Rich Gallagher
Technical Writing and Training Portfolio

Contents:

R.S. Gallagher and Associates
PO Box 4023, Ithaca, NY 14852-4023 USA
Phone: 607-564-9878, E-mail: rich@rsgallagher.com
www.rsgallagher.com/writer

CHAPTER
8

Geometry

8.1 MSC/Ultima Geometry

Geometry models are at the core of most MSC/Ultima design analysis problems. MSC/Ultima is designed to help you create or import geometry, and then simulate how it behaves. This chapter describes the capabilities in MSC/Ultima for creating or modifying geometry for use in design analysis.

In some cases, the geometry you work with may be created beforehand in another software environment such as a CAD system. Many of these models can be imported into MSC/Ultima using the File/Import menu, described in the MSC/Ultima Utilities chapter. In many cases, these models may need further modeling work so that they can be used in analysis. In other cases, you may wish to create your own geometric model from scratch, or modify an existing geometry. MSC/Ultima combines many of the most commonly used features to build or change geometry in a single menu form that you can use in either of these situations.

8.2 Understanding MSC/Ultima Geometry

Geometric models in MSC/Ultima are based around individual geometric components that can be created or modified. There are five basic building blocks used in MSC/Ultima geometry:

- Points - 0D entities with a single location in space.
- Curves - 1D entities such as a line, curve or surface edge.
- Surfaces - 2D entities such as a plane, surface or solid face.
- Solids - 3D solid entities.
- Coordinate Systems - coordinate axis definitions used in the construction of geometry.

Most MSC/Ultima geometry entities are parametric, which means that they can be described by a local coordinate system. In the case of a curve, you can view its parametric coordinate as being analogous to arc length: for example, a parametric coordinate value of .4 is equivalent to a point located at 40 per cent of the arc length between one point and another. Similarly, surfaces and solids can have any location within their geometry described as a set of parametric coordinates. As a user, the importance of parametric coordinates lies primarily in two areas:

1. In many of MSC/Ultima's modeling capabilities, parametric values can be used to specify exact geometric entities as a function of other geometry.
2. Entities at one parametric level can often be used to create entities at another level. For example, a curve can be extracted from the edge of a solid, and a surface can be generated between two curves.

Note: MSC/Ultima geometry is parametric in the sense of being defined in a parametric coordinate system. It is not geometry defined in terms of variables that can later be modified, as is the case with some definitions of "parameterized" models.

Coordinate systems are not geometry per se, but are often very important parts of the geometric modeling process. There is always a default coordinate system, labeled coordinate system 0, which is a global Cartesian system located at the model origin. In addition, you are free to create and specify your own coordinate systems with specific origin locations and directions. Proper use of coordinate systems can vastly simplify the process of creating 3D geometry, loads and boundary conditions.

Whether your geometric model is originally imported from a CAD system, or created directly within MSC/Ultima, these same geometry building blocks are the components you use to produce a model that is suitable for design analysis. However, creating an MSC/Ultima design model is a process that involves more than simple physical shape; your goal is to produce a model that accurately represents the physical behavior you are simulating. The next section discusses some tips for creating an accurate analysis model.

8.3 Creating Geometry For Analysis

When you build geometry in MSC/Ultima, your most important objective is to create a model that accurately represents your physical model and can be analyzed. As a practical matter, this means that your model must meet certain criteria that are often more stringent than those of other CAD applications. These include:

1. Complete geometry. If a model of a solid is simply represented by its outside visible surfaces, it may look "solid," but will not analyze correctly. The analysis would reflect the behavior of a hollow object rather than a solid one. An analysis model for a solid component must contain actual solids, so that the finite elements created for the model are solid as well. Similarly, problems such as missing edges or faces can prevent an accurate analysis model from being created.

2. Congruent geometry. In a design drawing, an infinitesimal crack between two adjacent regions may be of little consequence. In an analysis model, this same crack may cause an analysis to behave incorrectly, or even cause part of the model to fly off into space in what is known as rigid body motion.

Beyond cracks and gaps, it is also important that parts mate along common borders, with no discontinuities. For example, you may have a geometric model where three surfaces meet as shown in the figure.

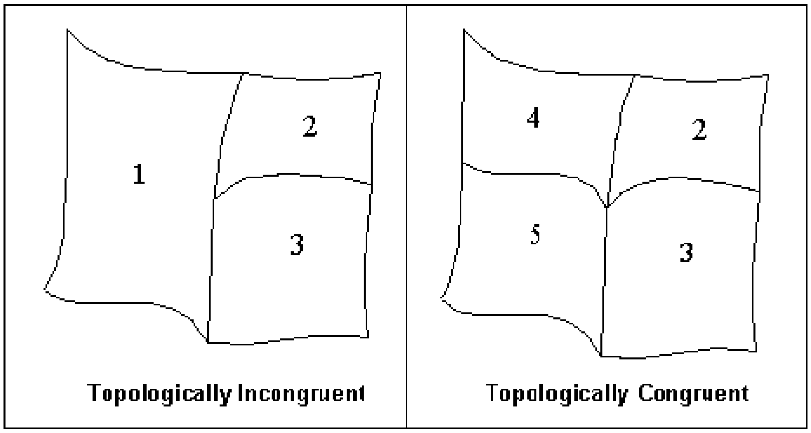


Figure 8-1 Maintaining topological congruency

This discontinuity can present serious problems for an analysis, because each surface is evenly subdivided into *finite elements* prior to analysis. This subdivision can result in an unequal number of elements along the boundary: This may cause the analysis results to be incorrect, because these elements may not share common vertices. This means that in MSC/Ultima, you should instead construct these surfaces so that every edge shares a common border, as shown in the examples in the figure.

3. Analysis efficiency. Imagine that you have a CAD model of a plate, and at one end of the plate are 20 tiny bolt holes in are region that you know will contain very little stress. If you are doing a structural analysis model of this plate, you might first remove these bolt holes to improve the efficiency of the analysis.

When you perform an analysis in MSC/Ultima, it creates a mesh of finite elements encompassing the interior of your model. When your geometry is complex, both the analysis and the meshing process itself may take extraordinarily long, as the program tries to satisfy the model's constraints. For this reason, you may wish to either remove these bolt holes prior to the import of your CAD model, or use MSC/Ultima to remove them after import.

In a similar sense, nearly any geometric model from a CAD system must be evaluated from an engineering perspective, to see if it will produce an accurate and an efficient analysis. Then you may want to modify this geometry to create a better model for analysis, using a combination of your CAD environment and/or MSC/Ultima's geometric modeling tools.

4. Engineering judgement. Perhaps the most important factor in the success of a design analysis is the engineering knowledge that you bring to the problem. A geometric model of a problem could be highly accurate, or completely wrong, depending upon how well it relates to the actual physical model.

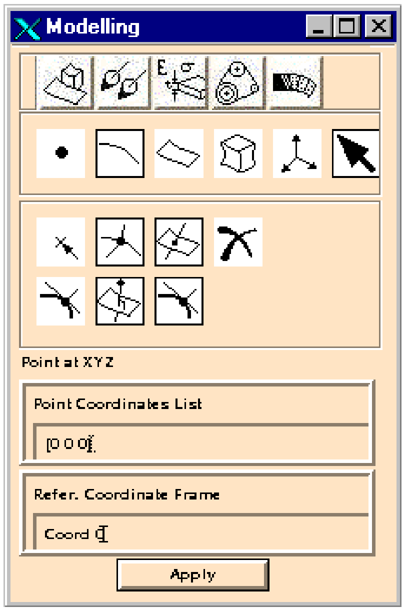
This list is far from exhaustive, and there are many other factors that bear on the accuracy of your analysis model. These factors include the modeling of loads and boundary conditions, specification of material properties, and the types and properties of finite elements used in the actual analysis, among many others.

Above all, a design analysis model is best validated through experience - your own correlation between analysis behavior and the actual physical model, combined with the experience that you and your colleagues gain from past models of similar problems. Decisions made at the geometry modeling level are fundamental to this process, and sound assumptions in the design of your geometry model form an important basis for the accuracy of your subsequent analysis.

8.4 The Geometry Menu

To work with geometry in MSC/Ultima, pick the Geometry menu button from the main menu:

This button will bring up a form as shown:



Point at XYZ

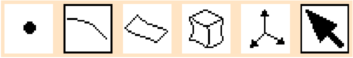
Point Coordinates List

[0 0 0]

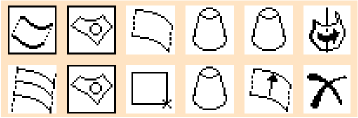
Refer. Coordinate Frame

Coord 0

The top row of icon buttons on this form represent the main kinds of geometry discussed earlier: points, curves, surfaces, solids and coordinate systems.



Once you select one of these buttons, the form will display additional icon buttons below this row, displaying the options available for this geometry type. For example, pressing the Surface button will show a list of icon options as shown:



You can learn the function of each of these icons by holding the mouse pointer over them, which will cause a short text "tool tip" to appear and provide a brief description of the icon's meaning. Selecting one of these option buttons will change the form to show what is needed to perform this operation. For example, in the Curve Fillet option, you are asked to to specify a fillet radius, and then select two curves and the endpoints oriented towards the fillet. These values are then used to generate a fillet curve between the two curves.

As discussed earlier in the chapter Getting Started with MSC/Ultima, you fill in these fields by either entering values in a specific format, or more commonly picking items from the screen. Other fields may involve other ways to specify the value, such as picking from a pull-down list or moving a slider. Once you have specified all of the values needed to complete the operation, simply select the Apply button (or click the right mouse button within the graphics display window to "Apply" the data) to execute the operation.

Note: In MSC/Ultima, you should explicitly pick an item's field before entering or picking data for it. This will help insure that each data item is placed in the proper field on its form.

Many fields in MSC/Ultima menus involve selecting geometry items, which can be done in one of two ways:

1. Picking an appropriate entity from the screen using your mouse. For these fields, a menu of icons will provide different ways of making this pick - for example, a point location can be selected from an existing point, a vertex of a finite element, or even as the intersection of two lines. You can select the appropriate icon, then perform your pick on the screen. A complete list of select menus are described in the chapter Getting to Know MSC/Ultima, and interactive "tool tips" show each icon's function using your mouse pointer.
2. You can also enter a specific value for this field, such as a coordinate location ("1.5 2.0 2.5") or an entity ID ("Point 4"). These values have a specific format that is part of MSC/Ultima's underlying PCL data entry syntax, described in the chapter Getting to Know MSC/Ultima.

Once you are familiar with the basics of using the MSC/Ultima menu, you will find that many of its modeling operations are self-explanatory, because they share a common user interface. Feel free to explore the menu forms and option buttons to get a feel for how these modeling options can be used in your projects.

8.5 Geometric Modeling Capabilities

The following tables describe the capabilities available in MSC/Ultima for creating or manipulating geometry

Points

Capability	Description
XYZ	Creates points from their coordinate locations or from existing vertices in the model.
Intersection	Creates a point at the intersection of two curves or edges.
Project point onto surface normal	Creates a point on a surface by creating a surface normal through an existing point.
Project point onto surface vector	Creates a point on a surface, at the point closest to an existing point.
On curve	Creates a point at a specified parametric location on a curve.
Project point onto curve	Creates a point on a curve, at the location closest to an existing point.

Curves

Capability	Description
Straight Line	Creates a straight curve between two points.
Arc 3 Points	Creates an arc curve that lies in a defined plane and passes through starting, middle and end point locations.

Chapter 3 An Overview of NEC Tracker

This section discusses the main features of NEC Tracker!, and provides an introduction to some of the conventions you will follow in using the program for your own customer site installations. It then covers an overview of the main menus and functions.

At this stage, the most important thing to learn is the basic philosophy behind using NEC Tracker! to manage the database of a PBX installation. There are a few basic things that make this program different from general-purpose software applications, particularly in areas such as data entry, navigation and use of menu functions. Once you become comfortable with these differences, they will add a great deal to your productivity, and help you manage a project with more visibility and fewer errors.

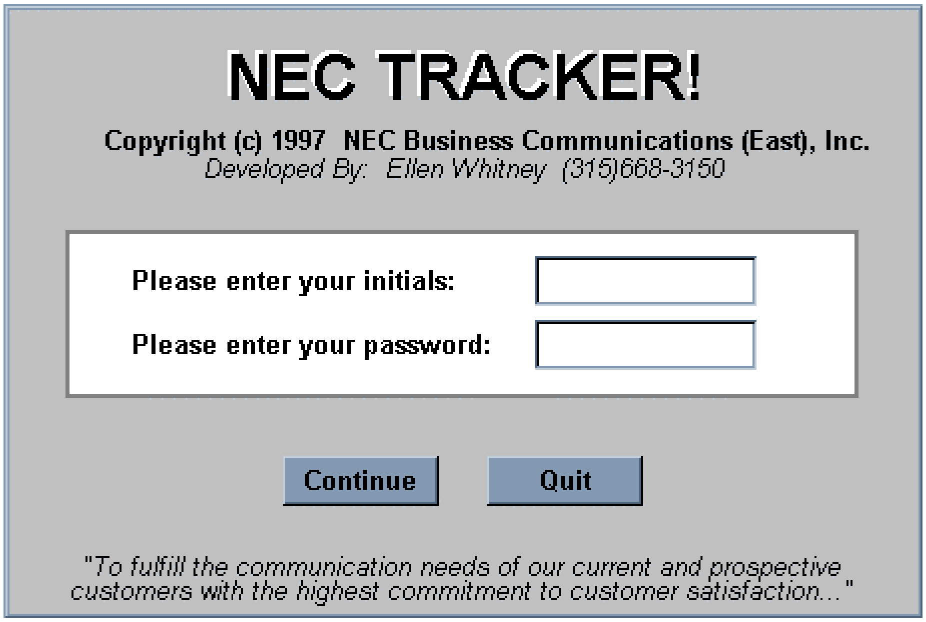
The sample project in Chapter 4 gives you an overview of the process of using NEC Tracker! In later chapters, we will discuss the specific functionality of each part of the program, and provide reference information for specific menus and reports.

Starting the program

NEC Tracker icon

- To start running NEC Tracker!, launch the program by picking the Start button at the lower left hand corner of the screen, and then select *Programs* from its pop-up menu. Select NEC Tracker! from the *Programs* menu, then select the *NEC Tracker!* program icon which appears..

After launching the program, you will see a login screen as shown in Figure 3.1.



NEC TRACKER!

Copyright (c) 1997 NEC Business Communications (East), Inc.
Developed By: Ellen Whitney (315)668-3150

"To fulfill the communication needs of our current and prospective customers with the highest commitment to customer satisfaction..."

Figure 3.1 NEC Tracker! login screen

If this is the first time you are starting the program, you must obtain the startup login initials and password from your software representative. Enter the initials and password in the appropriate fields, and select *Continue*. This should produce a blank screen with a menu that looks like the following:

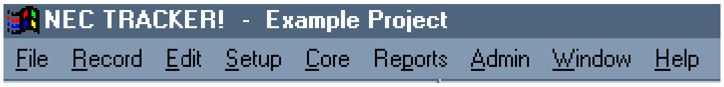


Figure 3.2 NEC Tracker! main menu

Main functions of NEC Tracker!

NEC Tracker! is a custom, application-specific database system designed for installing NEC PBXs. Some of its key features include:

- Data entry of global site information such as customer data, station equipment and outlet location types.
- Entry and management of specific data by extension or location.

- Report generation
- Database maintenance
- Logging and tracking of post-cutover trouble tickets

These features are implemented in an environment that supports the actual process of a PBX installation, including project initiation, customer site and department interviews, and facilities planning. They provide a single data source from which an installation project can be logged, tracked and managed.

General concepts

Managing a project using NEC Tracker! involves three basic steps:

- Project setup
- Data entry and maintenance
- Report generation

Project setup involves entering basic information about the customer and the nature of the installation, including aspects such as site contact information, system type, customer accounting levels, and ranges of extension numbers.

Data entry and maintenance is where the majority of your efforts will be directed. NEC Tracker! differs from many general-purpose database systems, in that it employs a two-stage approach to data entry. The following two-stage approach was designed to reduce data entry errors and to provide valuable tracking information:

1. At the beginning of a project, you provide *lookup table* information. Lookup table information defines the main characteristics of the project: for example, classes of service for this project and the types of station equipment ordered by your customer. These lookup tables cover numerous aspects of your project, ranging from types of extensions to features to be programmed on multi-line telephones.

   By creating these lookup tables, you provide information that will be used during the data entry of individual extensions and spaces, and provide a basis to verify aspects such as equipment needed versus equipment ordered. These tables can be changed as needed throughout the course of the installation.

2. Data is gathered and entered for individual spaces, extensions and cabling. Much (but not all) of this data is based on the information in your lookup tables. For example, if four types of telephones have been ordered by your customer and entered in the equipment lookup table, only these types can be specified for an extension. Moreover, NEC Tracker! can use the total quantities ordered to make sure that individual requirements are within the bounds of the actual equipment order.

These data entry capabilities also allow you to navigate, search and modify existing data. One of the prime reasons for using a database management system during the installation process is the constant change that is part of many installations: departments move, people come and go, and needs change. The same menus and dialog areas you use to enter data also give you the ability to add, delete and search for information while maintaining the current state of the project as a database.

Report generation allows you to communicate key aspects of your project to other project team members, including customer reports, NEC installation reports, administrative reports, and even automated printing of designation labels for NEC digital multi-line telephones.

In this chapter, we will provide a summary of what steps are involved in these three phases. But first, we need to understand a few specific items about how to use the program, including menus, data entry, and navigating through your database.

A quick tour of the menu

The main NEC Tracker! menu is organized as shown in Figure 3.2. These menu items have the following general functions:

File: Primarily used to create a new project, open an existing project, and exit the program.

Record: Record-based navigation and management functions, such as creating, closing, or browsing records.

Edit: Standard Windows editing functions, including cut, paste and copy.

Setup: Entry of project setup and lookup table data.

Core: Primary installation data entry, including voice networking extension, space and cabling data. This menu also contains forms for logging and tracking of post-cutover trouble tickets.

Reports: Controls report generation.

Admin: General database maintenance such as backup and restore.

Window: Standard management of window positioning.

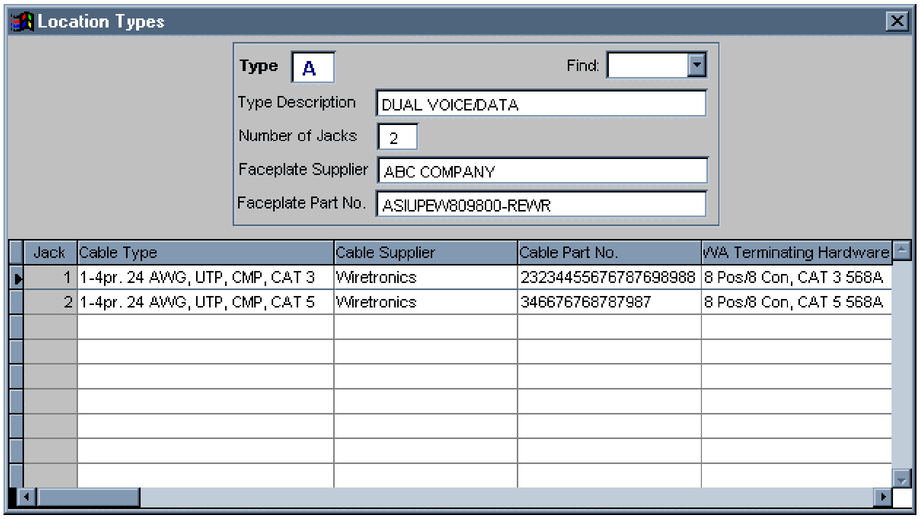
Help: Contains product acknowledgment information. Formal on-line help for NEC Tracker! is currently provided directly in specific functions, using help buttons, tool tip labels or informational screens.

TIP: As a convenience, there is often more than one way to access the same function, so feel free to explore menus and tools. For example, the functions on the Record menu can also be accessed by using key strokes or toolbar buttons, described later.

Entering data in NEC Tracker!

NEC Tracker! uses visual cues to let you know how the data should be entered into specific fields: for example, whether you can enter a value directly, or whether it is selected from a list. Since the ways that you enter data are specific to this application, it is important to understand these conventions before using the program.

Many of the menu functions will produce *dialog boxes* used to enter particular kinds of information. Figure 3.3, for example, shows what the dialog box for location type information from menu choice *Setup --> Location Types* looks like.



Type A

Find:

Type Description DUAL VOICE/DATA

Number of Jacks 2

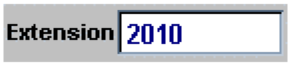
Faceplate Supplier ABC COMPANY

Faceplate Part No. ASIUPEW809800-REWR

Jack	Cable Type	Cable Supplier	Cable Part No.	WA Terminating Hardware
1	1-4pr. 24 AWG, UTP, CMP, CAT 3	Wiretronics	23234455676787698988	8 Pos/8 Con, CAT 3 568A
2	1-4pr. 24 AWG, UTP, CMP, CAT 5	Wiretronics	346676768787987	8 Pos/8 Con, CAT 5 568A

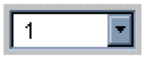
Figure 3.3 Location types dialog box

These dialog boxes will have a number of ways to input data. The main types of data entry that you will use include:



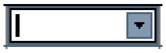
Extension 2010

Input fields. In general, you can directly enter a value within a field by clicking on it, then entering the value. Standard Windows editing conventions apply in these fields, including using the Backspace key to delete characters behind the cursor, the Delete key to delete characters in front of the cursor, and highlighting values by clicking and dragging to replace them.

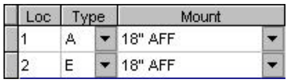


1

Pull-down list box fields. With these fields, you will select an item from a list of predefined values, by selecting the arrow symbol and then picking the appropriate value. Only values currently on the list can be entered in these fields. You can clear a previously defined value by selecting "None".



Combination pull-down list box fields. These fields allow you to either select from a list, or enter a value directly. You can recognize these list boxes by the space surrounding the pull-down arrow. You can clear a previously defined value by simply deleting it using standard Windows editing conventions. Note that you can only type or edit a value in this field when the list is *not* pulled down.



Loc	Type	Mount
1	A	18" AFF
2	E	18" AFF

Grids. Grids contain an array of data. A grid may contain one record or many records. For example, you may have two users assigned to the same extension, and you would like both users' names to appear in the directory. NEC Tracker! uses a grid to display and input "one-to-many" database relationships.

Each row in a grid represents a record. To add a new record, simply click within the next available blank row. Records may also be added above the current selected record by pressing Ctl-I. To delete a record, highlight it and then press Ctl-D. To edit a record within a grid, simply click on the field(s).

For detailed information on how to enter data for a specific dialog box, you can consult the detailed descriptions in later chapters of this document; however, the conventions above serve as a good general guide of how to input data at specific locations.

TIP: In general, you will often be entering data values directly during the Setup phase. By comparison, much (but not all) of the Core information that you enter for areas such as specific extensions or spaces will come from lists that you defined while in Setup.

Navigating the database

One of the most important concepts to understand in NEC Tracker! is that most data entry screens are operating upon a single record of a larger file (or table) within a database. Central to this approach is the distinction between records, tables and your database:

- The ***database*** contains all of the information for your project. You will have one database per PBX installation.
- ***Tables*** within the database contain groups of like information. For example, one table may contain the set of all information for extensions. Many tables in NEC Tracker! are related to other tables, so one table may contain information depending upon values in other tables.
- ***Records*** contain information on an individual item within a table, such as the information needed for installing a single extension. These records are composed of *fields* containing individual data items.

This approach means that most of your work will be centered around individual records. This differs from applications such as word processors, where you may open a file, and then see and change anything you want within the same screen. In NEC Tracker!, as with many database applications, you will generally have one database active at all times, and will open, update and close tables and individual records for different types of information.

You could correctly imagine the program as a viewer that can look at one record of your database at a time -- and only while you are looking, you can perform functions like change the record, close the record, or create a new record.

Most data entry functions in NEC Tracker! will produce a toolbar of functions. This toolbar, which is normally docked in the upper left-hand corner of your screen, can also be moved around the screen as needed by clicking and dragging. It looks like this:

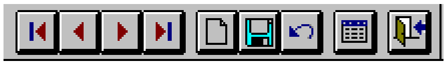


Figure 3.4 Toolbar

This toolbar helps you navigate through the records of the current table, as well as control updates for the records on which you are working. (Equivalents to these buttons are also available from *Record* on the main menu bar.) The functions of this toolbar, and their equivalent keyboard commands, are as follows:

First. Save the current record and go to the first record. (*Ctrl-Home*)

Prior. Save the current record and go to the prior record. (*Ctrl-PgUp*)

Next. Save the current record and go to the next record (*Ctrl-PgDn*)

Last. Save the current record and go to the last record. (*Ctrl-End*)

New. Save the current record and create a new record. (*Ctrl-N*)

Save. Save the current record without exiting. (*Ctrl-S*)

Restore. Restore the current record to its last saved contents. (*Ctrl-E*)

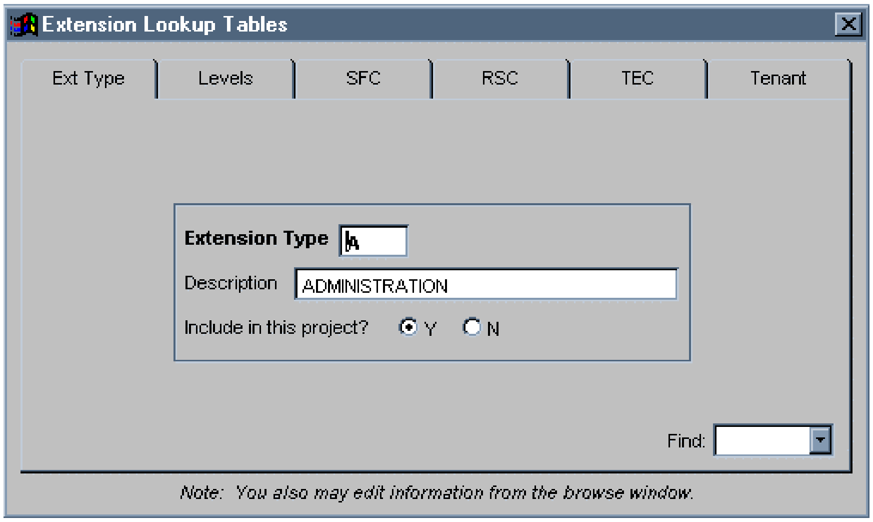
Browse. This will generally bring up a screen which lists all of the records in the current table, in tabular format. (*Ctrl-B*)

Close. Save the current record and exit the dialog box. (*Ctrl-F4*)

To delete a record, enter *Ctrl+Delete* (press the Ctrl and Delete keys simultaneously). To prevent accidental deletion, this function is not included on the toolbar.

TIP: To get a description of the function of a toolbar button, simply move your mouse pointer over it and a "Tool Tip" message will appear. These Tool Tips also display the equivalent keystrokes you can enter to perform these functions.

Let's look at an example where we are entering the types of extensions during the setup of a project, using the dialog box brought up by the menu choice *Setup --> Look Up Tables --> Extension Information*:



Extension Lookup Tables

Extension Type: A

Description: ADMINISTRATION

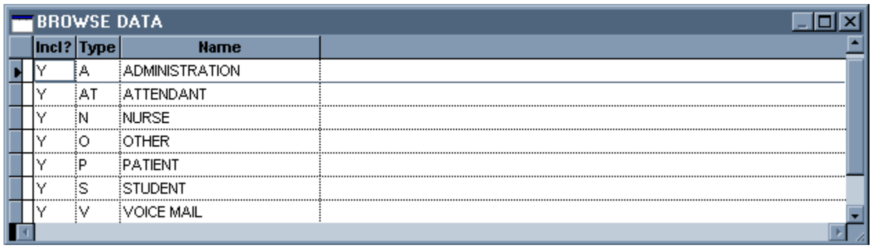
Include in this project? ● Y ○ N

Find:

Note: You also may edit information from the browse window.

Figure 3.5 Extension lookup tables

First, let's take a look at what records are already in the table of extension types. By clicking the *Browse* button, we see a list as follows:



BROWSE DATA

Incl?	Type	Name
Y	A	ADMINISTRATION
Y	AT	ATTENDANT
Y	N	NURSE
Y	O	OTHER
Y	P	PATIENT
Y	S	STUDENT
Y	V	VOICE MAIL

Figure 3.6 Extension types browse data

The type of extension we want for faculty extensions is not on this list, so we will enter a new one. Before we do this, close the Browse window (click the "X" in the upper right hand corner in Windows 95, or click the dash in the upper left hand corner followed by *Close* in Windows 3.1).

Create a new record by clicking the *New* button on the toolbar, which will create the new record and clear the screen.

AIMWorX™

APPLICATIONS INTEGRATION MANAGER

by NEC

User Training Course
Version 2.0

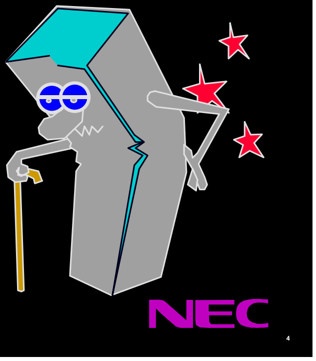
Section I. Introducing AIMWorX

Introducing AIMWorX

- An integrated telemanagement system
- True single point of entry (SPE) interface with synchronized database
- Combines:
  - Call accounting
  - Voice mail
  - Switch programming
  - Message center
  - Other current and future capabilities

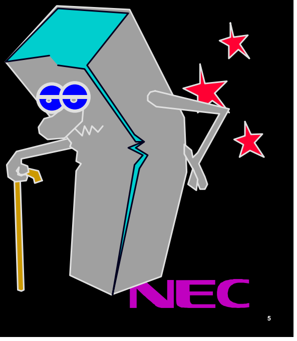
History of the Telemanagement System

- Long Distance Tool
  - Internal Long distance use/abuse
  - Cost Allocation
  - Toll Fraud
- Resource Management
  - Cable Management
  - Asset Management
  - Trouble Tickets
  - Work Orders



History of the Telemanagement System

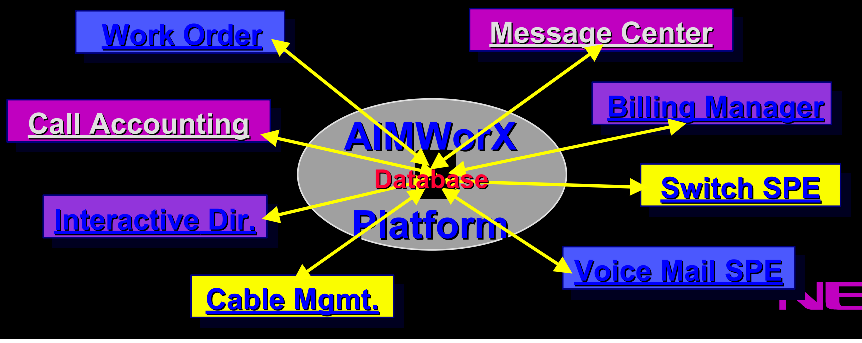
- Fragmented Development and Support
  - Small companies
  - Sold direct and via distribution
- Diverse, Proprietary Products
  - Telephone database
  - Voice Mail database
  - Work Order and Trouble Ticket
  - Directory database
  - Facilities database
  - Authorization/Account Codes



Today, there is AIMWorX:

A single interface to multiple databases

- Single point of entry (SPE) add-in modules
- True data synchronization between call accounting, switch and voice mail
- Modular capabilities updated via security key



AIMWorX Database Platform

Work Order

Message Center

Call Accounting

Billing Manager

Switch SPE

Interactive Dir.

Voice Mail SPE

Cable Mgmt.

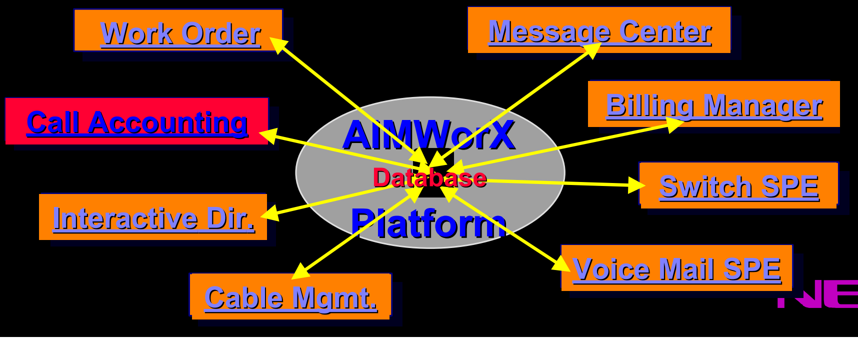
Integrated data management

- Import all data from switch or voice mail
- Import from spread sheets or delimited files
- Specific fields are easily labeled
- Export for use in spreadsheets or other apps
- Prints to HTML, ASCII and other formats
- Export “teaches” data to PBX or voice mail

Call Accounting

Call Accounting is the *immediate* core Benefit

- Multi Site and Multi Level (up to 5)
- From small to very large (up to 65,000 lines)
- Pre-defined and customized reports
- Queries allow one to study database in-depth
- Detailed queries provide precise information



Work Order

Message Center

Call Accounting

AIMWorX

Billing Manager

Database

Switch SPE

Interactive Dir.

Platform

Voice Mail SPE

Cable Mgmt.

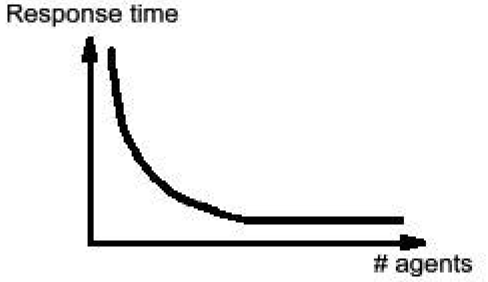
The Mathematics of a Support Center

By Rich Gallagher

One of my dirty little secrets, at least among fellow help desk professionals, is that I spent much of my career as an engineer who specialized in computer graphics. (Ever see the rotating "DC" logo at the end of a Dick Clark Productions television show? That's me!) My first (and still best selling) book isn't on help desks, but a knee-deep-in-equations textbook on 3D visualization techniques. And I'm probably the only person who goes to movies like *Toy Story* to thrill to things like good surface tesselation.

So, even after many years in the customer support profession, I must confess to a certain fascination with the numbers behind a support operation. If you stop to think about it, there is a logic and structure behind something as seemingly chaotic as helping human beings solve their computer problems. Moreover, understanding this structure can help make your life easier in a support center. Here are some of the ways that customer support is a science as well as an art:

Tracking how well you respond. In their excellent book *The Art of Software Support*, authors Françoise Tourniaire and Richard Farrell describe a technique known as an Erlang C analysis. It basically takes an equation involving factors such as the number of agents, average call length and call volume, and then fits the results to a curve telling you how responsive you are to inbound calls. I will spare you the details here (get their book!), but perhaps its most important aspect is an exponential curve of response time versus resources, that looks something like this:



See that steep part of the curve? It basically means that adding even one person can make a dramatic impact in responsiveness when you are understaffed, but adding the Mongol Hordes may not help much as your staffing improves. More importantly, it means that several small support teams – each with their own steep exponential curves – are less responsive on paper than one big one. Therefore, some judicious cross-training could have a big impact on what kind of customer volume your staff can handle – with the aid of a little math.

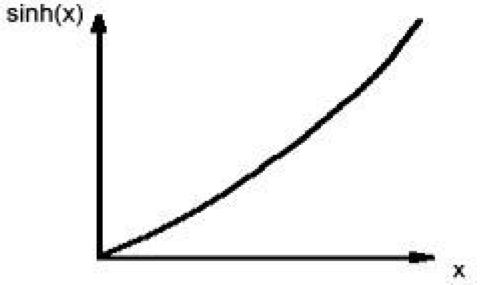
Managing the flow of inbound calls. My employer, The CBORD Group, has copious historical records on when calls come in to our support center. You would expect them to have a similar distribution from one day to the next – but how similar is truly amazing. I could literally blindfold myself, reach into my desk, select records from any random month, and find that the peak call volume is almost exactly 1.6 times the average call volume, occurs between 10 and 11 AM – and varies by less than three or four percent!

More importantly, this distribution is invariant with factors like call volume or time of year. The most languid summer vacation week, or the busiest back-to-work one, will both peak out at 1.6 times the average volume at the same time of day. This means that by the time the first couple of hours at work pass, I can tell you exactly what kind of day it's going to be – probably within a couple of decimal places.

This kind of mathematical consistency translates to real dollars in our pocket. We have live staffing from 7 AM to 7 PM, with call intake and CRM tracking performed by a real person. So this requires a person and a half, right? Wrong. This predictable distribution of calls lets us maintain just one shift of coverage, and then cross-train the company's front desk to handle the relative nano-percentage of calls that come in outside those hours.

Understanding horde behavior. We recently went through a process of making a large customer base Y2K-compliant, by personally contacting them, upgrading their systems, and having them sign a fax-back form certifying their compliance.

Each week, we'd call hundreds of customers. The first few weeks of this, in the early fall, would result in painfully tiny increases in customer compliance. Later weeks provided slightly better results. Then finally, as we moved through November and December, the floodgates opened and everyone finally upgraded. Is this an example of obstinate human behavior? No, it's actually more like an example of a classic hyperbolic sine function.



Knowing how a randomly distributed group of human beings are likely to behave when a deadline approaches can help you stage the appropriate level of resources to meet the demand. More importantly, it provides a model for similar situations in the future (not that I ever want to see another millenium dawn again). And just like pollsters can predict election results from a few percent of the votes, we can perhaps start to understand group behavior better from its early returns.

The bottom line is that technical support is increasingly becoming a numbers-driven profession. And while the wrong emphasis on the wrong numbers can do great harm to your service reputation, there is great joy and perhaps substantial revenue hiding in many of these figures. Nearly any aspect of this business, from the staffing and turnover to the life cycle of a technical problem, has an underlying mathematical framework that can be understood and then exploited. So next time you run your regular support metrics, tip your hat to us engineers – we have more to do with great customer service than you might think.

Rich Gallagher is Client Support Manager for The CBORD Group in Ithaca, NY, and author of several books and training programs on customer support and service. Visit Rich on-line at www.rsgallagher.com.