ACM

air combat simulation for UNIX and Windows

version 5.0

Web Simulations Incorporated
www.websimulations.com

Introduction 4

Acknowledgements 4

Abbreviations 5

ACM for UNIX 5

Command line options 5

Other Initialization Parameters 8

Keyboard Commands 11

The Head-Up Display 13

Ladders 13

Angle of Attack 13

The G-Meter 14

Weapon State 14

The Radar System 15

Electronic Countermeasures 18

Radar Warning Receiver 18

Weapon Systems 19

Air-to-Air Missiles 19

Cannon 19

The HSI 21

Drones 22

Stealth Mode 23

Developer Studio Setup (ACM for Windows only) 28

Building ACM on UNIX Systems 28

Defining New Scenes 30

Team Locations 30

Ground Color 30

Runway record 30

TACAN Record 31

ILS Record 31

Features 32

Feature Record 32

Defining New Aircraft 33

*Starting ACM for Windows*_____*41*

*Setting up an ACM session (ACM for Windows only)*_____*42*

Introduction

ACM is a distributed multiplayer air combat simulation that runs on UNIX and Windows. Players engage in air-to-air combat with infrared missiles and cannon.

The first version of ACM was released in 1991 via the venerable Usenet comp.sources.unix newsgroup. Since then, ACM has been upgraded to support the IEEE 1278 Distributed Interactive Simulation (DIS) protocol.

Acknowledgements

Parts of ACM were derived from other sources. The DIS protocol compiler is based on technology originated at Sun Microsystems in their public domain ONC RPC product. ACM's DIS glue code was created by Mats Loftkvist. The C++ interface to Microsoft's DirectDraw is derived from samples created by Nigel Thompson in his book, 3D Graphics Programming for Windows 95.

Many others have generously supplied bug fixes and other changes to ACM since it was originally released. In particular, I'd like to thank Brad Bass and Tim Tessin for their encouragement and continued support. Charlie Briggs and Tom Giertz have helped out a lot, too.

Suggested Further Reading

3D Graphics Programming for Windows 95
by Nigel Thompson ISBN 1-57231-345-5

Fighter Combat
by Robert L. Shaw ISBN 0-87021-059-9

Aircraft Control and Simulation
by Brian L. Stevens and Frank L. Lewis ISBN 0-471-61397-5

1278.1-1995 IEEE Standard for Distributed Interactive Simulation--Application Protocols ISBN 1-55937-572-8

The Surveying Handbook, second edition
edited by Russel Brinker and Roy Minnick ISBN 0-412-98511-X

Support

Support information can be found at our web site: http://www.websimulations.com/support

Abbreviations

ACM	air combat maneuvering
AAM	air-to-air missile
AOA	angle of attack
DIS	Distributed interactive simulation
ECM	electronic countermeasures
HUD	head-up display
IAS	indicated airspeed
IR	infrared
kts	nautical miles per hour. A nautical mile is defined as 6076 feet.
KIAS	knots indicated airspeed
RWR	radar warning receiver
TAA	target aspect angle
TAS	true airspeed
TEWS	threat early warning system – a form of radar warning receiver

ACM for UNIX

Command line options

ACM uses POSIX style command line formatting:

```
acm [ -init command-file-name ]
    [ -stealth ]
    [ -subject-entity-id site-id.appl-id.entity-id ]
    [ -geometry x11-geometry-specification ]
    [ -frame-rate target-frame-rate-hertz ]
    [ -update-rate internal-update-rate-hertz ]
    [ -plane { F-16 | MiG-29 | Su-30 | C-172 } ]
    [ -team { 1 | 2 } ]
    [ -latitude latitude-degrees ]
    [ -longitude longitude-degrees ]
    [ -altitude altitude-msl-feet ]
    [ -heading initial-heading-degrees-true ]
    [ -airspeed-kts initial-airspeed-knots ]
    [ -dis-site DIS-site-id ]
    [ -dis-appl DIS-application-id ]
    [ -dis-exercise DIS-exercise-id ]
    [ -visibility flight-visibility-nm ]
    [ -js joystick-serial-port-pathname ]
    [ -end-game ]
    [ -threshold-range threshold-nautical-miles ]

-init command file-name
```

Take extra command options, in command line format, from the specified text file. For example, say there is a file in your home directory named '.acmrc'. Its contents look like this:

```
-dis-site 34
-dis-appl  4
```

New lines are treated as normal white space, so feel free to separate command line options onto multiple lines.

From a shell, you enter the command:

```
$ acm -geometry 800x600 -init ~/.acmrc
```

This would be equivalent to:

```
$ acm -geometry 800x600 -dis-site 34 -dis-appl 4
```

-stealth

Start ACM in stealth mode. ACM allows users to monitor out-the-window views for any aircraft active in an exercise. Additionally, ACM supports an experimental DIS Request/Grant Control protocol that would permit ACM to "take over" aircraft of similarly enabled applications. See the "Stealth Mode" section for detailed information on this capability.

-subject-entity-id *site-id.appl-id.entity-id*

This option can be used in conjunction with the -stealth flag to identify the initial DIS entity to be "stealthed".

```
$ acm -stealth -subject-entity-id 32.1.1
```

-geometry *x11-geometry-specification*

Specify precise location and size settings for the main ACM window.

-frame-rate *target-frame-rate-hertz*

Specifies a not-to-exceed rate at which ACM will attempt to render the cockpit scene. The default is to update the scene as fast as possible, which will use all available CPU time on a system.

-update-rate *internal-update-rate-hertz*

Specifies the update rate for the inner integration loop for the simulation. The default is to run as fast as possible. If the -frame-rate option is specified, the default changes to 50 hertz.

-plane { F-16 | MiG-29 | Su-30 }

Allows the user to select the aircraft type to be flown.

-scene *scene-file-name*

ACM takes airport and NAVAID information from a text-based scene file. Normally, a file named "default-scene" is used.

-team { 1 | 2 }

Allows the user to select which team the aircraft is to be associated with. Team 1 corresponds to the DIS "Friendly" force, team 2 is the "Opposing" force. Each team has an associated "home" location which is defined in the scene file read by ACM at startup.

-latitude *latitude-degrees*
-longitude longitude-degrees
-altitude altitude-msl-feet
-heading *initial-heading-degrees-true*
-airspeed-kts *initial-airspeed-knots*

These options may be combined to tailor the startup location of the ACM aircraft. If the -altitude option is *not* supplied, ACM automatically sets the aircraft on the ground at the specified location. Care should be taken not to specify an altitude too close to the ground.

-dis-site *DIS-site-id*
-dis-appl *DIS-application-id*
-dis-exercise *DIS-exercise-id*

Use these options to select a specific IEEE 1278 DIS site or application identifier. By default, ACM uses the last two octets of the host machine's IP address as the site and application identifiers. The default exercise identifier is one (1).

-visibility *flight-visibility-nm*

Sets the flight visibility in the ACM environment. The default is 50 nautical miles.

-js *joystick-serial-port-pathname*

ACM supports Noesis and Colorado Spectrum (now defunct) serial joystick adapters. This command specifies the name of the serial port connected to the joystick adapter and enables joystick polling.

-end-game

This flag is only valid in stealth mode. Hostile aircraft near the subject aircraft are tracked. If any of these aircraft moves within the subject aircraft's radar locking range, then ACM will request control of subject aircraft[1]

-threshold-range *threshold-nautical-miles*

This flag is only valid in the end-game mode. If a hostile aircraft moves within the specified threshold distance of the stealthed (subject) aircraft, ACM will attempt to take control and engage the hostile target. If threshold range is not specified, then the subject aircraft's radar lock range is used as the distance threshold.

Other Initialization Parameters

At startup, ACM reads several text-based data files to configure the operating environment.

default-scene

This file contains the locations of each team's home base, airport runway information, NAVAID information, as well as the locations of man-made ground features. The -scene command line option may be used to direct ACM to read an alternate scene file.

object-map.txt

This file defines a mapping from DIS entity types to 3D object definition files. The object files are used to render an image of the entity.

munition-map.txt

This file defines the explosion and damage producing characteristics of DIS munition entity type and warhead combinations.

inventory

This file defines the performance characteristics of all aircraft types modeled by ACM. The layout of this file is defined in the section titled "Defining New Aircraft".

1 Control is requested using a variant of the proposed DIS 2.1.4++ transfer control protocol sequence.

IEEE 1278.1 (DIS) Compliance

ACM emits and recognizes the following DIS PDU Types:

Entity State

Fire

Detonation

Electromagnetic Emission

ACM uses the DIS/x DIS interface library. DIS/x supports dead reckoning of external entities using any of these methods:

Name	Enumeration Value
Static	1
FPW	2
RPW	3
RVW	4
FVW	5

These dead reckoning methods are currently **unsupported** by DIS/x:

Name	Enumeration Value
FPB	6
RPB	7
RVB	8
FVB	9
RPW	10
RVW	11

Entities managed by ACM emit entity state PDU's specifying the RPW(3) dead reckoning method. No articulation parameters are currently sent by entities managed by ACM. As with most current DIS-based simulations, altitudes are expressed as heights above the WGS-84 ellipsoid, not as heights above the geoid. Dead reckoning threshold values are hard-coded in this version of the product:

Description	Value
Maximum time between Entity State PDUs	5 seconds
Dead Reckoning Cartesian distance error	2 meters

Dead Reckoning angular orientation error	3 degrees

When operating in IEEE 1278 mode, the default UDP port number for PDU transmission is 3000.

Keyboard Commands

General

G	Landing gear up/down
B	Wheel brakes on/off
H	Flaps down 10 degrees
Y	Flaps up 10 degrees
S	Deploy speed brakes
W	Retract speed brakes
L	Create a drone opponent aircraft.
Q	Quit the simulation

? or F1	Help
F2	Set Preferences

Views – view buttons are located on the PC's numeric keypad

KP8	Forward view
KP2	Rear view
KP4	Look Left
KP6	Look Right
KP5	Look Up
N	Chase plane view (this view button is not located on the numeric keypad)

Joystick Buttons

Button2 (top)	Select weapon (Sidewinder IR missile or 20mm cannon)
Button1 (front)	Fire selected weapon

Radar

R Toggle Radar Mode (Standby, Normal, ACM 20x30, STT, HSI-mode) (also Button3 on Microsoft Sidewinder™ Joysticks)

T Break Lock -- track a different target (also Button4)

Throttle (for joysticks without a throttle control)

4 Full Power

3 Increase Power

2 Decrease Power

1 Flight Idle

A Toggle Afterburner

NAVAID receiver control

7 Rotate HSI Selected Course Pointer (SCP) 20 degrees

8 Rotate HSI SCP 1 degree

9 Increment NAV1 receiver frequency by 1.0 MHz

0 Increment NAV1 receiver frequency by 0.05 MHz

Trim

Trim buttons may be pressed until the desired trim is attained

Ctrl-Shft-UpArrow Forward (down) pitch trim

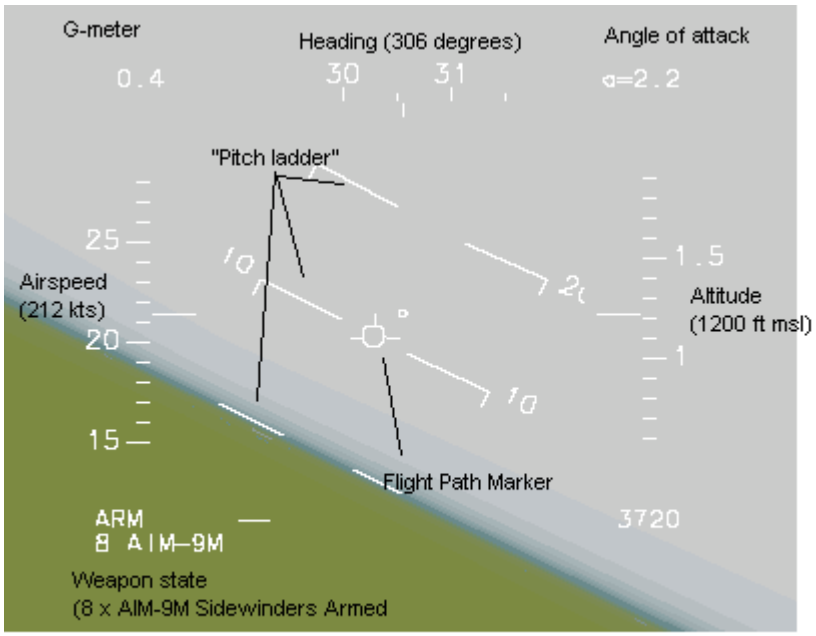
Ctrl-Shft-DownArrow Aft (up) pitch trim

Ctrl-Shft-LeftArrow Left roll trim

Ctrl-Shft-RightArrow Right roll trim

The Head-Up Display

The Head-Up Display (HUD) is a collection of indicators that permits pilots to focus their attention on what's going on outside of the cockpit. These indicators are projected onto a flat pane of glass located near the windscreen. Much of the information is displayed graphically to give the pilot a quicker understanding of the immediate situation.



The Heads-Up Display

THE ACM HEAD UP DISPLAY

Ladders

Four tape-style "ladders" display basic flight information. On the left, is the plane's true airspeed. Each minor tick on the airspeed ladder represents 10 nautical miles per hour (also known as knots and abbreviated "kts").

On the right is the altitude ladder. Each tick represents 100 feet. In ACM, the ground is completely flat (except for a few mountains) and located at sea level. A rate-of-climb readout is located just below this ladder; rate-of-climb is expressed in feet per minute.

In the lower center is a horizontal heading ladder. Each tick represents five degrees.

In the center is an attitude ladder. Each line corresponds to ten degrees of aircraft pitch. The ladder rolls as the aircraft does, providing an artificial horizon.

Angle of Attack

The ACM HUD has two indicators to give the pilot cues as to the planes current angle of attack and sideslip. First, above the altitude ladder, is a readout of the plane's current angle of attack in degrees. The ACM F-16 will stall at a positive angle of 30 degrees and a negative angle of -30 degrees.

Additionally, a plane-shaped "flight path marker" indicates the aircraft's current direction of travel. Level flight occurs when the flight path marker is aligned with the zero- degree artificial horizon line.

The G-Meter

A readout of the current vertical G-force on the pilot is located above the airspeed ladder.

Weapon State

Below the throttle indicator are discretes that show the state of the currently selected weapon system.

The Radar System

The ACM radar system has a field of view that extends 130 degrees laterally and vertically from the front of the aircraft. The F-16 radar set's range is about 80 nm. All aircraft within that range and field of view will appear as a box on the radar display

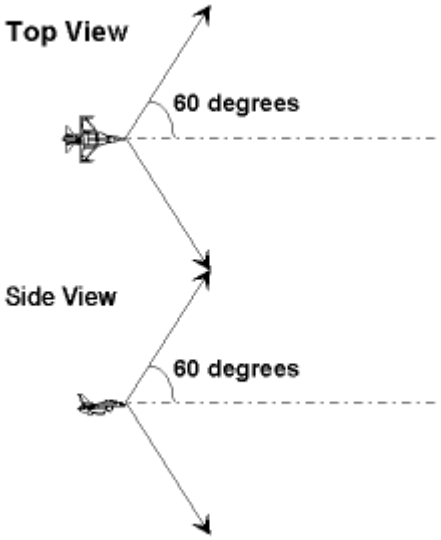
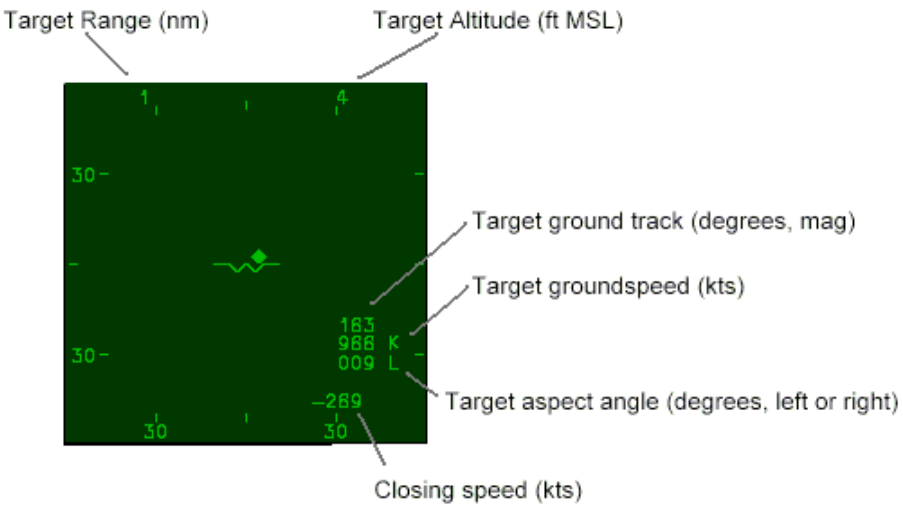


FIGURE 2 RADAR COVERAGE

The ACM radar display presents a forward looking view of radar targets. Think of it as a television monitor connected to a forward-pointing TV camera that has a very wide angle lens.

If any objects are close enough to be radar-visible, the set will automatically lock onto the nearest threat. With radar lock acquired, your display will provide detailed information about the locked, or "primary", target's disposition. A primary target appears as a filled diamond (◆) on the radar display -- the display provides information on the primary target's range and altitude, as well as it's current heading, your desired relative heading to intercept and the rate of closure of your two aircraft.

FIGURE 3 THE ACM RADAR DISPLAY



The radar set's ability to establish a positive lock on a target extends for a shorter range than the radar's maximum detection range -- radar lock is limited to about 60 nm on the F-16 and 30 nm on the MiG-29.

If multiple targets are plotted on your radar display, you can lock onto other targets by pressing the target reject key (Q).

The radar on/standby key (R) can be used to toggle your radar set between its normal operating mode and a standby state. When your radar is in standby mode, your aircraft emits no radar energy and can be more difficult to detect.

Electronic Countermeasures

Radar Warning Receiver

Each aircraft is equipped with a special device called a Radar Warning Receiver (RWR). It's display is the round "CRT" to the left of the radar display. It works much like a radar detector that you can buy for your car. The receiver can detect radar emissions from other aircraft and will plot a box on the RWR display representing the relative direction of that radar threat. Opposing team's radar emissions appear as filled boxes. Friendly aircraft appear as open boxes. This receiver cannot detect aircraft that have their radar emissions directed away from your aircraft, nor is it capable of detecting aircraft that have their radar sets turned completely off.

Weapon Systems

Your aircraft is equipped with heat-seeking missiles and a 20-millimeter cannon. Weapon information is displayed in the lower left-hand corner of your HUD. Different weapons may be selected by pressing mouse button 3. The currently selected weapon is fired by pressing mouse button 2.

Air-to-Air Missiles

The missiles are patterned after U.S. AIM-9M Sidewinders. They can detect infrared (IR) targets at any aspect (not just from the rear). Their range varies dramatically with the altitude and closure rate. The missile subsystem couples with your radar set to provide time-to-impact information.

FIGURE 4 MISSILE DISCRETES

<Picture: (Figure)>

This missile has a solid rocket motor that burns for about 8 seconds. After burn-out, it will still track towards its intended target but may lose speed too rapidly to catch it. Your heat seeking missiles don't arm themselves until three seconds after launch. Because of that, you should not fire at a target that is closer than three seconds of flight time away.

Missile status discretes and their meanings:

ARM	A missile is armed, but its seeker has not yet acquired a target in its field of view.
LOCKED	A missile is armed and its seeker has locked onto a heat source. The missile is ready to fire.
----	The missile's seeker has acquired a target, but it is out of range based on current time-to-target information.

Cannon

Cannon can be used to engage targets at closer range. Your cannon is modeled after the U.S. M61-A1 Vulcan. Aircraft typically begin a mission with 500 rounds of ammunition; the cannon fires at 3000 rounds per minute so you only have enough ammunition for a ten-second continuous burst -- use it carefully.

FIGURE 5 CANNON DISCRETES

<Picture: (Figure)>

When cannons are selected, the HUD is in Lead Computing Optical Sight (LCOS) mode. On the HUD, a circular aiming reticle is displayed. The HUD couples with the radar set to provide a visible cue of the target's current range. The aiming reticle is surrounded by 12 ticks. An inner arc represents the current range to the target: each arc tick represents 1000 feet of distance. The aiming reticle moves across the HUD to show a good aiming point based on the target's range and your aircraft's pitch and turn rate. If the range is large and or your pitch and turn rates are fast, you may see no reticle at all: the aiming point is simply out of the HUD's field of view.

An unmoving cross (a "+") will be displayed on the HUD in LCOS mode. This marker denotes the boresight of the cannon -- the direction that the cannon's barrel is actually pointing.

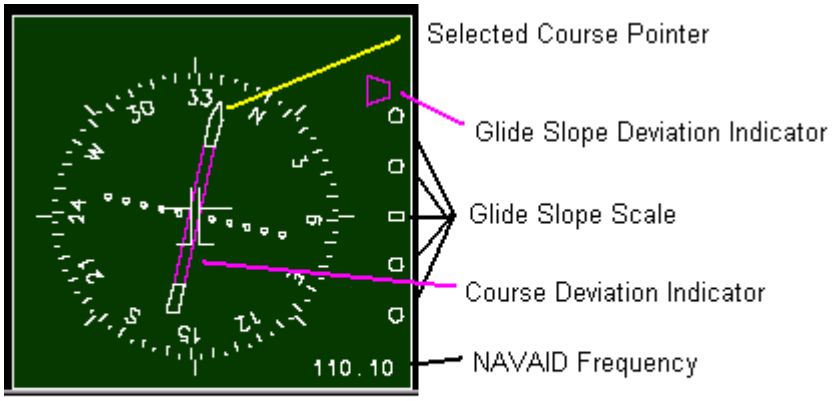
FIGURE 6 A HARD-TURNING SHOT

<Picture: (Figure)>

Figure 6 shows another example of what you might see in a dogfight. All planes are turning hard to the left at a relatively low speed. You are in LCOS mode and see that the reticle is positioned ahead of the intended target. To get the kill, simply relax the turn long enough to get into position and then take the shot.

The HSI

The Horizontal Situation Indicator, or HSI, is a nifty device to aid in instrument flying. In ACM, the HSI shares space with the radar display. Use the (R) key to change radar modes until the HSI is displayed. Use the (9) and (0) keys to select the desired NAVAID frequency. Use the (7) and (8) keys to orient the Selected Course Pointer (SCP).



The Horizontal Situation Indicator (HSI)

Drones

Drones are automated opponent aircraft. They are created by pressing the “L” key.

Drones aggressiveness setting controls how hard a drone aircraft will attempt to maneuver to attack other planes.

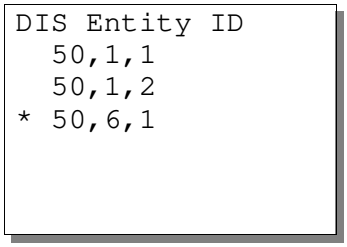
A setting of zero means the drone will leave the stick and rudder untouched – it will probably impact with the ground in just a minute or less. A setting of five will usually make for an interesting opponent. A maximum setting of ten will probably make even skilled pilots sweat a bit.

Stealth Mode

ACM's stealth mode allows users to monitor out-the-window views for any aircraft active in an exercise.

```
$ acm -stealth
```

In stealth mode, the radar display is used to display DIS entity id's of aircraft participating in the exercise. To select and aircraft to follow, use the mouse to click on one of these identifiers. The entity id of the aircraft you are following will be highlighted in magenta.



If there is a large number of entities in an exercise, use the PAGE UP and PAGE DOWN keys to scroll through the list.

Transfer of Control

ACM supports an experimental DIS 2.1.4++ control request protocol. It permits it to "take over" aircraft of similarly enabled applications. For ACM to take over an aircraft, it must have model information in the "inventory" file describing the characteristics of that aircraft type. Those aircraft entities will be marked with an asterisk (*) in the left margin of the stealth browser display. Double click an aircraft entity id to take control.

The Transfer Control Protocol

Transfer Control Request PDU

Field Size (bits)	Transfer Control Request PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration Exercise ID—8-bit unsigned integer PDU Type —8-bit unsigned integer Protocol Family—8-bit enumeration Timestamp—32-bit unsigned integer Length—16-bit unsigned integer Padding—16-bits unused
48	Originating Entity ID	Site—16-bit unsigned integer Application—16-bit unsigned integer Entity—16-bit unsigned integer
48	Receiving Entity ID	Site—16-bit unsigned integer Application—16-bit unsigned integer Entity—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Transfer Type	8-bit enumeration
48	Entity ID to be Transferred	Site—16-bit unsigned integer Application—16-bit unsigned integer Entity—16-bit unsigned integer
32	Number of Record Sets (R)	32-bit unsigned integer
$96 + (L_1 \times Q_1) + P_1$	Record Set #1	Datum ID—32-bit enumeration Record Set Serial Number—32-bit unsigned integer Record Length—16-bit unsigned integer (L_1) Record Count—16-bit unsigned integer (Q_1) Record Values—($L_1 \times Q_1$) bits Padding—P_1 bits
		• • •
$96 + (L_R \times Q_{R)} + P_R$	Record Set #R	Datum ID—32-bit enumeration Record Set Serial Number—32-bit unsigned integer Record Length—16-bit unsigned integer (L_R) Record Count—16-bit unsigned integer (Q_R) Record Values—($L_R \times Q_R$) bits Padding—P_R bits

$$\text{Total Transfer Control Request PDU size} = 320 + \sum_{i=1}^{R} (96 + (L_i \times Q_i) + P_i) \text{ bits}$$

where

R is the number of Record Sets

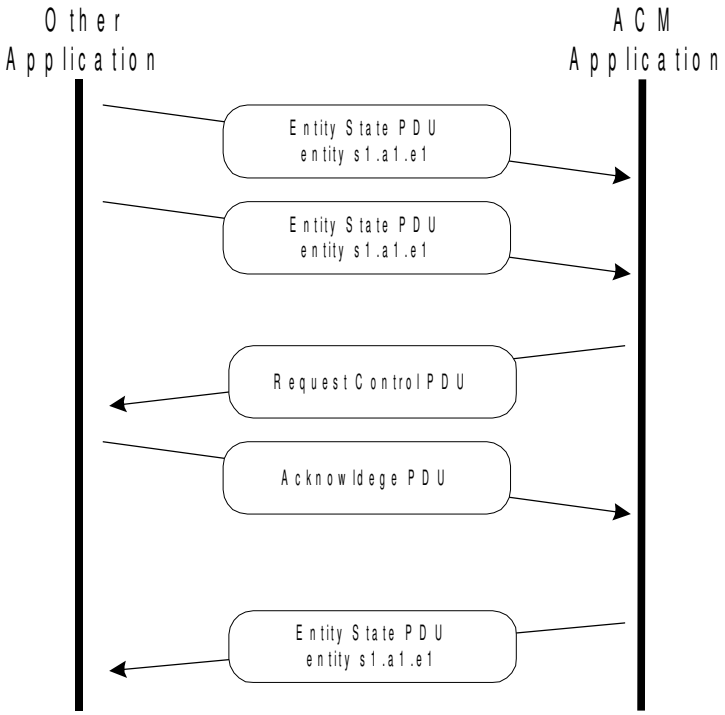
ACM currently ignores record sets in a transfer control PDU.

Transfer Type

This section specifies the 8-bit enumeration for the Transfer Type field of the Transfer Control PDU.

Field Value	Nature
0	Other
1	Controlling application requests transfer of an entity
2	Application desiring control requests transfer of an entity
3	Mutual exchange / swap of an entity
4	Controlling application requests transfer of an environmental process
5	Application desiring controls request transfer of an environmental process
6	Mutual exchange of an environmental

Assuming Ownership



Accepting Ownership by Request

Another feature of ACM's transfer control protocol support is that other applications may request that ACM take control of an aircraft that the other application controls.

The Source Code (ACM for Windows only)

This section contains some information to help you navigate through the ACM for Windows sources.

Here's a summary of the contents of each of the directories in the source code tree:

```
source\acm              Toplevel directory for ACM for Windows
source\acm\pc               The MFC-based front-end to the
simulator
                            This directory holds the main
'acm.dsw'
                            VC++ 5.0 project workspace.
source\acm\pc\hlp            Help files associated with the
project
source\acm\pc\res            Windows resources (bitmaps, icons,
…)
source\acm\src          C-based flight simulator code
source\acm\V\lib        The 3D graphics library
source\acm\V\test
source\acm\dis\lib           The DIS/x IEEE 1278 library
source\acm\dis\test
source\acm\dis\disgen IEEE 1278 protocol compiler
source\acm\tools        Various infrequently used tool sources
source\include          Supplemental system header files
source\oncrpc           A Windows port of Sun's ONC RPC libraries
source\rpcgen           A Windows port of Sun's RPCGEN command
```

Developer Studio Setup (ACM for Windows only)

1. Install this software (if you are reading this, then you have already done that).
2. Ensure Visual C++ 5.0 has been installed
3. Install the DirectX Software Development Kit.
4. Several directories included in the source release must be added to the default search directories. From the **Tools>Options…** menu in Developer Studio select the **Directories** tab. Select **Include files** in the **Show Directories for:** box. Double click the empty entry at the ***bottom*** of the current list and add the **`c:\sources\include`** directory – adjust the directory name to suit the correct ACM for Windows source installation location on your system.
5. From the **Tools>Options…** menu in Developer Studio select the **Directories** tab. Select **Executable files** in the **Show Directories for:** box. Double click the empty entry at the ***bottom*** of the current list and add the **`c:\bin`** directory – adjust the directory name to suit the correct ACM for Windows source installation location on your system.
6. Double click the empty entry at the ***bottom*** of the current list and add the **`c:\DXSDK\sdk\inc`** directory – adjust the directory name to suit the correct DirectX Software Development Kit installation location on your system. *Use the arrow keys to move this item to the top of the directory list*. If you do not do this, Visual C++ will incorrectly use the DirectX version 2 include files that shipped with VC++ 5.0.
7. From the **Tools>Options…** menu in Developer Studio select the **Directories** tab. Select **Library files** in the **Show Directories for:** box. Double click the empty entry at the ***bottom*** of the current list and add the **`c:\DXSDK\sdk\lib`** directory – adjust the directory name to suit the correct DirectX Software Development Kit installation location on your system. *Use the arrow keys to move this item to the top of the directory list*. If you do not do this, Visual C++ will incorrectly use the DirectX version 2 library files that shipped with VC++ 5.0.
8. Open the main ACM for Windows VC++ workspace. It is named **`\source\acm\pc\acm.dsw`**

The basic workspace includes five projects:

acm	\source\acm\pc\acm.dsp	The main program
fsim	\source\acm\src\fsim.dsp	the flight simulation library
vlib	\source\acm\V\lib\vlib.dsp	the 3D graphics library
dis	\source\acm\dis\lib\dis.dsp	the IEEE 1278 DIS library
disgen	\source\acm\dis\disgen\disgen.dsp	a DIS header file generator

Building ACM on UNIX Systems

ACM can be downloaded from the Web Simulations site (www.websimulations.com) as a gzipped tar archive. Upack the source tree:

```
$ zcat acm-5.0.tar.gz | tar xvf -
```

To build ACM, type:

```
$ cd acm-5.0
$ ./configure
$ make
```

On systems where X11 is installed in an unusual location, configure may have trouble locating the X include files and libraries (you'll know that because the compiler will complain that it can't find files like "X11/X.h"). If this problem crops up, you can help configure to set up the Makefiles by telling it where to find X11:

```
$ x_includes=/usr/myX11/include; export x_includes
$ x_libraries=/usr/myX11/lib; export x_libraries
$ ./configure
$ make
```

By default, configure will compile ACM using gcc if gcc is found in your PATH environment variable. Some combinations of gcc and OS platforms have problems with ACM (gcc 2.2.2 seems to have problems on SunOS and HP-UX, for example). You can override configure's selection of gcc:

```
$ CC=cc; export CC
$ ./configure
$ make
```

Defining New Scenes

The scene file defines the visible ACM "world". It should be stored as a UNIX format ASCII text file. Records are delimited by line terminators (ASCII linefeeds). Fields in a record are separated by one or more spaces or tab characters.

NOTE: This document describes the elements of the scene file format that are currently used by ACM. Some elements may be seen in distributed scene files that are not processed by the current version of ACM. In effect, ACM treats those undocumented elements as comments.

On startup, ACM uses the '`default-scene`' file from the '`objects/`' directory. With ACM for UNIX, you can override this default by specifying the '-s scene-file-pathname' option.

The general format of the file follows this layout:

<team 1 location record>

<team 2 location record>

<ground color record>

<runway/NAVAID records>

`FEATURES`

<object/feature records>

Team Locations

ACM groups participants (players) into two teams. The scene file defines the initial location for each team. Typically, this location will be positioned and oriented at the end of a runway at the player's "home" airport.

Example

```
32-58-18.798N 096-50-16.604W 644.0 154
```

Fields

Latitude

Longitude

Altitude (ft, MSL)

Initial Heading (degrees, true)

Ground Color

An X11-style **`(#RRGGBB)`** color specification defining the color of the ground. Red, green and blue color components are hexadecimal value from 00 to FF.

Runway record

```
RWY ADM 17/35 762 6879 150 34-18-43.074N 097-01-24.530W 34-17-35.031N 097-01-25.130W
495 830
```

Fields

Record Identifier: 'RWY'

Three/Four letter airport code

Runway identifier

Runway Altitude (ft, MSL)

Near End Latitude

Near End Longitude

Reciprocal End Latitude

Reciprocal End Longitude

Near End displaced threshold

Reciprocal End displaced Threshold

TACAN Record

```
NAV NBE TACAN 32-44-05.152N 096-58-00.211W 463 08E 108.3 020X
```

Fields

Record Identifier: 'NAV'

Three/Four Letter identifier

NAVAID Type: 'TACAN'

Latitude

Transmitter Longitude

Transmitter Altitude (ft, MSL)

Magnetic variation at station (degrees east or west)

VHF Frequency (MHz)

TACAN channel number

ILS Record

```
ILS 30 ILS IAIW 108.9 34-18-47.161N 097-01-38.280W 34-17-59.533N 097-00-35.018W 729.2
06E 4.92 309 3.00
```

Fields

Record Identifier: 'ILS'

Runway Identifier:

ILS Type: either 'ILS', 'ILS/DME', 'LOCALIZER', or 'LOC/DME'

Four letter Identifier

VHF Frequency (MHz)

Localizer Transmitter Latitude

Localizer Transmitter Longitude

Glide Slope Transmitter Latitude ('-' for Localizer-only approaches)

Glide Slope Transmitter Longitude ('-' for Localizer-only approaches)

Glide Slope transmitter altitude (ft, MSL) ('-' for Localizer-only approaches)

Magnetic variation at localizer station (degrees, east or west)

Localizer width (degrees)

Localizer bearing (degrees, magnetic)

Glide slope angle (degrees) ('-' for Localizer-only approaches)

Features

Use feature records to place objects in an ACM scene.

Feature Record

```
tower.obv 32-58-04.800N 096-50-16.800W 644 0
```

Fields

Object filename

Latitude

Longitude

Altitude (ft, MSL)

Orientation angle (degrees, true)

Defining New Aircraft

The stout-of-heart may be interested in creating new aircraft types. Some of this information must be generated by hand, but I did create a program to help me generate aircraft objects: GEDIT. GEDIT is a rudimentary Motif program that allows you to create 3-dimensional objects. It is available from me.

The syntax of this file is not complex.

```
# Aircraft Inventory for ACM 5.0
# See ACM-Bibliography for further information about the sources of this
# information.
# MiG-29 Fulcrum
# MiG-29M information was derived from two sources: [Spick87] and [AirI Aug92].

aircraft "MiG-29" {
       Object        "mig29.obj"
```

The visual description of an ACM aircraft is stored separately in V-library format. This file describes a set of polygons that roughly approximates the shape of the actual air plane. I used GEDIT to create most of the ACM aircraft objects.

```
       WingArea      400.0  # (wingS) Wing surface area (ft^2)
       WingHalfSpan 18.87   # (wings) Wing half-span (ft)
       Chord         9.61   # (c) Mean geometric chord of wing (ft)
       AspectRatio   3.56   # (aspectRatio) aspect ratio
       EmptyWeight   22500.0 #(emptyWeight) Empty weight (lb)
```

These values are usually quoted with an aircraft's performance figures.

```
       # (I(x,y,z)) Moments of inertia
       Ixx           10000.0
       Iyy           75000.0
       Izz           80000.0
```

These are hard values to simple guess. I have grabbed values where they were available and simply guessed sometimes, too.

Moments of inertia in an actual aircraft are not constant. Most notably, as an aircraft burns fuel, it's weight distribution, center-of-gravity, and moments of inertia change. In ACM, however, moments of inertia and CG do not change.

```
       # (cFlap) Lift due to flaps (yields Clift of 1.0 at max extension)
```

```
CFlap          0.64279
CLNegStall     17.0    # (deg)
CLPosStall     30.0    # (deg)
BetaStall      15.0    # (deg)
CFlapDrag      0.0467 # (cFlapDrag) Drag due to Flaps
CGearDrag      0.03    # (cGearDrag) Drag due to Gear
CSpeedBrake    0.03    # (cSpeedBrake) Drag due to Speed Brake
                       # at 90 degrees


# (CDb) Drag Characteristic equation [Wave+body]
# independent variable is mach number

CDb     {              # (CDb) Drag Characteristic equation [Wave+body]
                       # independent variable is mach number
        0.0,    0.020,
        0.2,    0.020,
        0.95,   0.015,
        1.05,   0.045,
        2.00,   0.030,
        10.0,   0.030
        }
```

In ACM, wave/body drag coefficient is a function of the airspeed expressed as a Mach number. Less drag makes for a faster aircraft. This function generally has a maximum near Mach 1.0. You probably should not change this table unless you really know what you're doing.

```
CLift   {              # (CLift) Lift-slope curve (vs alpha)
        -3.0,   0.0,
        -1.05,  0.0,
        -0.576, -2.199,
        -0.556, -2.23,
        -0.524, -2.199,
        0.524,  2.199,
        0.556,  2.23,
        0.576,  2.199,
        1.05,   0.0,
        2.0,    0.0
        }
```

Lift is defined by the equation :

<Picture: (Figure)>

Where r is the air density, S is the wing reference area (ft^2), and V is the airspeed (ft/ sec), and CL is the coefficient of lift.

In ACM, the coefficient of lift is purely a function of angle of attack (alpha).

FIGURE 10 LIFT COEFFICIENT TABLE

<Picture: (Figure)>

```
CnBeta {              # (CnBeta) Yaw due to sideslip equation
       0,     -0.08125,
       0.523540, -0.078125,
       0.785340, -0.0609375,
       1.047198, 0.125,
       1.58,  0.0
       }
```

In ACM negative yaw due to sideslip means that the aircraft is "weathercock stable", that is, the plane tends to yaw in a way that reduces the sideslip component.

```
ClBeta {            # (ClBeta) Roll due to sideslip equation
    0,    -0.0125,
    0.43633, -0.015,
    0.78540, 0.125,
    1.58,  0.0,
    3.142, 0.125,



    }
```

A negative value means that the aircraft will roll in the opposite direction of the sideslip component, tending to make the aircraft more stable.

```
CDBOrigin      0      # (CDBOrigin, CDBFactor) Drag due to sideslip
CDBFactor      0.5
CDBPhase       0      # (CDBPhase) sideslip drag phase (deg)
```

CDB values define the body drag added by sideslip on the aircraft.

```
CYBeta         -0.85  # (CYbeta) Side-force from side-slip [dCY/dBeta]
```

CYBeta is the "lift" due to sideslip. This is the factor that allows to roll 90 degrees from level, kick the rudder a lot, and still stay level. Not all simulators take this into account.

```
EffElevator    0.60   # (effElevator) Elevator effectiveness
EffRudder      0.35   # (effRudder) Rudder effectiveness
MaxRudder      20     # (maxRudder) max Rudder (deg)
SeTrimTakeoff 0       # (SeTrimTakeoff) Elevator trim at takeoff (deg)
MaxAileron     20     # (maxAileron) max Aileron (deg)
MaxFlap        20     # (maxFlap) max flap setting (deg)

# (flapRate) flap extension rate (about 2 secs to fully extend flaps)

FlapRate       10     # (deg/sec)

# (gearRate) gear extension rate (about 3 secs to fully extend gear)

GearRate       30     # (deg/sec)
MaxSpeedBrake 80      # (maxSpeedBrake) max Speed Brake setting (deg)

# (speedBrakeRate) rate of speed brake extension (2 secs to full ext)

SpeedBrakeRate 40     # (deg/sec)

# (speedBrakeIncr) speed Brake increment per keystroke

SpeedBrakeIncr 80     # (deg)
```

These values determine the characteristics of the aircraft's speed brakes (speed brakes are used in the air, wheel brakes are used on the ground). MaxSpeedBrake deter mines the maximum extension of the speed brake panels. SpeedBrakeIncr deter mines how far one press of the (S) key will increase the speed brake extension. It does not need to be an even increment of MaxSpeedBrake. SpeedBrakeRate deter mines how fast the brakes will deploy or retract.

```
Clda          0.048  # (Clda) roll moment from aileron offset
Cldr          0.004  # (Cldr) roll moment from rudder offset
Clp           -0.27  # (Clp) roll damping
Cmq           -8.0   # (Cmq) pitch damping factor
Cnr           -2.0   # (Cnr) yaw damping factor
```

These factors characterize some of the roll, pitch and yaw characteristics of the aircraft. They are in the NACA form. Damping factors determine how quickly an aircraft returns to a steady state after some change in control input. Larger negative damping factors make for a more stable aircraft.

```
CmAlpha       -0.30  # (cmSlope) CmAlpha curve slope
CmFactor      -0.20  # (cmFactor) CmAlpha factor when stalled
MaxFuel       9750   # (maxFuel) maximum internal fuel (lb)

# engine lag factor (how fast does it respond to throttle changes

EngineLag     -3.0

# Engine data based on updated RD-33K engines cited in [AirI Aug89].
# (maxThrust) max static thrust, military power (lb)

MaxThrust     27000

# (maxABThrust) max static thrust, afterburner on (lb)

MaxABThrust   44000
```

Maximum static military and afterburner thrust values can usually be found in documents that describe aircraft performance.

```
Thrust {              # (Thrust) Mach Number vs. thrust
       0,     1,
       1.7,   1.5,
       2.0,   0.84,
       5,     0.5
       }
ABThrust {    # (ABThrust) afterburner thrust table
       0,     1,
       0.5,   1,
       1,     1.21,
       1.7,   1.7,
       5,     1.64
       }
```

The Thrust and ABThrust corrects the static thrust value for the aircraft's Mach number. Unless you

have hard information about this, leave these fields alone.

```
# (spFuelConsump) specific fuel consump(lb fuel/lb T x hr)

SpFuelConsump        0.68

# (spABFuelConsump) AB specific fuel consump(lb fuel/lb T x hr)

SpABFuelConsump      2.55
```

These fields define the specific fuel consumption (fuel consumed per pound of thrust per hour). Unless you have hard information about this, leave these fields alone.

```
# (groundingPoint) hypothetical single point of contact w/ground

GroundingPoint       { 0, 0, 4.222 }
```

GroundingPoint is best calculated using tools/balance.c. It defines a good initial guess of the "rest" Z position for a newly created aircraft. To be more accurate, I should make this a singe Z value and then add an initial rest pitch, as well. Since I don't, this is why you see an aircraft pitch a bit just after it is created.

```
# (viewPoint) pilot's viewing location wrt CG

ViewPoint            { 14.75, 0, -5.375 }
```

This is the XYZ location relative to the aircraft center of gravity that locates the pilot's eye view outside of the cockpit.

```
# Landing Gear control parameters

MuStatic      0.08   # (muStatic) static coeff of friction no-brakes
MuKinetic     0.05   # (muKinetic) moving coeff of friction no-brakes
MuBStatic     0.7    # (muBStatic) static brakes-on
MuBKinetic    0.6    # (muBKinetic) moving brakes-on
```

These four values characterize the amount of friction that the wheels generate when in contact with the ground. The maximum theoretical value of any of these is 1.0.

```
MaxNWDef      72     # (maxNWDef) maximum nosewheel deflection (deg)
NWIncr        0.0397 # (NWIncr) deflection for each unit (1.27/32)
MaxNWS        85     # (maxNWS) maximum NWS velocity (fps)
```

You can probably just leave these three nose-wheel steering parameters alone.

```
# (rm) location of main gear attachments

Rm              { -2.206000, 0.000000, 1.702 }

# (rn) location of nose gear attachments

Rn              { 9.395000, 0.000000, 1.807 }

Dm              10000  # main oleo damping factor (lbs-ft/sec)
Dn              80     # nose oleo damping factor (lbs-ft/sec)

Km              18220.7 # main oleo spring factor
Kn              4278.34 # nose oleo spring factor

Gm              1.5    # main strut length with tire (ft)
Gn              1.5    # nose strut length with tire (ft)

CmMax           1.0    # (cmMax) main max oleo extension distance (ft)
CnMax           1.5    # (cnMax) nose max oleo extension distance (ft)
```

Landing gear parameters are difficult to estimate. GEDIT can be used to determine cor rect landing gear locations. I then use the program tools/balance.c to generate the appropriate values for these factors for each new aircraft type that I create. Note that neither GEDIT nor balance.c currently calculate oleo damping factors.

```
# (tailExtent) as we rotate, this part may drag

TailExtent            { -18.165, 0.0, 1.191 }
```

This defines where the aft-most lower part of the aircraft is located. This is the part of the plane that may scrape the ground as we rotate. This value is not currently used. Instead we simply limit up-pitch on the ground to 20 degrees.

```
StructurePoints       15     # (structurePts) maximum structural damage
```

This value characterizes how much damage can be absorbed by the plane before it sim ply explodes.

```
# Radar data based on N-019 Pulse Doppler radar cited in [AirI Aug89].

RadarOutput           15000  # (radarOutput) radar output (watts)
RadarTRange           38     # (radarTRange) tracking radar range (nm)
```

```
RadarDRange          55     # (radarDRange) detection radar range (nm)
TEWSThreshold        0      # Radar Warning Receiver threshold (watts)
```

If you have any information about the radar capabilities of the aircraft, here's the place for them. The detection range is the maximum range that a target can be seen on radar. No attempt is made to take into account the radar cross section of the target. The track ing range is the range required to get a lock onto the target.

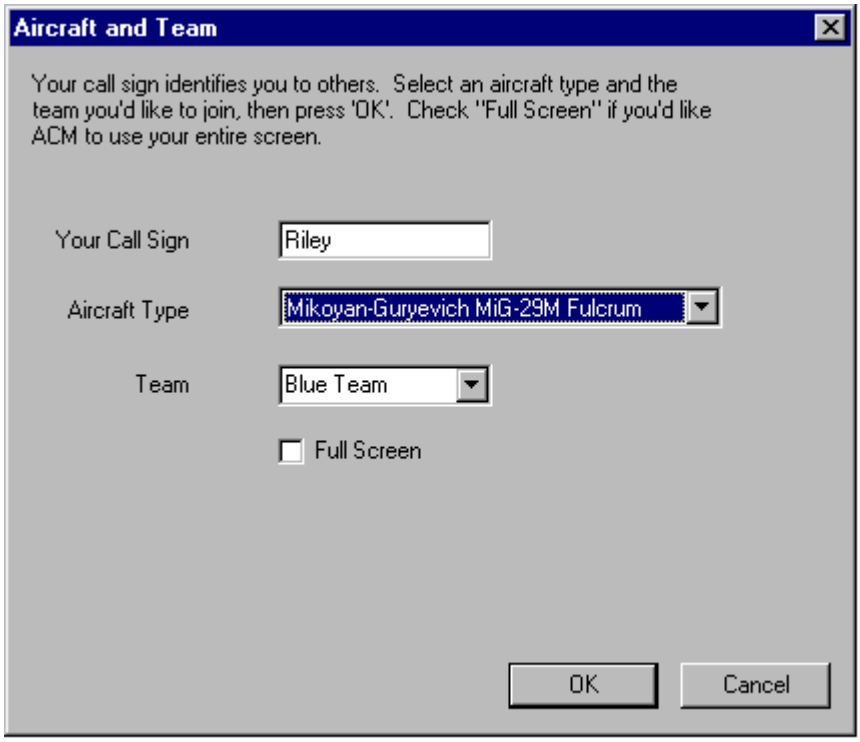
```
# Weapons

WeaponCount 9
HardPoint0 { 7.0, -4.0, 0.0 }
HardPoint1 { 0.357, 15.6, 0.0 }
HardPoint2 { 0.357, -15.6, 0.0 }
HardPoint3 { 1.5, 9.0, 2.0 }
HardPoint4 { 1.5, -9.0, 2.0 }
HardPoint5 { 1.5, 8.0, 1.5 }
HardPoint6 { 1.5, -8.0, 1.5 }
HardPoint7 { 1.5, 10.0, 1.5 }
HardPoint8 { 1.5, -10.0, 1.5 }
```

These are the XYZ locations relative to the aircraft CG of each weapon. By defining these, each missile or cannon fires from its appropriate location on the aircraft.

```
WeaponStation 0 "m61a1" 500   0   0
WeaponStation 1 "aim9m" 0   0   0
WeaponStation 2 "aim9m" 0   0   0
WeaponStation 3 "aim9m" 0   0   0
WeaponStation 4 "aim9m" 0   0   0
WeaponStation 5 "aim9m" 0   0   0
WeaponStation 6 "aim9m" 0   0   0
WeaponStation 7 "aim9m" 0   0   0
WeaponStation 8 "aim9m" 0   0   0
```

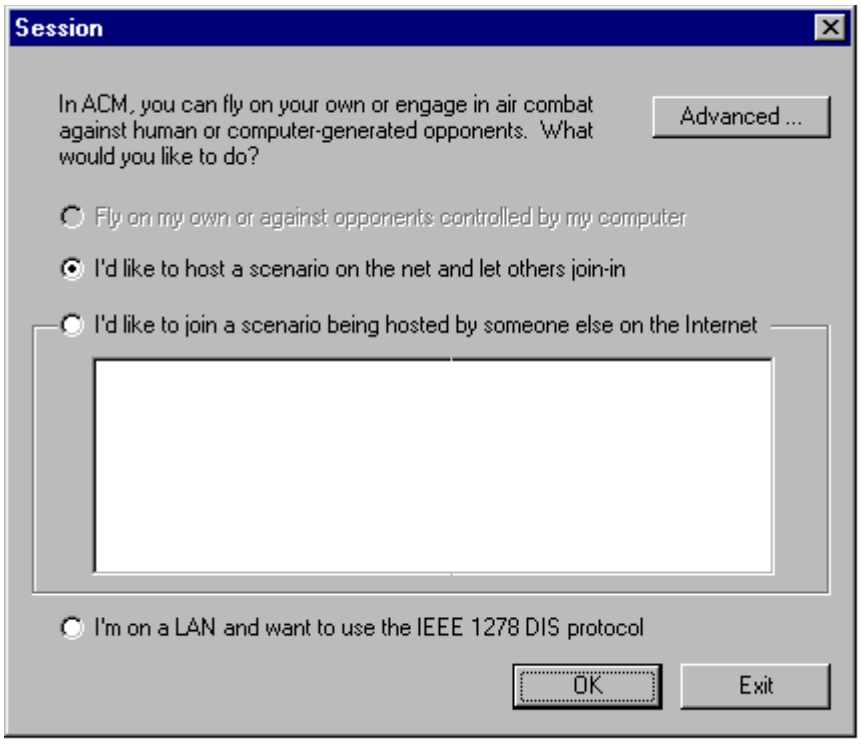
The WeaponStation directive defines the type of weapon located at each hard point. The only currently defined weapon types are "m61a1" for Vulcan cannons and "aim9m" for Sidewinder missiles. Three extra weapon-defined integer fields are provided. These fields have no meaning for "aim9m" missiles. The first field defines the initial number of rounds supplied to "m61a1" cannons.

Starting ACM for Windows



Call Sign	Although this is not completely implemented, yet, you may register your aircraft's call sign – which will used to identify you to other players.
Aircraft Type	You may select either a MiG-29M or F-16C Falcon.
Team	Team locations are defined in the scene file. In the scene file shipped with ACM, team one is based at Addison airport in Dallas, Texas. Team two is based a Sheppard Air Force Base, Wicita Falls, Texas.
Full Screen	If you select full-screen mode, ACM will occupy the full screen (surprise). Currently, the full screen layout is 800x600 and 256 colors.

Setting up an ACM session (ACM for Windows only)



Advanced	Use this button to select a different DirectPlay service provider (modem, TCP/IP, direct connect serial, or IPX) or to set IEEE 1278 DIS parameters.