510-1-103." SLEEPER.EXE uses this information to activate the ACE-IT system and initiate negotiation for access to the room requested.

The GIS software is important to the overall system because it allows the user to point and click on a room within the map to initiate access requests. The system then shows the correct information for that location.

6.3 LAYERS USED IN CMC.DWG

AutoCAD® drawings are arranged in layers, with different items or entities having common features grouped together on a layer that is given a number or name. One can think of AutoCAD's® layers as a series of tracing paper drawings that are stacked one upon another. Each of these layers contains a different piece of the whole drawing on an invisible sheet of electronic tracing paper. By careful organization and manipulation of these individual drawings, the amount of information presented to the user can be precisely controlled. The following table presents the GIS layers used by ACE-IT and their content or use.

The layers and their use by the system software:

| CMC Drawing Layer | Use |
|-----------------------|--|
| 0 | Layer "0" is reserved by AutoCAD®. |
| ESRI_THEMES | Used by ArcCAD® to establish a zero reference point in X, Y space on the map. |
| QUERY_SCRATCH | A scratch pad layer where the system software can draw highlights and cross hatching, which can later be erased. |
| ACAD_TEXT | Unique text string used as a room number identifier by the system. |
| ROOMS | Closed polygon room outlines used in conjunction with the ACAD_TEXT layer to marry the room number to the room location in the database. |
| 500_BLDG_NO | Building 10500's building number (10500). |
| 500_BLDG_OUTLINE | Footprint for building 10500. |
| 500_DOORS | The blue line representing the door location and width for building 10500. |
| 500_EVACUATION | The red arrows indicating the evacuation routes for building 10500. |
| 500_FURNITURE | Furniture for building 10500. |
| 500_INTERIOR | The interior walls, doorswings, and interior details of building 10500. |
| 500_PLUMBING_FIXTURES | The restroom details for building 10500. |
| 500_RM_PEOPLE_TEXT | The human-readable room numbers of building 10500. |
| 500_SIDEWALK | The sidewalks surrounding building 10500. |
| 500_VENT | The vents on the roof of building 10500. |
| 500_WINDOWS | The red line indicating the interior window location and widths of building 10500. |

Continued on next page

Technical Manual for the ACE-IT Software

| CMC Drawing Layer | Use |
|--------------------------|--|
| 510_BLDG_NO | Building 10510's building number (10510). |
| 510_BLDG_OUTLINE | Footprint for building 10510. |
| 510_DOORS | The blue line representing the door location and width for building 10510. |
| 510_EVACUATION | The red arrows indicating the evacuation routes for building 10510. |
| 510_FURNITURE | Furniture for building 10500. |
| 510_INTERIOR | The interior walls, doorswings, and interior details of building 10510. |
| 510_PLUMBING_FIXTURES | The restroom details for building 10510. |
| 510_RM_PEOPLE_TEXT | The human-readable room numbers of building 10510. |
| 510_SIDEWALK | The sidewalks surrounding building 10510. |
| 510_VENT | The vents on the roof of building 10510. |
| 510_WINDOWS | The red line indicating the interior window location and widths of building 10510. |
| 520_BLDG_NO | Building 10520's building number (10520). |
| 520_BLDG_OUTLINE | Footprint for building 10520. |
| 520_DOORS | The blue line representing the door location and width for building 10520. |
| 520_EVACUATION | The red arrows indicating the evacuation routes for building 10520. |
| 520_FURNITURE | Furniture for building 10520. |
| 520_INTERIOR | The interior walls, doorswings, and interior details of building 10520. |
| 520_PLUMBING_FIXTURES | Restroom details for building 10520. |
| 520_RM_PEOPLE_TEXT | The human-readable room numbers of building 10520. |
| 520_SIDEWALK | The sidewalks surrounding building 10520. |
| 520_VENT | The vents on the roof of building 10520. |
| 520_WINDOWS | The red line indicating the interior window location and widths of building 10520. |
| LEGEND | Map Legend. |
| PERIMETER | Requested perimeter of the site. |
| PERIMETER_SUGGESTED | Suggested perimeter of the site. |
| SENSOR_CAMERA | Camera locations, as well as, their field of view. |
| SENSOR_INFRA_RED | Infrared sensor locations, as well as their field of view. |
| SENSOR_MAGNETIC | Magnetic sensor locations, as well as their fields of view. |
| SENSOR_SEISMIC | Seismic sensor locations, as well as their fields of view. |

6.3.1 Layer Control

By carefully manipulating the display of the layers contained in the AutoCAD® drawing, different portions of the site map can be displayed. The manipulation of these layers is controlled through customized AutoLISP® files. The following table lists the LISP files and their use in the ACE-IT system.

The AutoLISP® files that determine layer control.

| AutoLISP File | Use |
|---------------|---|
| 500.lsp | The interior of building 10500 only. This includes all exterior and interior walls, doors, rooms, room numbers, plumbing, vents, and sidewalks. |
| 500evac.lsp | The evacuation routes of building 10500. |
| 510.lsp | The interior of building 10510 only. This includes all exterior and interior walls, doors, rooms, room numbers, plumbing, vents, and sidewalks. |
| 510evac.lsp | The evacuation routes of building 10510. |
| 520.lsp | The interior of building 10520 only. This includes all exterior and interior walls, doors, rooms, room numbers, plumbing, vents and sidewalks. |
| 520evac.lsp | The evacuation routes of building 10520. |
| all_ext.lsp | The building footprints and their relative positions along with the roads and sidewalks. |
| all_int.lsp | The building footprints with interiors and their relative positions, along with the roads and sidewalks. |
| perim.lsp | The building footprints, roads, the requested perimeter, and suggested perimeter. |
| sensors.lsp | The building footprints, roads, the requested perimeter, and suggested perimeter, as well as the sensor's origination point and field of view. |

7.0 VERSION CONTROL

The **ACE-IT** Version 1.0 software comprises the custom software modules described previously. The package includes Microsoft Word® and Write® documents, AutoCAD® drawing files, digital image files, digital video files, and Microsoft Access® database files.

Commercially available software (listed in section 1.0) must be used for proper system operation. **ACE-IT** will not function correctly if any of the listed software packages are upgraded to subsequent versions.

8.0 COMPUTER SECURITY

The contents of the *ACE-IT* system are unclassified. No sensitive information is included as a part of the package. Those items labeled as “sensitive” or “classified” in the system are fictitious and have been so labeled to enhance the realism of the training exercises.

The hardware and software for *ACE-IT* Version 1.0 is composed entirely of exportable technology and data. As a security precaution to prevent unauthorized access to the system, it is recommended that computers be provided with password protection. All users are responsible for obtaining proper licensing agreements for all commercial software packages. The additional precaution of regular, scheduled system backups is highly recommended, following appropriate procedures and precautions for storage of backup data.

APPENDIX A

A.0 INSTALLATION OF ACE-IT

Installation of the entire **ACE-IT** package and the supporting commercial software packages requires approximately 500 MB of free hard disk space. This must be contiguous space on a single hard drive, local to each computer. Install the packages in the following order:

1. DOS 6.0 or higher
2. Windows® 3.11
3. AutoCAD® release 12c (for Windows)
4. ArcCAD® 11.3 (for Windows)
5. Access® 2.0
6. ZyIndex® 5.15 (for Windows)
7. Visual BASIC® 3.0 (optional)
8. Access® 2.0 Compatibility layer for Visual BASIC®
9. **ACE-IT** system files

NOTE: Before beginning the installation of **ACE-IT**, the user should be aware that the installation can require from one to eight hours, depending on your computer.

A.1 AUTOEXEC.BAT AND CONFIG.SYS NOTES

During the installation process, each of the systems to be installed may require that changes be made to the AUTOEXEC.BAT and CONFIG.SYS files. The best choice is to allow the system to make the required changes and to test each application after installation.

A.2 SHARE

The DOS program SHARE.EXE must be installed. Check your AUTOEXEC.BAT File to see if SHARE has already been installed on the system. If not, there are many possible parameters for SHARE. Use the following parameters for SHARE.EXE in your AUTOEXEC.BAT File:

```
C:\DOS\SHARE.EXE /L:500
```

This set of parameters will work with **ACE-IT**. If SHARE is already in use by your system with other parameters, it may be necessary to replace the SHARE parameters in your AUTOEXEC.BAT file with the above parameters.

A.3 EMM386

Do not use EMM386.EXE either for Windows® or DOS. This memory manager may cause problems with **ACE-IT** and should never be used.

A.4 SMARTDRIVE

There are two different SMARTDRIVE files to choose from, one in the **C:\Windows** directory, SMARTDRIVE.EXE, and the other in the **C:\DOS** directory, SMARTDRIVE.SYS. Either will work with **ACE-IT**. Check the contents of your AUTOEXEC.BAT and CONFIG.SYS before making these changes, since only one of these two files can be installed. SMARTDRIVE can be added to your AUTOEXEC.BAT or CONFIG.SYS files without parameters, if desired, as follows:

In AUTOEXEC.BAT, add this line

```
C:\WINDOWS\SMARTDRIVE.EXE
```

If using the SMARTDRIVE.SYS file, then in the CONFIG.SYS file, add this line:

```
DEVICE=C:\DOS\SMARTDRIVE.SYS
```

(Remember, add only one; adding both will cause system problems.)

A.5 HIMEM.SYS

As with the SMARTDRIVE.EXE files, there are two options for HIMEM.SYS, one in DOS and the other in Windows®. If your system already has Windows®, and runs fine, you will not need to make any changes to settings for HIMEM.SYS. Your settings for HIMEM.SYS will be found in the CONFIG.SYS file.

A.6 AUTOCAD® (ACAD) FOR WINDOWS® INSTALLATION

Insert diskette 1 of AutoCAD® for Windows® into the floppy disk drive.

Use Run... under the File Menu of the Windows® Program Manager to run the program setup.exe located on the first installation disk.

Install from displayed menu. NOTE: If you have plenty of disk space just install all files but remember that **ACE-IT** requires approximately 500 MB even with the pared down ACAD installation. If you have limited disk space available, the following sections from the AutoCAD® custom installation menu should be installed:

ACAD Executable/support
ACAD Support source

Technical Manual for the ACE-IT Software

ACAD Iges font
ACAD Tutorial
ACAD ADS/DDE
ACAD SQL extension files

Install AutoCAD® to C:\ACADWIN directory (the default selection).

Follow the directions displayed on the screen as you insert the diskettes.

The install program will ask you for a program group name. Use the AutoCAD® defaults and pick your choice of icons.

Next, start up AutoCAD® by double-clicking on the Windows® AutoCAD® icon.

After installing AutoCAD®, the system will require configuration before running. The following options have been tested with **ACE-IT**. For all components except the plotter or printer, the default values should be acceptable. For the plotter/printer, configure your AutoCAD® to match your system's plotting/printing device.

Video displays: choose < 1 > Windows® Accelerated Driver
Accept default driver configuration
Do not correct the aspect ratio of CRT display
Choose current system pointing device
Choose the appropriate plotter/printer from the list
Choose "**ACE-IT**" as a login name
Select one as the maximum number of users
Reply "No" when asked whether or not to use the executable from a read only directory
Reply "No" for password authorization for a network server
Reply "No" when asked whether or not to use file locking

Examine current configuration before exiting
Keep the configuration if correct
Exit configuration menu (Save Changes)

AutoCAD® must also be configured so that the displays in **ACE-IT** are correct. After configuring AutoCAD®, the system will automatically start up. Choose PREFERENCES under the FILE menu.

In the PREFERENCES Window, select COLOR and then select BLACK as the Screen Background color from the pallet. Deselect the TOOLBOX from being shown on every AutoCAD® Drawing. When finished, select Save to ACAD.INI to ensure that later AutoCAD® sessions called by **ACE-IT** will have these settings.

Exit AutoCAD® to save changes and have them take effect on your next AutoCAD® run.

If there are any problems with the AutoCAD® installation, see the AutoCAD® users manual and use the command CONFIG in AutoCAD® to reconfigure as necessary.

A.7 ARCCAD® RELEASE 11.3 FOR WINDOWS® INSTALLATION

Prior to installing ArcCAD®, place the proper hardware key on the parallel port on back of computer. ArcCAD® may install without this key, but will not configure or otherwise run. This key is worth approximately \$3,000 US. PLEASE EXERCISE PROPER SAFEGUARDS!

Insert diskette 1 of ArcCAD® for Windows® into the floppy disk drive.

Use Run... under the File Menu of the Windows® Program Manager to run the program install.exe located on the first installation disk.

The initial menu will have several choices, but you must INSTALL ArcCAD® before configuring. The installation program will present you with several installation options. Follow the instructions on the screen to install ArcCAD® Release 11.3 for Windows®.

Check to make sure that AutoCAD® is in C:\ACADWIN directory, and that ArcCAD® has found and identified its location properly.

ArcCAD® should be installed into the default directory C:\ARCADWIN. Follow the defaults for all installation options.

After installation is complete choose AUTOMATIC CONFIGURATION from the menu. Allow the system to make the required changes to AUTOEXEC.BAT and CONFIG.SYS files.

Select "Load ArcCAD®" on each AutoCAD® startup.

A.8 MICROSOFT ACCESS® VERSION 2.0 INSTALLATION

Insert diskette 1 of Access® 2.0 for Windows® into the floppy disk drive.

Choose Run... under the File Menu of the Windows® Program Manager to run the program setup.exe located on the first installation disk.

Install Access® into the C:\ACCESS directory (this is the default). Choose Complete/Custom from the setup menu. Do not install ODBC or SAMPLE APPS from the complete/custom menu.

The Program group goes into the Access® group (default).

Complete the loading of the disks.

A.9 ZYINDEX® INSTALLATION

Insert diskette 1 of ZyIndex® for Windows® into the floppy disk drive.

Choose Run... under the File Menu of the Windows® Program to run the program setup.exe located on the first installation disk.

Install ZyIndex® to the C:\ZYINDEX (this is the default).

Follow the instructions that appear and perform a default install of ZyIndex®.

***NOTE:** You will need your license agreement card to install ZyIndex® as it requires the input of the keycode from that agreement to properly install the program.*

A.10 VISUAL BASIC® INSTALLATION

Insert diskette 1 of Visual BASIC® into the floppy disk drive.

Choose Run... under the File Menu of the Windows® Program to run the program setup.exe located on the first installation disk.

A full installation of Visual BASIC® takes approximately 32 MB of hard disk space. If you have hard disk space, it is recommended that you perform a complete install. If you do not have disk space, do not install the Icon Library, Sample Applications, Help Compiler, and clipart; this will cut down the disk space requirement to approximately 17 MB.

Install Visual BASIC® to C:\VB directory (this is the default).

Follow the instructions that appear and perform the install of Visual BASIC®.

A.11 Visual BASIC®/Access® Compatibility Layer Installation.

For Visual BASIC® to interface properly with Access® Version 2.0, the Jet 2.0/Visual BASIC® 3.0 must be installed.

Insert diskette 1 of Jet 2.0/Visual BASIC® 3.0 into the floppy disk drive.

Choose Run... under the File Menu of the Windows® Program to run the program setup.exe located on the installation disk and perform the default installation procedure.

A.12 VISUAL BASIC® GANTT VBX INSTALLATION

Insert diskette 1 of Visual BASIC® Gantt VBX into the floppy disk drive.

Choose Run... under the File Menu of the Windows® Program to run the program setup.exe located on the installation disk and perform the default installation procedure.

One of the Visual BASIC® applications used in *ACE-IT* requires the second party Visual BASIC® tool Gantt.VBX. Installation of the Gantt.VBX file into the C:\WINDOWS\SYSTEM is necessary at a minimum. Installation of the Gantt VBX installable which includes various examples will also allow the *ACE-IT* programs to run.

A.13 ACE-IT PROJECT FILES INSTALLATION

The *ACE-IT* project files are provided either on a CD-ROM, QIC (Quarter Inch Cartridge) Backup Tape, or other format as required. They should be restored into the C:\INSPECTR directory. There are approximately 300 files that will be placed on your hard drive, in the directory structure described in section 2.0.

Technical Manual for the ACE-IT Software

Distribution

- 2 U.S. Department of Energy
Attn: Josh Segal
International Policy and Analysis Division (NN-42)
1000 Independence Ave., SW
Washington, DC 20585

- 2 Ogden Environmental and Energy Services
Attn: Jon Nelson
7301 Indian School Rd., NE
Albuquerque, NM 87110

- 1 MS-0567 Richard Preston, 5346
- 1 MS-1373 Arian Pregonzer, 5341
- 22 MS-1373 Pauline Dobranich, 5341

- 1 MS-9018 Central Tech Files, 8940-2
- 5 MS-0899 Technical Library, 4916
- 2 MS-0619 Review & Approval Desk, 12690 for DOE/OST I
- 1 MS-0161 Patent and Licensing Office, 11500