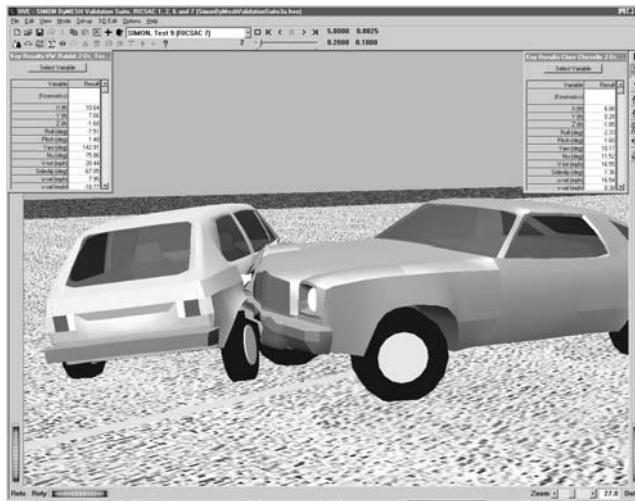


Technical Newsletter

On-going Developments In HVE and HVE-2D

EDC's programmers and test engineers have been hard at work making significant improvements in the robustness and functionality of the HVE and HVE-2D code. Implementing and testing these improvements has resulted in delayed release dates for HVE v4.1 and HVE-2D v2.0. However, the latest updates provide users with greater efficiency and functionality when using their programs for their work. Here is a brief overview of the developments in HVE and HVE-2D.



Example of HVE's 3-D Event Viewer displaying a staged crash simulation (RICSAC7) using SIMON. DyMESH (Patent No 6,195,625) was used to calculate the 3-D collision forces and moments. DyMESH is in its final stages of validation.

HVE Version 4.1

HVE users have recently received their HVE Version 4.1 updates. This latest update provides improvements to the user interface, as well as the following new enhancements:

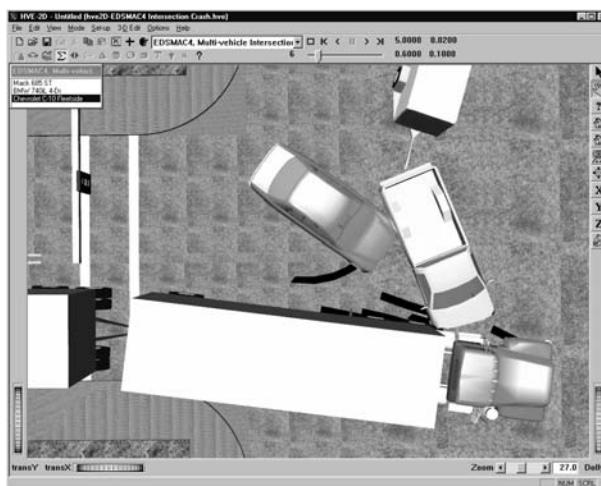
- Support for new EDCRASH features
- Support for more than one steerable axle on a vehicle
- Additional support for the HVE Tire Blow-out Model and Virtual Accelerometers
- New SIMON Calculations Options Dialog
- Significantly improved Installation program

HVE-2D Version 2.0

Although most HVE-2D users have not yet worked with the new user interface for Version 2.0, they are directly benefiting from the work that has gone into the HVE code over the past few months. HVE-2D users who want to know when they will receive their updates are encouraged to visit the HVE-2D FAQ page on our website, found at www.edccorp.com/products/hve2dfaqa.html.

Many HVE-2D users are looking forward to using the EDSMAC4 collision simulation model with HVE-2D Version 2.0 so they can:

- Simulate crashes involving any number of vehicles, such as a freeway pile-up
- Simulate simultaneous collisions between multiple vehicles
- Simulate crashes involving tractor-trailers
- Simulate the effects on vehicle trajectory caused by wheel displacement from a crash
- Simulate a tire blow-out
- Compare a simulated damage profile to an actual damage profile to validate inputs



Example of HVE-2D's Event Viewer displaying a multi-vehicle crash simulation using EDSMAC4. The simulation involves simultaneous vehicle collisions, as well as collisions between articulated vehicles.

Managing a Large Software Project

We thought our users might find it interesting to learn how a large software project, like HVE, is actually managed. People might think that EDC has a programmer or two with the source code on their computers and they just, well, *program*. As we'll see, running a large, professionally managed software project is not nearly that simple.

HVE currently has about 350,000 lines of code distributed among 1,176 individual source, header and resource files (we're not including any Windows-related files or other tools in these statistics). At any one time, the number of programmers working on HVE varies between four and nine (at one time during the peak of development in 1995, there were 12 programmers working on the HVE project). Some common issues arise in a project of this magnitude: What if two programmers are making changes to the same file? What if the changes made by one programmer are not compatible with the changes made by another? How do we keep track of the changes made over the course of the project?

The key to managing a large software project (and, therefore, being able to deal with questions such as those posed above) lies in designing and maintaining a strictly controlled process for managing the HVE project. The HVE project management system is designed around the following five components:

- ◆ Developer Environments
- ◆ Resource Control System
- ◆ Nightly Build
- ◆ Bug Report
- ◆ Manufacturing Area

These five components are shown graphically in Figure 1, and are described below.

Developer Environment

Each HVE programmer (or *developer*) has a developer environment (we call it a *sandbox*) installed on his/her computer. Each sandbox has a private copy of each of the 1,176 files required to build their own working version of HVE. The sandbox is where a programmer adds new code and modifies existing code. Compiling and preliminary testing are also performed in the programmer's sandbox.

Resource Control System

The resource control system (RCS) is the heart of the HVE project. RCS is like your public library:

You can check books (or in our case, *files*) in and out. When a file is checked out, the RCS sets a flag and no one else can check it out. Therefore, only one person can make changes to a given file at a time. When the file is checked in, the RCS automatically increments its version number. With over a thousand individual files being revised by several individuals, it would be impossible to manage the HVE project without the strict file version control provided by the RCS.

An HVE programmer will check out the latest version of a file from the RCS, creating a local copy of the file in their sandbox. The programmer then modifies the code as required to accomplish their goal. After modification and testing, the programmer will check the file back in to the RCS, along with a series of check-in comments describing the additions or changes made to the file. These comments are later used to compile the Release Notes. (The Release Notes is an important text file that is shipped with each version of HVE. You can review the Release Notes to learn the differences between the new and previous versions of HVE.) Using the RCS ensures that changes made by one programmer are not overwritten by changes made by another programmer.

Nightly Build

But what if one programmer makes changes that are not compatible with changes made by another programmer? With several programmers simultaneously working on the project, this is certainly a possibility. The solution to this potential problem is to build a new HVE program every night (when no one is making changes to the source files). This program is called the *nightly build*. The nightly build checks out a local copy of the latest version of each file and uses these files to build HVE. If incompatible changes were made, the nightly build fails and issues a report of the conflicts. The programmers then go back and resolve the conflicts. Fortunately, this is a rare occurrence, because the program manager instinctively knows if two programmers are likely to create conflicting changes. Programmers are assigned tasks that are in different areas of the program.

Bug Report

Several engineers at EDC actually use the nightly build whenever they run HVE. Therefore, they are constantly testing the overall quality of the code. However, there is also an official list of program bugs, called the Bug Report. Each bug is assigned a number, date, severity, priority, description and



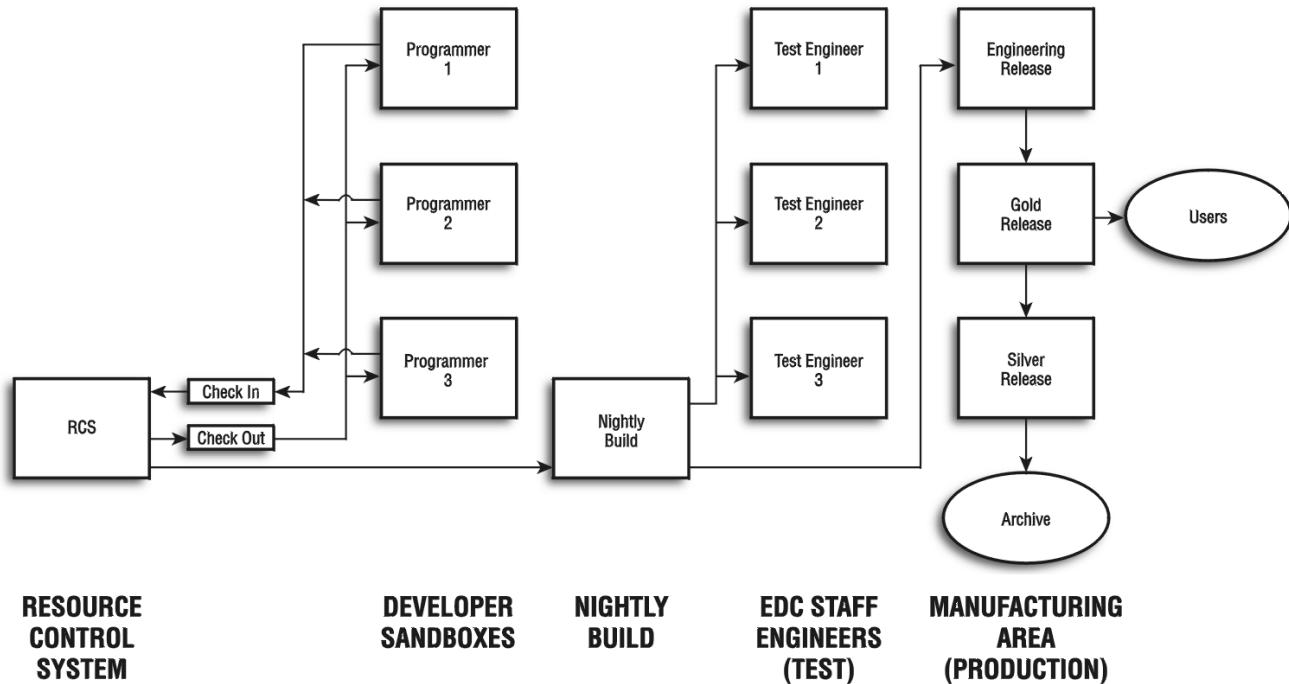


Figure 1: HVE Project Management System.

test engineer. Each week, programmers are assigned one or more bugs to resolve. On resolving the bug, the programmer identifies in the Bug Report the cause of the bug, the solution and identifies him/herself as the person fixing the bug.

All the code is normally checked in to the RCS by Sunday evening. On Monday morning, the Bug Report is distributed to the engineering staff at EDC and testing of each bug fix is performed. Bugs that are fixed are closed; bugs that require additional programming remain open and the results of testing are added to the Bug Report. This formal testing procedure concludes on Tuesday afternoon. The updated Bug Report is returned to the programmers and the report is then archived.

Each new version of HVE has associated with it a functional description that includes the program enhancements and a list of required bug fixes. This functionality is usually determined about a month before the program is actually released. This is called a *version freeze*. During this month, internal and external (Beta) testing is performed to ensure proper functionality.

Manufacturing Area

The Manufacturing Area is where the HVE executable program, resource files (e.g., Language.rsc) and data files (e.g., EDC.db, the tutorial case files, and approximately 300 other files) actually reside. These files are actually used to produce the master CD, which is duplicated and received by HVE users. Files are carefully copied

by hand from the Nightly Build to the Manufacturing Area after testing is completed.

EDC maintains three separate versions of HVE in the Manufacturing Area:

Engineering Release – This is the version that is currently under development. An engineering release may also include incremental changes to specific parts of HVE, such as the installation program. An engineering release is used for testing by outside organizations.

Gold Release – This is the production version of HVE that users see. It is created from the engineering release after testing is concluded. It is created by simply renaming the directory (i.e., from *Engineering* to *Gold*). Thus, there is no danger of human error in creating the Gold Release, because no files are copied.

Silver Release – This is the previous Gold Release version of HVE. We maintain this as an insurance policy – just in case! The Silver Release is created by simply renaming the directory (i.e., from *Gold* to *Silver*). Thus, there is no danger of human error in creating the Silver Release, again because no files are copied.

The previous Silver Release is removed from the production system and archived.

As you can see, managing a software project like HVE is an involved process. However, these procedures are essential to ensure and maintain a high quality program.

Technical Session

HVE allows tremendous flexibility for the user to control both the HVE-compatible physics simulation or reconstruction and the presentation of results from these analyses. Many of these options are found in the Simulation Controls or Calculation Options dialogs. It is important to understand the functions of these variables. In this article we take a close look at an often overlooked, yet vital, parameter: Output Time Interval.

In order to demonstrate the importance of Output Time Interval, a simple EDSMAC4 event was created. A passenger car and a pickup truck collide in an oblique collision. We are particularly interested in the collision pulse, so let's take a close look at the output.

Critical data are located in the Accident History, Damage Data, and Variable Output output reports. The Accident History output report reports $T_{\text{impact}} = 0.03$ sec and $T_{\text{separation}} = 0.13$ sec, resulting in a total collision pulse duration of 0.10 seconds (Table 1). In the Damage Data output report, Peak Acceleration for the passenger car is 40.7 g at time = 0.084 sec. These values for peak acceleration and pulse duration seem quite reasonable, but, when we graph Total Acceleration from the Variable Output Table, it looks like the pulse duration is 0.23 sec and peak acceleration is just over 10 g at time = 0.1 seconds (Fig. 2). How do we reconcile this apparent inconsistency?

Table 1: Comparison of alpha-numeric output to Variable Output graph

Variable	Alpha-numeric	Graph
T_{impact} (sec)	0.03	0.00
$T_{\text{separation}}$ (sec)	0.13	0.23
Duration (sec)	0.10	0.23
A_{max} (g)	40.7	10

The key is in choosing an appropriate Output Time Interval. The Output Time Interval is set in the Simulation Controls dialog, under the Options menu. This parameter controls the time interval at which physics, in this case EDSMAC4, reports results back to HVE. Unlike the other time intervals that are set in this dialog, the Output Time Interval does not affect the calculations within the simulation itself.

If we accept all of the default values in the Simulation Controls dialog, the Collision Time Interval = 0.001 sec and the Output Time Interval = 0.10 sec. Therefore, even though EDSMAC4 is doing calculations every 0.001 seconds during the collision phase, it is only outputting its results to

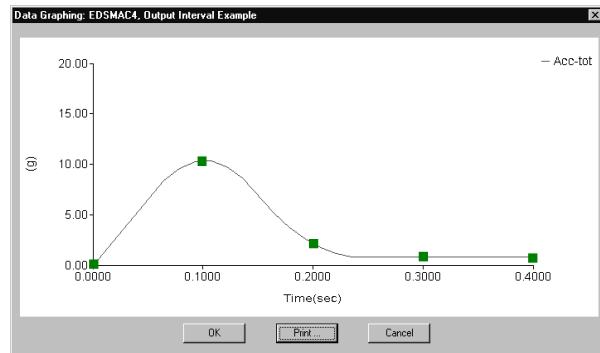


Figure 2: Total acceleration with Output Time Interval = 0.10. Solid squares are overlaid at timesteps where EDSMAC4 reported results to HVE. In the Variable Output table, HVE draws the graph based on these values.

HVE every 0.10 seconds. In other words, for every 100 calculations EDSMAC4 does, HVE receives only 1 value. You can see this when you watch an event execute. Keep your eye on the timer; it increments every 0.1 seconds, and your vehicles move every 0.1 seconds.

So, why does the graph look different from what we expect based on the Accident History and Damage Data output reports?

Alpha-numeric output reports (e.g., Accident History, Damage Data) are created by physics and critical values are stored as the simulation is executing. On the other hand,

The Variable Output table is created by HVE using the values reported by physics at the Output Time Interval (Fig 3).

HVE uses a smoothing algorithm to create a graph based on the data sent to HVE at the Output Time Interval. If things are changing very rapidly, like reaching the peak of a collision pulse, and the Output Time Interval is not sufficiently small, it is likely that key data will be missed. This is exactly what happened in Figure 2 of our example.



Figure 3: Process used to pass numeric data to the Variable Output Table.

The solution is to decrease the Output Time Interval. In this case, decreasing the Output Time Interval from the default, 0.1, to 0.001 sec produced graphical results that matched the alpha-numeric results (Fig. 4). Comparing Figures 2 and 4, it is clear that the beginning, peak, and end of the collision pulse were all missed by using an Output Time Interval of 0.1 sec.

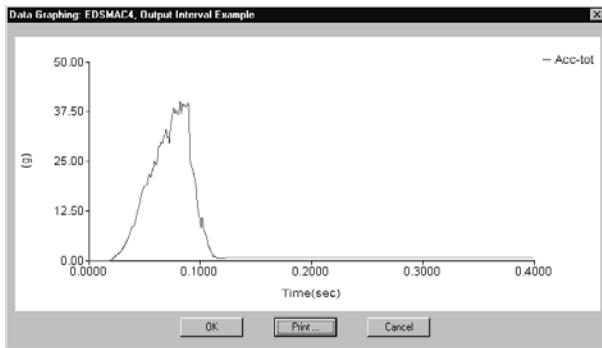


Figure 4: Total acceleration with Output Time Interval = 0.001.

Before you start decreasing the Output Time Interval in *all* of your events, however, consider the questions that you are trying to answer with your simulation. (Remember, decreasing the Output Time Interval will increase the time required to execute a simulation event because the scene is redrawn that many more times.) Are you looking for delta-V, damage, and rest positions? If so, it is probably not necessary to decrease the value. But if you are looking at the details of the collision pulse, you will have better results by decreasing the Output Time Interval. Users of occupant or pedestrian simulation models should be aware that the default Output Time Interval in these events is 0.001 sec for this very reason.

User Hints and Tips

- Want your simulations to run even faster while you are working to refine your events? Click the right mouse button and choose Draw Style – wireframe. (Or, under the Options menu choose Render and set Render Method to Wireframe.) Your humans, vehicles, and environment will be displayed as wireframes and the simulation will execute faster because of the decreased rendering time. To get it back to normal, choose Draw Style – as is. (Or, under the Options menu choose Render and set Render Method to Phong.)
- If you have humans or vehicles in a scene whose movements are just “for looks”, the simplest approach is often to use EDGEN. EDGEN is an object kinematics program that uses user entered positions and velocities to determine the path and speed of the object. Don’t try to use the steering, brake, throttle, and gear tables to back that tractor-trailer up. Instead, place the tractor-trailer at up to eight positions and let EDGEN do the rest. Best of all, EDGEN is free to all HVE users and will soon be free for all users of HVE-2D.

- Broke a Rim? Occasionally a 3-D simulation, such as EDVDS, EDVSM or SIMON, will terminate with the message “Excessive Tire Deflection! (Broke a Rim?).” This typically occurs when a vehicle’s tire strikes a curb. To understand why the simulation terminated (and this message is issued) requires a basic understanding of how 3-D simulations calculate radial tire force. These simulation models begin by calculating the earth-fixed coordinates of the center of the wheel, then the earth-fixed coordinates of the “bottom” of the undeflected tire (“bottom” is defined in the direction of the radial tire vector normal to the terrain). The distance between these two earth-fixed coordinate positions is the tire deflection. Multiply the deflection by the tire’s radial stiffness and you have the radial force acting on the tire normal to the surface. Of course, the tire’s radial stiffness is not a simple linear constant. In fact, it is a 2-stage linear spring with a knee at about 80 percent of the tire’s section height (Figure 5).

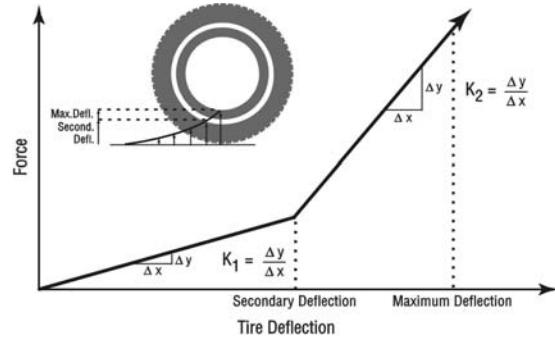


Figure 5: Force vs Tire Deflection curve.

Obviously, the forces change drastically when the deflection reaches the tire’s section height. At that point, the tire deflection has reached a maximum. Any further deformation is not deforming the tire, it is deforming the rim. Of course, the rim does not deform in a linear manner, like the tire does. Depending on the rim’s material and structural characteristics, it might either bend in a non-uniform manner, or it might simply fracture. These behaviors are quite different from the elastic behavior of the tire, and would require a finite element simulation to predict the resulting forces and moments. This is, of course, completely beyond the scope of vehicle dynamics simulation. Therefore, the simulation terminates with the message “Excessive Tire Deflection! (Broke a Rim?).”

It is rather easy to predict that a curb impact by a tire might terminate a simulation: Simply compare the curb height with the tire's section height. If the curb height is greater, chances are quite good that the simulation will terminate.

If the curb height is only slightly greater than the section height, the termination can be avoided by editing the tire's section height. This is done in the Vehicle Editor. Simply click on the desired wheel, choose Tire, select Physical Data, and increase the Maximum Deflection. This should only be done when necessary, and the increase should be nominal, say an inch or two. Obviously, if the actual event you are simulating really did result in a broken rim, you must recognize that no vehicle dynamics simulation can rigorously account for it. However, your simulation might provide some insight.

Another interesting condition can also terminate a simulation with the same message. That condition occurs when the terrain elevation is below the origin (i.e., its Z coordinates are positive), and the tire encounters a hole in the terrain. The hole is assigned a default elevation of 0.0, thus there is an elevation change. If the elevation change is greater than the tire's section height, the result is the same – for exactly the same reasons as outlined above. In this case, the solution is to find the hole in the terrain and repair it.

HVE and HVE-2D FAQ

This section contains answers to frequently asked questions submitted to our Technical Support staff by HVE and HVE-2D users.

Q: *I am locking up my tires in a 3D physics program, but I don't see skidmarks. How could this be?*

A: This is a fairly common question, especially for unloaded or lightly-loaded tractors (for the reason described below).

The programming logic that determines whether or not a tire leaves a skidmark is complex and varies between physics programs (e.g. EDSMAC4 is different from EDVSM). Refer to your Physics Manual, Calculation Method chapters for the details.

In general, however, the most common reason that a skidmark is not left when the user

believes that it should be is that the vertical tire force, F_z , is not large enough. All 3D tire models require a minimum vertical tire force to leave a skidmark. This minimum load value, "Minimum F_z for Skidmark", is a user-editable parameter found in the Tire, Physical Data dialog.

Q: *When I use the HVE Path Follower, my vehicle does not go toward the target vehicle. Why not?*

A: The HVE Path Follower allows the user to specify a desired vehicle path and attempts to force the vehicle, subject to the constraints of physics, to drive through those targets. Sometimes the vehicle cannot match the path because the maneuver is simply impossible for the vehicle or the driver or both. Other times, however, the vehicle may not follow the path because it does not "see" where it is intended to go.

The HVE Path Follower uses the user-editable value of Driver Preview Time (Driver Controls, Path Follower, Driver Data) to specify how far ahead, in time, the simulated driver should look for the path. The default value is 1 second.

Imagine that you want to drive in a circle at 55 mph. How far will you travel in 1 second? 80 feet. That means that at any timestep the model will ignore anything happening for the next 80 feet. It is therefore recommended that, instead of beginning the simulation right on the circle, you begin the simulation on a tangent to the circle about 80 feet away. Then the vehicle and driver will have plenty of time to ease into the turn.

Q: *When using EDCRASH, what is the difference between the Δ -V displayed in the Event Editor Damage Profiles dialog and the value in the Accident History output report?*

A: The Δ -V value displayed in the Damage Profiles dialog is the Δ -V that would have caused the entered damage had the vehicle hit a rigid barrier. Because the barrier would completely stop the vehicle, this is also the barrier equivalent velocity (or energy equivalent speed).

The Δ -V values reported in the Accident History report are the values computed by EDCRASH for the vehicle-to-vehicle collision. This calculation is based on either damage data or linear momentum, depending on the data that you entered and the details of the collision. Because the vehicle that you hit is neither rigid nor stationary (barriers are not allowed in EDCRASH), this value of Δ -V will always be lower than the barrier-based Δ -V.



EDC Vehicle Database Continues to Grow

Over the past 12 months, EDC has built 26 custom vehicle models for HVE users who required a specific vehicle for their casework. These vehicles were also added to the Vehicle Database for HVE, adding 187 vehicle model years to the available selection when all model years are considered. EDVDB-3D now contains over 800 vehicles for use in HVE simulations.

As with all vehicles in the HVE Vehicle Database, a complete vehicle consists of both a vehicle geometry file and a vehicle data file. When EDC builds a custom vehicle for a user, all of the data for these files are measured from an actual exemplar vehicle. All of the measured data is subjected to an intense quality control process to ensure the data is correctly entered in the data set for the vehicle. For more information about the measurement of the data used in a vehicle model, please contact EDC Customer Service and request a copy of technical paper WP# 2000-6 "Building Vehicles For HVE".

HVE users are encouraged to order at least two custom vehicles every year. Why? Because the purchase of a custom vehicle (typically around \$950.00) can be applied directly towards the renewal of their annual Update/Technical Support for the EDVDB-3D vehicle database. This policy allows HVE users to have the vehicles they need for their own cases, as well as all of the other vehicles added to the database between updates.

HVE-2D users who own the EDVDB vehicle database will notice a significant increase in the number of vehicles available for their work when they receive their HVE-2D Version 2.0 updates. The vehicle data displayed in the Vehicle Editor in HVE-2D is a subset of data (applicable for 2D simulations and reconstructions) taken directly from the complete vehicle data set for a HVE vehicle. This includes the geometry file of the vehicle.

HVE and HVE-2D users who want to order a custom vehicle should contact EDC Customer Service for a Custom Vehicle Order Form. Once we receive your completed order form, we will begin an extensive search for an exemplar vehicle to meet your requirements. Please place your order well before your required date, as it may take some time to locate an exemplar vehicle and other orders may be ahead of yours.

Visit the EDC Website at www.edccorp.com

The latest information about our products and services can be found on our website at www.edccorp.com. Please note that the domain is spelled e d c c o r p. Several users have tried to visit our website recently entered edcorp (missed typing the second c in the domain name) and ended up at a website that is "Under Construction". Our website has been up and running continuously for five years and waiting for you to visit, so if you see this message, please check the address you entered and try again.

HVE-2D EDKEYS

Several HVE-2D users have recently contacted EDC Customer Service and asked "Does the new HVE-2D Version 2.0 use the same EDKEY that I presently use, or will it use a new one?" The answer is that HVE-2D Version 2.0 does use a different EDKEY than HVE-2D Version 1.33.

The new EDKEY is smaller in physical size and is also fully compatible with the faster (> 700 MHz) computers available today. HVE-2D users who want to continue to have the flexibility to install their HVE-2D Version 2.0 software on their laptop and desktop and simply move the EDKEY to whichever computer they are currently using are encouraged to place their order for their new EDKEY.

If you have not already placed your order for the new EDKEY or to add EDSMAC4 to your HVE-2D Version 2.0 update, please contact EDC Customer Service for more information.

Call for Papers

HVE users interested in presenting a technical paper in the "HVE White Paper" session for the 2002 HVE Forum are invited to submit an abstract for consideration. This session is an opportunity for you to showcase your skills to other HVE users as well as to non-HVE users who may wish to hire you as a consultant.

Please submit your abstract, of 100 to 250 words in length, to EDC Customer Service before November 15, 2002. Please include complete contact information for the lead author with each abstract submission. If you would like to discuss possible topics for your paper, please contact EDC Customer Service at 503.644.4500, or send an email to forum@edccorp.com.



Training Courses

EDC Reconstruction

EDC offers an excellent 1-week training course on the use of the EDC reconstruction software program, EDCRASH. Even long-time users agree that the course is extremely beneficial and challenging.

The course is designed to investigate fully the program's inner workings. Lectures are full of helpful hints gained from years of experience.

EDC Simulation

EDC offers an excellent 1-week training course on the use of EDC simulation software, such as EDSMAC, EDSMAC4, EDSVS and EDVTS.

The last EDC Simulations course of 2001 will be held this November on the University of Miami, Coral Gables campus. Those wishing to attend are encouraged to sign up as soon as possible, as there are still a few openings available. Please note that the EDC Simulations course will not be offered again until January 2003.

HVE Forum

The HVE Forum is an excellent opportunity for HVE, HVE-2D and EDVAP users to jump to a new level of ability. By participating in workshops, attendees brush up on their present skills, learn new techniques of applying simulation modeling, and learn how to use the latest advancements in the software. The HVE Forum also presents a great opportunity to meet other users and expand your network of resources and information.

Engineering Dynamics Corporation Training Courses

EDC Simulations

Los Angeles, CA.....January, 2003
Coral Gables, FL.....November 13-17, 2001

EDC Reconstruction

Los Angeles, CA.....January 21-25, 2002
Coral Gables, FL.....November 11-15, 2002

Theoretical and Applied Vehicle Dynamics

Raleigh, NC.....Spring 2002
New Orleans, LA (Short Course).....May 13-14, 2002

2002 HVE Forum

New Orleans, LA.....May 13-17, 2002

Vehicle Dynamics

Theoretical and Applied Vehicle Dynamics extends the theory of the SAE course and includes direct applications using the HVE three-dimensional simulation environment as well as a solid theoretical background for such simulations. The course is focused towards vehicle design engineers and safety researchers with an interest in a greater understanding of vehicle dynamics and automotive chassis systems development.

Course Registration

Our courses are often full, so be sure to register early to avoid missing out! You may register for a course simply by contacting EDC Customer Service at 503.644.4500, or by sending an email to training@edccorp.com. You can also visit the Education & Training pages on our website and download a course registration form. We look forward to seeing you at our next course!

NUCPs Training

Northwestern University Center for Public Safety, NUCPS (previously NUTI), offers four different courses using EDVAP and/or HVE-2D. For more information about the courses and registration, please contact NUCPS at 800.323.4011.

Northwestern University Center for Public Safety Training Courses MATAR (EDCRASH)

Evanston, IL September 2002

COMPTAR I (EDCRASH)

Evanston, IL September 2002

COMPTAR III (EDSMAC)

Evanston, IL October 2002

COMPTAR IV (EDCRASH & EDSMAC Extended)

Evanston, IL Call for dates

Engineering Dynamics Corporation
8625 SW Cascade Boulevard, Suite 200
Beaverton, Oregon 97008-7100 USA
Phone 503.644.4500 / FAX 503.526.0905
Email: info@edccorp.com
Website: www.edccorp.com

Windows is a trademark of Microsoft Corporation.

GATB is a trademark of Collision Engineering Associates, Inc.

EDVAP, EDCRASH, EDSMAC, EDCAD, EDSVS, EDVTS, EDHIS, EDVSM, EDVDS, EDGEN, EDVDB, `getSurfaceInfo()`, HVE-2D, HVE and HVE Brake Designer are trademarks of Engineering Dynamics Corporation. All Rights Reserved.

SIMON and DyMESH (Patent No. 6,195,625) are registered trademarks of Engineering Dynamics Corporation. All Rights Reserved.

