



## What is Enforcer®?

Enforcer is an MDL (MicroStation Development Language) application, written by Australian Data Systems as a drafting aid to be used across the board in all disciplines of the industry: electrical, architectural, mechanical, structural and cartographic. Enforcer is certified for MicroStation 95, SE, J and V8, and PowerDraft. It is compatible with all popular operating systems: Intel Windows, DOS, and several flavours of Unix. The developers recommend that it be run as a Design Application that is loaded automatically every time that a new drawing is opened.



To realise the full potential of Enforcer, it is necessary to understand the reason for its creation and the process of its evolution:

Since the inception of Computer Aided Design, the drafting industry has recognised the need for and the advantages of further automating the process of creating and maintaining drawings. As contracting and subcontracting became increasingly popular, maintaining consistent drafting standards became crucial. With the emergence of a variety of design-related software packages and the subsequent sharing of data amongst these and different organisations, standards and quality assurance assumed an even higher priority. (Semi) automation was achieved through the creation of User Commands, Basic Macros, MDL's and Bar Menus.

Enforcer takes this automation to the next level and in so doing, has turned the draftsman into a true design professional:

The fundamental principle of Enforcer is feature-driven drafting where a feature is regarded as a real world object that is drawn at a reduced scale. Drawings no longer consist of a series of lines, shapes and curves on different levels with different colours, line styles and weights – they are composed of a collection of intelligent features. A feature consists of a number of predefined properties: its name or description (road, wall, air conditioning unit), its element type (line, text, cell), its symbology (colour, level, weight, line style) and an associated placement command, usually derived from the element type. The placement command is not restricted to the standard MicroStation commands such as Place Line and Place Text – it can be a key-in or an MDL, a Basic Macro or a User Command. The core of Enforcer is a database of Control Files and Command Files that together define the properties of features. Features are presented to the operator in the form of menus and

submenus that are built dynamically when Enforcer is instructed to read and interpret a particular Control File.

From the above it can be seen that Enforcer evolved through the combination and enhancement of previous automation utilities and existing MicroStation components. The potential of Enforcer was further extended through the inclusion of many new and unique functions, several of which were written and refined to satisfy real clients' needs, resulting in the creation of a powerful toolbox for the professional designer. Every function added to Enforcer was developed with a single-minded purpose: to help the drafting industry become as efficient as possible by saving time and money, preventing errors and consequently creating reliable data sets of a consistently high standard and integrity.

## ***Drawing with Enforcer***

In its simplest form when drawing with Enforcer, the operator selects an item from a menu or sub menu without having to worry about how that object is defined, as Enforcer sets all the appropriate symbology and activates the relevant placement command. However, Enforcer incorporates more sophisticated functions: the selection of Smart Tools in the Drawing Toolbox:



One of the most impressive and powerful Smart Tools is the pencil or Draw tool that allows the operator to visually select a feature to draw. When the operator clicks on the selected feature, Enforcer identifies it, sets the appropriate symbology and activates the correct placement command associated with that feature. This is where the inherent intelligence associated with a feature really comes into its own.

The next function in the Toolbox, the Reveal tool, is used to point to an existing item in the design file and to identify it. Enforcer will search for a feature definition in the Control File that matches the symbology of the item. If one is found, a symbology window will display detailing the name and attributes of the feature. If no appropriate feature definition is found, the symbology window will still display with the attributes of the element, but with the description "Not in Control File".

The Fix tool (represented by a spanner) is very useful as well. It changes the symbology of an item (whether valid or invalid) to that of a valid feature. Use the Reveal tool to identify an item (valid or invalid) to change. Then either select a feature from a menu or use the Match tool to graphically select a valid feature, hit the spanner to change the symbology of the Revealed item to that of the Matched one. Enforcer will not allow you to create an invalid object.

The Match tool is the next icon in the Toolbox. It works in conjunction with the Fix tool as described above.

The Activate tool (key symbol) is used to activate the placement command associated with the most recently Revealed feature. It is worth noting at this point that the Reveal tool does not activate the appropriate placement command.

Next in the Toolbox is the Change tool, that combines the functionality of a few tools. It changes the symbology of an item to that of the active feature without having to repeat the Reveal – Match - Fix sequence each time.

The two functions Check and NextError are usually used together. The Check tool is used to do a quick analysis of the design file, not as detailed as the Analyse function, and NextError can be used to scroll through, identify and zoom in on each error encountered.

### ***Analysing with Enforcer***

In the very early stages of its evolution, Enforcer was simply a checking tool that analysed the contents of a drawing, compared each feature to its matching definition in the Control File, and reported any exceptions (non-matches) as errors.



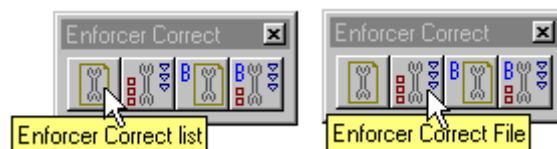
Apart from all the other functions that have been added, the checking function, or Analyse function as it has become known, has advanced well beyond these early beginnings. In addition to checking individual features, special Plus Required statements (so called because of their particular syntax that expects a “+” symbol as the first character) can be used to ensure that the design file conforms to specific settings regarding global origin and working units. Other Plus Required statements can be used to ensure that the correct reference files, cell library and colour table are attached. Then there are Plus Special and Plus Allow statements that are used to cater for exceptional circumstances, such as text features with negative height and width, or forcing the components of cells and not just the cell header to be analysed for conformity as well. Plus Report statements control the format of the analysis report, for example whether to include an element type summary or whether to list a feature summary. The Plus statements are very powerful and cover an extensive range. The few examples mentioned above have by no means exhausted the list.

The Analyse function always reports on elements encountered in the design file which do not have a matching feature definition in the Control File. This list can be produced in detail (one line for each non-matched element in the file) or in summary (one line for each unique error with a counter specifying the number of times it occurred). These error lines have a very special significance. By viewing the analysis report (or log file as it is called) on the screen from within Enforcer and double-clicking on an error line, the faulty element is windowed up on the screen and selected. From here it is a simple matter of either deleting the element or using any of the Enforcer tools (described earlier) to fix the problem.

The Analyse function also incorporates a batch mode, which makes it possible to analyse and report on multiple drawings during the same run. On completion of the process, each design file has an associated log file that contains the details related to that specific drawing, and there is also a batch log file that contains a summary for the entire batch. The batch may consist of a number of files in a particular directory or it can be a list of specific file names entered into a text file.

## ***Correcting and Converting with Enforcer***

During the next stage of Enforcer's development cycle, some very powerful functions were created that implemented the ability to process many elements at once as opposed to fixing errors on an individual basis within a single design file. At that time the concepts of Correction vs Conversion were introduced.



**Correction** is defined as the process of modifying elements within the **same design file**. The rules for correction can be obtained from one of three sources:

1. A correction list which is built manually by the operator.
2. By using two different Control Files (From and To) and changing the element symbology from one design standard to another, where one standard is defined in the From Control File and the new standard is defined in the To Control File.
3. By using the same Control File for both From and To, but changing the Scale Factor so that the relevant features are recreated at a different scale.



**Conversion** is the process of extracting elements from one design file, possibly but not necessarily changing their symbology from one design standard to another and recreating them in **another design file**. The rules for conversion can be obtained from the following sources:

1. By using two different Control Files and converting the elements from one design standard (the From Control File) to another standard (the To Control File), while copying the elements from one design file to another.
2. By using the same Control File for both From and To, but changing the Scale Factor so that the relevant features from the original design file are recreated at a different scale in another design file.

Both Correction and Conversion can be done in Batch mode, which makes it possible to correct or convert many design files during the same run.

The Conversion function in particular has many beneficial side effects. Because it extracts elements from one file and recreates them in another, it is a useful tool for creating subsets of data from design files received from a third party.

Conversion and Correction can be used very efficiently in a contract drafting environment where it may be necessary to supply the same data to different clients in different design standards.

### ***Viewing and Plotting with Enforcer***

Enforcer provides a powerful and extremely useful tool for performing visual sanity checking on drawings. Control Files are perfect for ensuring that features are drawn with the correct symbology, but it still requires the expertise of a skilled technician to ensure that a building has an adequate number of doors and windows and that a major highway hasn't replaced a firebreak.



The viewing filter in Enforcer is similar in operation to MicroStation's Select by Attributes function for elements, except that Enforcer's version is feature-based. The viewing filter applies to plotting and printing as well.

### ***Creating Control Files for and with Enforcer***

As stated previously, the core of Enforcer is a database of Control Files. While the creation and maintenance of Control Files may appear a daunting task to the CAD operator, it is usually the domain of the System Administrator. The in-built functions in Enforcer for creating a Control File automatically from the standards employed in a selected design file, and allowing the user to edit this file via a GUI interface with all the necessary validation and error-checking included, further assist in making this an easy, user-friendly experience.



## ***Electronic Certification of Design Files***

When a design file is error-free and complies one hundred percent with the expected design standard, Enforcer can be instructed to write a digital certification stamp into the design file, detailing who approved it, the date and time, and the names and revision numbers of the Control Files used. If even the most insignificant modification is subsequently made to the design file, Enforcer will detect that fact and warn the operator that the certification is no longer valid.

## ***The Command File***

One of the special Control Files in Enforcer's armoury is the Command File. This file is normally used only when it is necessary to override the default placement command associated with a particular element type. A typical example of a situation where an override command comes into its own is with element type 6. The default associated placement command is *place shape*. However, for a particular application it may be more logical (and more efficient) if the default command could be *place block*. This is achieved by defining *place block* as the override command for that feature in the Command File.

But defining override placement commands is not the only advantage of the Command File. It is used extremely effectively to define utility commands, for example MDL's that require key-ins to operate. The number of instances where this technique can be put to good use is unlimited, especially when Enforcer is loaded as a seamlessly integrated design application.

## ***What is Enforcer?***

And so we return to the original question. In short, Enforcer is whatever you want it to be, especially when set up as a design application which is always there and available when you need it. With a look and feel similar to standard MicroStation, it is often difficult to distinguish where one starts and the other ends. Enforcer is a toolbox of utilities that can be applied in a variety of ways limited only by your imagination.



### **Australian Data Systems**

55 Morrison Road, Midland, Western Australia 6056

Ph: +61 8 9250 8044 Fax: +61 8 9250 8344

[ads@ozdata.com.au](mailto:ads@ozdata.com.au)

[www.ozdata.com.au](http://www.ozdata.com.au)