Natural Engineer

Version 4.4.2

Concepts and Facilities

Manual Order Number: NEE442-006ALL

This document applies to Natural Engineer version 4.4.2 and to all subsequent releases.

Specifications contained herein are subject to change, and these changes will be reported in subsequent revisions or editions.

Readers' comments are welcomed. Comments may be addressed to the Documentation Department at the address on the back cover. Internet users may send comments to the following e-mail address:

document@gensystems.com

Acknowledgements

TABLE OF CONTENTS

ABOUT THIS MANUAL

Purpose of this manual

This manual contains the Concepts and Facilities for Natural Engineer.

It describes some of the concepts and principles of operation in using Natural Engineer.

Target Audience

The target audience for this manual is intended to be any User of Natural Engineer at any level of experience.

Typographical Conventions used in this manual

The following conventions are used throughout this manual:

UPPERCASE TIMES	Commands, statements, names of programs and utilities referred to in text paragraphs appear in normal (Times) uppercase.
UPPERCASE BOLD COURIER	In illustrations or examples of commands, items in uppercase bold courier must be typed in as they appear.
< >	Items in angled brackets are placeholders for user-supplied information. For example, if asked to enter <file number>, you must type the number of the required file.
Underlined	Underlined parts of text are hyperlinks to other parts within the online source manual. This manual was written in MS-Word 97 using the "hyperlink" feature.

The following symbols are used for instructions:

⇒	Marks the beginning of an instruction set.
❑	Indicates that the instruction set consists of a single step.
1.	Indicates the first of a number of steps.

How this manual is organized

This manual is organized to reflect all the Concepts & Facilities of Natural Engineer in the following chapters:

Chapter	Contents
1	Provides an overview of the main Natural Engineer processes.
2	Describes some of the common facilities available within Natural Engineer.

Terminology

It is assumed that you are familiar with general Natural and mainframe terminology, as well as the terms and concepts relating to Microsoft Windows operating systems. This section explains some terms that are specific to the Natural Engineer product.

Analysis

The Analysis process of Natural Engineer searches application data within the Natural Engineer Repository, according to specified Search Criteria and generates reports on the search results.

Application

An Application is a library or group of related libraries, which define a complete Application. In Natural Engineer, the Application can have a one-to-one relationship with a single library of the same name, or a library of a different name, as well as related steplibs. The Application refers to all the source code from these libraries, which Natural Engineer loads into the Repository.

Browser

An Internet Browser such as Microsoft Internet Explorer or Netscape.

Category

Categories in Natural Engineer specify whether and how a Modification is applied to the Natural code. Valid categories are: Automatic change, Manual change, Reject the default Modification, No change to the data item, and the data item is in Generated Code.

A category is further broken down according to type of change (for example: Keyword, Literal, Data Item, Database Access, Definition).

Consistency

An option in the Analysis process that causes Natural Engineer to trace an Impact through the code, using left and right argument resolution to identify further code impacted by the code found.

Environment

The Environment process is the means by which Natural Engineer generates a structured view of the application code in the Natural Engineer Repository. This provides application analysis reports and inventory information on the application and is used as the basis for Impact Analysis.

Exception

An Exception is an Item identified as impacted that does not require a Modification. Where there are a few similar Exception Items, they can be treated as Exceptions, and rejected in the Modification review process. Where there are many similar (therefore not Exceptions), consideration should be given to changing the Search Criteria so they are not identified as impacted in the first place.

Generated Code

This is code which has been generated by a Natural code generator, such as Construct, and which is not normally modified directly in the Natural editor.

Impact

An Impact is an instance of a Natural code Item; e.g., data item or statement (a “hit” scored by the Analysis process) that matches the defined Search Criteria used in the Analysis process.

Iteration

An Iteration is one examination cycle of a field identified according to the specified Search Criteria. For example, one Iteration is reading the field right to left. Multiple Iterations are performed when the option of ‘Consistency’ or Multi Search is requested for Analysis, and Natural Engineer performs as many Iterations as necessary to exhaust all possibilities of expressing and tracing the field, and can be limited by a setting in the NATENG.INI file.

Library

A single library of source code, which exists in the Natural system file.

Modification

A Modification is a change suggested or made to an object or data item resulting in the required compliance of that object or data item. Modifications in Natural Engineer are classified according to Category and Type.

Presentation Split Process

The Presentation Split Process is a sub-function of the Object Builder function that removes screen I/O statements from current application objects and places them in generated subprograms.

Soft Link

A Soft Link is where a link between two objects has been defined using an alphanumeric variable rather than a literal constant.

Technical Split Process

The Technical Split Process is a sub-function of the Object Builder function that results in the encapsulation of each database access within the application, into a sub-program so that the application is separated into 'presentation and logic' and 'database access'.

Type

The Type of Modification available, for example: Data Item, Keyword and Literal.

TLM

Text Logic Members are used to contain the code required to support inclusion of common code into the application. An example of this is the code to include into an application before updating a database.

Related Literature

The complete set of Natural Engineer manuals consists of:

1 **Natural Engineer Concepts and Facilities (NEE442-006ALL)**

The Concepts and Facilities manual describes the many application systems problems and solutions offered by Natural Engineer, providing some guidelines and usage that can be applied to Natural applications.

2 **Natural Engineer Release Notes (NEE442-008ALL)**

The Release Notes describe all the information relating to the new features, upgrades to existing functions and documentation updates that have been applied to Natural Engineer.

3 **Natural Engineer Installation Guide (NEE442-010ALL)**

The Installation Guide provides information on how to install Natural Engineer on both PC and mainframe platforms.

4 **Natural Engineer Administration Guide (NEE442-040WIN)**
Natural Engineer Administration Guide (NEE442-040MFR)

The Administration Guide provides information on all the various control settings available to control the usage of the different functions within Natural Engineer.

5 **Natural Engineer Application Management (NEE442-020WIN)**
Natural Engineer Application Management (NEE442-020MFR)

The Application Management manual describes all the functions required to add Natural applications into the Repository.

6 **Natural Engineer Application Documentation (NEE442-022WIN)**
Natural Engineer Application Documentation (NEE442-022MFR)

The Application Documentation manual describes all the available functions to document a Natural application within the Repository. These functions will help enhance / supplement any existing systems documentation such as BSD / CSD / Specifications etc.

7 **Natural Engineer Application Analysis and Modification (NEE442-023WIN)**
Natural Engineer Application Analysis and Modification (NEE442-023MFR)

The Application Analysis and Modification manual describes all the available functions to carry out analysis of Natural applications; including basic keyword searches. The modification process is described and detailed to show how it can be applied to modify single selected objects within a Natural application, or the entire Natural application in one single execution.

8 Natural Engineer Application Restructuring (NEE442-024WIN)
Natural Engineer Application Restructuring (NEE442-024MFR)

The Application Restructuring manual describes the analysis and modification functionality required to carryout some of the more sophisticated functions such as Object Builder.

9 Natural Engineer Utilities (NEE442-080WIN)
Natural Engineer Utilities (NEE442-080MFR)

The Utilities manual describes all the available utilities found within Natural Engineer and, when and how they should be used.

10 Natural Engineer Reporting (NEE442-025ALL)

The Reporting manual describes each of the reports available in detail, providing report layouts, how to trigger the report and when the report data becomes available. The various report-producing mediums within Natural Engineer are also described.

11 Natural Engineer Batch Processing [Mainframes] (NEE442-026MFR)

The Batch Processing manual describes the various batch jobs (JCL) and their functionality.

12 Natural Engineer WebStar (NWS442-020ALL)

The WebStar manual describes the concepts and facilities, installation and configuration options, how to web enable a Natural application and how to create and execute Natural Short Transactions using the Natural Engineer add-on component WebStar.

13 Natural Engineer WebStar Release Notes (NWS442-008ALL)

The Release Notes describe all the information relating to the new features, upgrades to existing functions and documentation updates that have been applied to the Natural Engineer add-on component WebStar.

14 Natural Engineer Messages and Codes (NEE442-060ALL)

The Messages and Codes manual describes the various messages and codes produced by Natural Engineer.

1

HOW NATURAL ENGINEER WORKS

Chapter Overview

This chapter provides an overview of the processes that provide Natural Engineer functionality.

The topics covered are:

1. The Natural Engineer Package
2. Environment Process
3. Analysis Process
4. Modification Process
5. Utilities
6. Error Handling
7. Consistency Analysis

The Natural Engineer Package

Natural Engineer consists of a number of processes, which, when executed sequentially, perform the Environment, Analysis and Modification functions.

- PC versions of Natural Engineer provide a PC-based Repository and a GUI front-end that allow you to specify the various required parameters and triggers for the individual processes.
- Mainframe versions of Natural Engineer provide a mainframe based Repository and mainframe screens that allow you to specify the various required parameters and triggers for the individual processes.

Inventory

All Natural Engineer processing options start with the creation of the Natural Engineer Repository. By this process of extracting and loading application information into the database, Natural Engineer provides a comprehensive inventory of statistics, metrics, structures, cross-reference, high-level and detail-level data. In addition, Natural Engineer gives you the ability to review this inventory, interactively, or through reporting.

The creation of an Inventory requires the following processes:

- Extract Source Code
- Load extracted Source Code into the Repository

Natural Construct

Natural Engineer ‘understands’ and processes Natural Construct code.

Construct Models and User Exits are identified and lines of code are marked as either generated or User Exit code.

During each Natural Engineer process, Construct-specific reports are provided to show the effect of the process on the Construct Model, as well as to provide referencing information.

During the Modification phase, only those lines of code that are in User Exit code are modified. Generated Code is not modified.

If a customer has modified their Construct-generated object by removing the **SAG lines at the top of the program (thus making the object non re-generatable), Natural Engineer will treat the object as a normal non-generated object.

Predict Case

Natural Engineer ‘understands’ and processes Predict Case code.

Predict Case Components are identified and lines of code are marked depending on which Component they are part of.

An impact report is available which identifies which impacted lines of code are related to which Predict Case Component.

Steplib Processing

Natural Engineer can be used with applications that make use of steplib libraries during run time, where the steplib library may contain additional / common processes used by the application.

Natural Engineer needs to know the relationship between the application and the referenced steplib library, to maintain integrity during the Load, Impact and Modification processes.

In order for this relationship to be recognized, both the steplib library and the application library need to be extracted and loaded into the Repository. The steplib library needs to be loaded first, in order for the correct cross-reference information to be generated for the application.

Up to eight steplib libraries can be catered for per application, with each individual steplib library going through the Extract and Load processes before the application.

The following example illustrates a simple steplib application process, based on the application library NATLIB1, which utilizes a steplib library NATSTEP1.

1. Define an application for the steplib library NATSTEP1.
2. Execute Extract Source Code and review the extract by checking the Quality Logs: Extract Source Code Summary and Missing Natural Objects.
3. Load the application NATSTEP1 into the Repository.
4. Define an application for the application library NATLIB1.
5. Specify the steplib relationship for application NATLIB1, by adding the steplib library name NATSTEP1 to the steplibs list using Application Preferences.
6. Execute Extract Source Code and review the extract by checking the Quality Logs: Extract Source Code Summary and Missing Natural Objects.

7. Load the application NATLIB1 into the Repository. This will generate the cross-reference records for each object being referenced from the steplib library NATSTEP1, by objects within the application library NATLIB1. This data will be held as part of the application NATLIB1.

   There will also be cross-reference records created for each object on the steplib library NATSTEP1, which are referenced by objects in application library NATLIB1. This data will be held as part of the steplib application NATSTEP1.

   This provides a complete picture of the relationship between the application library NATLIB1 and steplib library NATSTEP1.

Environment Process

This process creates the Natural Engineer Repository with the source code of the application to be analyzed. The scope of this process ensures that the most complete Repository information is generated on which to perform Analysis.

- The source code is extracted from the application for each statement, checked, neutralized, and loaded into the Natural Engineer Repository. This allows Natural Engineer to handle all versions of Natural application code.
- From the cross-referenced Repository created, Natural Engineer provides comprehensive application Analysis, with reports at summary, object and detail levels. These application Analysis reports provide inventory information about all the application code in the Repository. These reports can also be represented graphically using the interface to an OLE-compliant diagramming tool.
- The Environment process also performs a quality check on the code, reporting any missing objects or incomplete syntax.

In order to ensure that DDM field formats are available to Natural Engineer, it is recommended that you include all DDMs referenced by the application. After extraction of the source code, the Missing Natural Objects report also reports on missing DDMs. Before you continue, you must ensure that all relevant DDMs are extracted. Any objects that reference those DDMs must be re-extracted to include the DDM information.

Fundamental Steps

1. Create an application name and specify the application preferences to be applied by identifying the Natural library from which the source code is to be extracted.
2. Define Extract Selection Criteria to extract either the complete library or selected objects.
3. Execute Extract, which will create an output file.
4. Load the Repository with the extracted code, which will load the file created by the Extract process.
5. Review the Extract Quality Logs to identify and correct errors.

   Note: For more information on the errors, refer to the section Error Handling.

6. Where errors have been identified, for errors prefixed with APP, identify the location of any missing objects and copy them to the application library, or a defined Natural Engineer steplib. These will be missing DDMs, Data Areas, or COPYCODE. Missing DDMs on the mainframe, mean they do not exist in Predict. Missing DDMs on the PC, mean they do not exist in the application library, a steplib library or the System library.

   Investigate source code syntax errors identified for correction or removal of the object.
7. Where missing objects are required for Natural Engineer processing, either selectively Extract and Load the relevant missing objects identified from the Extract Quality Log (that is, execute steps 2 to 4 again for these objects), or execute Extract Missing Object once the objects have been copied.
8. Repeat the above steps until all required objects are no longer identified as errors on the Extract Quality Log.
9. Review the Load Quality Log for any errors that occurred during loading the Natural Engineer Repository. Errors in this log can also relate to the database session parameters on which the Natural Engineer Repository file resides.

   Note: For more information on the errors, refer to the section Error Handling).

10. Review the Application Reports as required you can choose between summary, object and detail reports.

Analysis Process

The Analysis process provides the ability to define a set of Search Criteria for scanning and identifying impacts in the Application, for statements, keywords, fields and text, in order to find code that may need to be changed for maintenance, standards, migrations or componentization.

Known Search Criteria may be provided, for example the Analysis and changes required for migrating versions of Natural. Where Consistency is specified for Analysis, for each field identified as a result of the process, Natural Engineer performs unlimited Iterations (or limited to the number specified in the NATENG.INI/CINI settings), using left and right argument checking until all possible combinations are exhausted.

The Impact Analysis provides the information to ascertain the size of the impact for the application, as well as the level of automatic Modification available. Reports of the Impact Analysis are generated at summary, object and detail levels, and impacted source code is highlighted.

Note: For more information on NATENG.INI settings refer to Chapter 1 in the Natural Engineer Administration Guide for Windows manual.

Note: For more information on CINI settings refer to Chapter 1 in the Natural Engineer Administration Guide for Mainframes manual

Fundamental Steps

1. Define the Impact Search Criteria and any applicable Modification strategy, and specify whether a Consistency check is required. The Impact Search Criteria can be versioned so that multiple sets of criteria are available per application loaded into the Repository.
2. Execute Impact Analysis.
3. Review the Data Item Impact Inventory report.
4. In order to review the Impacts identified, you can review the code on the Impact Element Maintenance screen, or view the highlighted Impacted Source Code in Browser by selecting View Impacted Source Code.
5. Execute the Impacted External Objects report to determine if any impacted data elements are passed to external routines. The owner of the external routine must be identified to determine what effect the impacted data element has on the external object and how it may need to be changed.
6. Review Impact Reports as required: you can choose between summary, object and detail reports.

Modification Process

The Modification process allows you to confirm and modify the categories of Modification applied by Natural Engineer. Categories of Modification are: Automatic, Manual, Reject, Not Applicable and Generated.

Before executing the Modification, you can view and confirm the code and definition changes to be applied. Reports of Modification changes are generated at the summary, object and detail levels, as well as change audit logs.

Fundamental Steps

Note: After Modification, Natural Engineer places all modified code into the Modification Library as specified in the Preferences for the application. If this is not set, Natural Engineer places all modified code in a library name with an 'X' as the last character of the application name. If the name is already 8 characters long, the last character is removed and replaced with the 'X'.

Warning: Once you have started modifying objects with Natural Engineer, do not re-execute the Impact Analysis until the Modification phase has been completed. Current Modification information within Natural Engineer is over-written if Impact Analysis is re-executed.

1. Review Modification categories and types by using the , Modification Element Maintenance option.
2. Verify and modify the default category and type changes for each object as required. Assign the Reject category to any change not required. Natural Engineer does not check parents of fields when changes are made to the Category and Type, so these must be made manually.
3. Execute automatic modification either by object using the Modification Element Maintenance screen, or for all objects using the Execute Modification for All Objects function on the menu.
4. Execute the Impacted Objects Not Directly Modified report to determine impacted objects that contain no direct Modification lines that must be copied to the library containing the modified code. These are Objects that reference impacted data areas or Copycode.
5. Execute the Predict Changes report and make changes to Predict definitions so that the new DDMs can be generated. Allow access to the new DDMs from the library containing the modified code.
6. Stow the Data Areas.
7. Execute the 'check' command in Natural to determine if any errors exist in the code which may require manual change, such as overlapping fields in maps. This verifies the automatic Natural Engineer Modification.
8. Make any further manual changes required to the application objects identified by Natural Engineer.

9. ‘Stow’ the application code in the modified library.
10. Review Modification Reports for information of interest, including summary, object, and detail reports.
11. Audit information is available by using the Change Management Tracking option from the Utilities menu.

Utilities

Natural Engineer provides several Utility options that complement the core processes:

- **Beautification**
  Applies various enhancements to the layout of the source code for program, subprogram, subroutine and helproutine objects. This helps to improve the readability of source code.

  Note: This option is only available to the Windows version of Natural Engineer.

- **Task Scheduler**
  Provides the facility to schedule unattended tasks for long running Environment, Analysis and Modification batch tasks. Long running tasks can be scheduled to run out of normal working hours to provide less disruption to the normal working day.

- **Compare**
  Provides the facility to compare Natural objects between two or three Natural libraries. The Compare Results show differences between objects using customizable color schemes. Code differences can be copied into the base object using basic editing functions, copy and paste. User changes can be applied using basic editing functions insert, delete and undo delete. Changes can then be saved to a Natural library.

  Note: This option is only available to the Windows version of Natural Engineer.

- **Change Management Tracking**
  Provides Audit Trail information on source code changes applied to objects using Natural Engineer's Modification option. Information is available either in interactive mode or via reports using display medium of Natural Reporter, MS Excel or the Natural screen.

Error Handling

Environment

Natural Engineer may generate two error logs during execution, one when extracting source code and one when loading the Natural Engineer Repository.

The error logs have the following prefixes:

- **APP**
  This error type identifies:
  - missing Natural objects within the application library. They could be DDMs, Data Areas or Copycode. You can resolve these errors by moving the identified object to the application library, a defined steplib library or the SYSTEM library. These can then either be selectively Extracted and Loaded, or you can execute Extract Missing Objects, or re-Extract and Load the application.
  - basic application syntax errors, which are further identified by the error text: SRC ERR. Investigate source code syntax errors and correct, or remove the object.
- **GSL**
  This error type relates to the actual processing of the application code. The error relates to both a Natural Engineer object as well as an object in the user application.

Support

Before you contact the Support Desk on any error log, please take the following actions:

1. Investigate whether the user application object can be ‘checked’ within the Natural environment.
2. If the check fails, then either make appropriate changes to the object, or remove it from the library. No further action is required.
3. If the object ‘checks’ under Natural, then provide a copy of the relevant line of code with the appropriate error log to the Support Desk.

Consistency Analysis

Consistency Analysis is available when the consistency option is switched on when specifying Impact Search Criteria for search keywords DATAITEM, DBFILE and DEFINITION, or when using the search keyword MULTI SEARCH.

Consistency Analysis provides the relationship information for a specified Impact Search Criteria in relation to other data items within an impacted object. It will also analyze across objects to build up a full relationship of the specified criteria across all impacted data items, across all objects within an application.

There are three distinct phases during this type of Analysis:

1. Single Object Impacts
2. Inter Object Relationships (IOR)
3. Propagation Phase

Single Object Impacts

The Impacted Object phase will search for the specified Impact Search Criteria and if a match is found within an object, the Impact is marked as 'specified'. Then the Impact process will analyze the impacted object for any data manipulations of the specified criteria.

If the impacted data item is moved to another data item, then this target data item is impacted as 'derived'. The derivation occurs on the second-n Iterations until all possible derivations have been exhausted. For example:

For the statement: **MOVE #A TO #B**

Using Impact Search Criteria: Search Keyword = **DATAITEM** and Search Value = **#A**, Impact Analysis would Impact **#A** as the specified Impact and **#B** as the derived Impact.

If **#B** was then further manipulated by statement: **MOVE #B TO #C,** then **#C** would be impacted as derived.

Inter Object Relationships (IOR)

Once the single object impacts have been completed the IOR processing is activated which checks for impacts across object boundaries. This part of Consistency Analysis is controlled via the IOR parameter setting in the NATENG.INI file. IOR can be turned off by setting IOR=N.

Given a Search Keyword of DATAITEM and a search value of ?#DATE? and using the following program and subprogram below

PGM1	**SUBPGM1**
DEFINE DATA LOCAL 01 #DATE(N06) END-DEFINE CALLNAT 'SUBPGM1' #DATE END	DEFINE DATA PARAMETER 01 #A(N06) LOCAL 01 #B(N06) END-DEFINE MOVE #A TO #B END

Single Object Impact will only have marked #DATE as being impacted. IOR knows that #DATE is passed to SUBPGM1 and that the receiving field is #A. It checks to see if #A is already impacted, if not it impacts it as a derived impact.

Natural Engineer then has to check all derivations of #A and as such will also mark #B as derived in SUBPGM1. It will perform this IOR checking for all fields in the following statements:

- CALLNAT
- PERFORM (External subroutines only)
- STACK
- FETCH/FETCH RETURN
- RUN/RUN REPEAT
- INPUT USING MAP/FORM
- WRITE USING MAP
- INPUT USING HELP
- OPEN DIALOG

It will also try and match impacts 'from target to parameter' so for example if the receiving field in the subprogram is impacted but the parameter being passed is not, then Natural Engineer will mark the parameter field as being impacted.

Propagation Phase

The Propagation phase comprises six stages:

1. **Propagate Data Areas** where any impacts in code are propagated back to external data areas if they didn't previously exist.
2. **Propagate DDMs** where any impacts found within views within the code are propagated back to external DDMs.
3. **Propagate Copycode** where any impacts in code are propagated back to external Copycodes if they didn't previously exist.
4. **Propagate Definitions** where impacts to locally defined view fields are pushed back to external DDMs.
5. **Global Data Areas** where impacted Global data item(s) have been impacted, each Global data item will then be analyzed using the three phases. This process is optional and is controlled by the GLOBAL_DATAITEM= parameter setting in the NATENG.INI file.
6. **DDMs** where impacted DDM data item(s) have been impacted, each DDM data item will then be analyzed using the three phases. This process is optional and is controlled by the DDM_DATAITEM= parameter setting in the NATENG.INI file.

Special Features when using Consistency Analysis

Mapping of Redefined Fields

If an impacted field is a child of a redefined field, then Natural Engineer assumes that the bytes that the field occupies are impacted and will mark other fields that occupy these bytes as impacted as well.

For example:

```
DEFINE DATA LOCAL
01 #WORK-AREA(A20)
01 REDEFINE #WORK-AREA
    02 #FIELD-1(A10)
    02 #FIELD-2(A06)
    02 #FIELD-3(A04)
01 REDEFINE #WORK-AREA
    02 #FIELD-4(A10)
    02 #DATE-1(A06)
    02 #FIELD-5(A04)
END-DEFINE
WRITE #DATE-1
END
```

Impact Search Criteria: Search Keyword = **DATAITEM**, Search Value = **?DATE?** and consistency switched on.

Natural Engineer will mark **#DATE-1** as a specified impact. **#DATE-1** occupies bytes 11-16 inclusive of the field **#WORK-AREA**. Natural Engineer will then map this impact over all redefinitions of **#WORK-AREA**. The field **#FIELD-2** occupies the same byte range of **#WORK-AREA** so Natural Engineer will mark this field as a derived impact.

This facility may be turned off by the REDEFMAP= parameter setting in the [IMPACT] section of the NATENG.INI file. The default setting is REDEFMAP=N.

Deriving Parents of Redefined Fields

If a field is a child of a redefined field and is impacted Natural Engineer will derive the parent field.

For example:

```
DEFINE DATA LOCAL
01 #WORK-AREA(A16)
01 REDEFINE #WORK-AREA
    02 #FIELD-1(N08)
    02 #FIELD-2(N06)
    02 REDEFINE #FIELD-2
      03 #YY   (N02)
      03 #MM   (N02)
      03 #DD   (N02)
    02 #FIELD-3(N02)
END-DEFINE
WRITE #YY
END
```

Impact Search Criteria: Search Keyword = **DATAITEM**, Search Value = **?YY?** and consistency switched on.

Natural Engineer will mark the field **#YY** as being impacted and then derive the field **#FIELD-2**.

COMPRESS and SEPARATE Statements

If a source field is impacted in a COMPRESS or SEPARATE statement then Natural Engineer will only derive the target field. It will NOT derive all other source fields. However, if the target field is impacted Natural Engineer will derive all source fields.

Source Field impacted

For example, using the following lines of code and a Search Keyword of **DATAITEM** and a Search Value of **?DATE?** and consistency switched on.

```
COMPRESS 'TODAY IS:' #DATE ':' #END INTO #ALPHA LEAVING NO
```

For this COMPRESS statement Natural Engineer will impact **#DATE** as a specified impact and it will then derive the target field **#ALPHA**.

```
SEPARATE #X INTO #5 #DATE1 #6 WITH DELIMITER ':'
```

For this SEPARATE statement Natural Engineer will impact **#DATE1** as a specified impact and it will then derive the target field **#X**.

Target Field impacted

However if a target field is impacted then Natural Engineer will derive all source fields.

For example, using the following lines of code and a Search Keyword of **DATAITEM** and a Search Value of **?DATE?** and with consistency switched on.

```
0330 COMPRESS #7 #8 #9 INTO #DATE-ALPHA
```

For this COMPRESS statement Natural Engineer will impact **#DATE-ALPHA** as a specified impact and it will then derive all source fields **#7**, **#8** and **#9**.

```
SEPARATE #DATE1 INTO #1 #2 #3 WITH DELIMITER ':'
```

For this SEPARATE statement Natural Engineer will impact **#DATE1** as a specified impact and it will then derive the source fields **#1**, **#2** and **#3**.

2

COMMON FACILITIES

Chapter Overview

This chapter describes some of the common facilities available within Natural Engineer.

The topics covered are:

1. General Selection
2. Remote Development Environments

General Selection

The General Selection is a multi-purpose screen, which provides a mechanism to assist in selection decisions within various Natural Engineer options.

The basic functionality of the General Selection screen is to provide a list of items relating to the option that has invoked the screen.

The types of lists available are:

- FUSER Natural Libraries
- Natural Objects
- Dataitems
- Data Definition Modules (DDMs)
- Data Definition Module Fields
- Text Logic Members (TLM)

General Selection Window

The General Selection screen utilizes a common format used for all types of selection lists.

Each of the variations of the General Selection screen are now illustrated. The subsequent description is only shown once, but equally applies to all variations.

The following Figure 2-1 illustrates the General Selection screen displaying FUSER Natural Libraries.

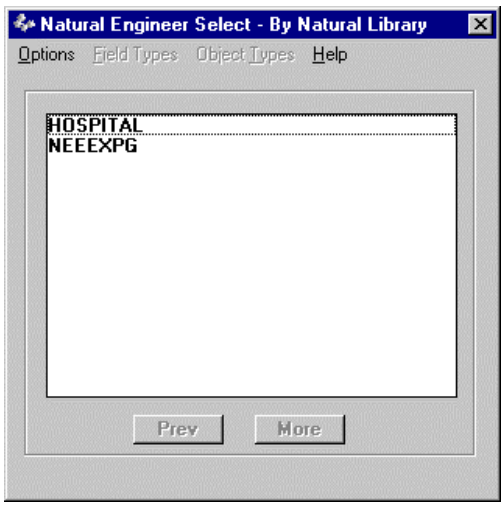

Figure 2-1 General Selection screen displaying FUSER Natural Libraries

The following Figure 2-2 illustrates the General Selection screen displaying dataitems.

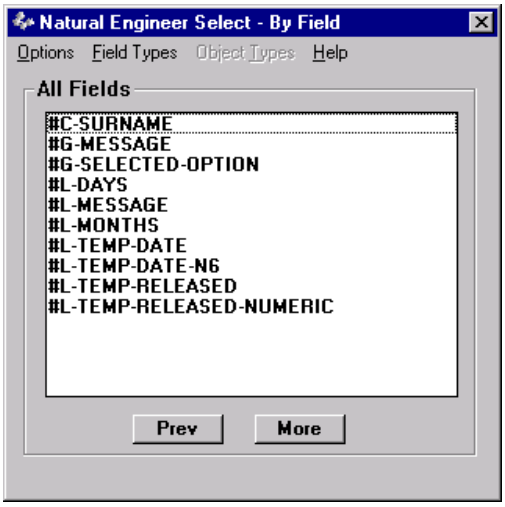

Figure 2-2 General Selection screen displaying dataitems

The following Figure 2-3 illustrates the General Selection screen displaying Natural Objects.

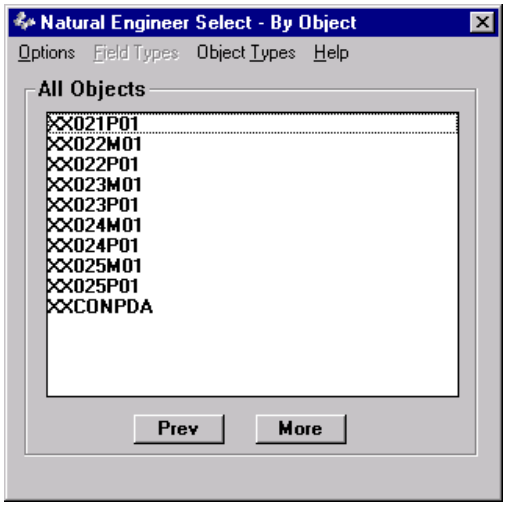

Figure 2-3 Natural Engineer Selection screen displaying Natural Objects

The following Figure 2-4 illustrates the General Selection screen displaying Data Definition Modules.

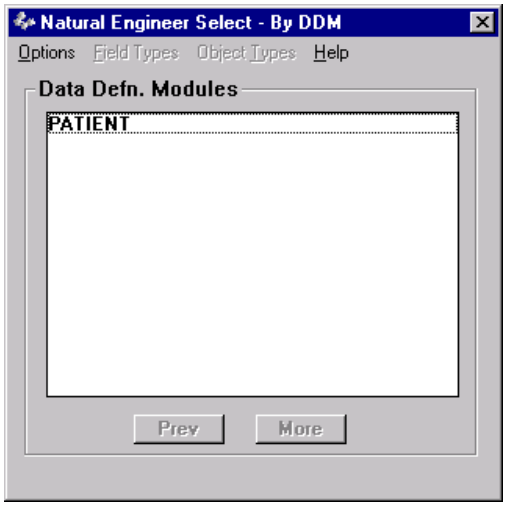

Figure 2-4 General Selection screen displaying Data Definition Modules

The following Figure 2-5 illustrates the General Selection screen displaying Data Definition Module fields.

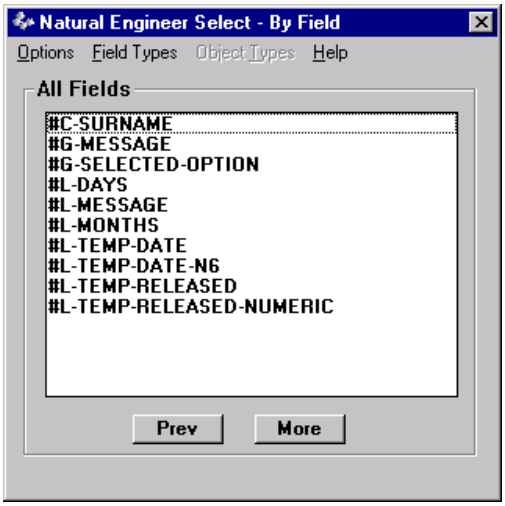

Figure 2-5 General Selection screen displaying Data Definition Module fields

The following Figure 2-6 illustrates the General Selection screen displaying Text Logic Members (TLM).

Figure 2-6 General Selection screen displaying Text Logic Members (TLM)

MENU ITEMS	OPTIONS	DESCRIPTION
Options	Provides various screen display options.	
	Show TLMs from SYSTEM	Display TLMs from the Natural library SYSTEM. *Note: This option is only available when the General Selection screen has been selected to display TLMs.*
	Show TLMs from Modification Library	Display TLMs from the Modification Library as defined in the Application Preferences screen. *Note: This option is only available when the General Selection screen has been selected to display TLMs.*
	Change Start Position of List	Allows the list of items to be restarted from a particular item.
	Close	Will close the General Selection screen and return you back to the previous screen.
Field Types	Allows you to select the types of fields to be listed. Available selections are: ▪ **All Fields** ▪ **Non-Database Fields** ▪ **Database Fields** ▪ **System Variables** *Note: This menu is only available when the General Selection screen has been selected to display Dataitems.*	
Object Types	Allows you to select the Object Types to be listed. Available selections are: ▪ **All Objects** ▪ **Programs** ▪ **Maps** ▪ **Data Defn. Modules** ▪ **Parameter Data Areas** ▪ **Global Data Areas** ▪ **Local Data Areas** ▪ **Copycodes** ▪ **Subprograms** ▪ **Subroutines** ▪ **Helproutines** ▪ **Dialogs** ▪ **Classes** *Note: This menu is only available when the General Selection screen has been selected to display Objects.*	
Help	Invokes the General Selection help.	

SCREEN ITEMS	DESCRIPTION
Items List	List of items. The items can be any of the following: ▪ **FUSER Natural Libraries** ▪ **Natural Objects** ▪ **Dataitems** ▪ **Data Definition Modules** ▪ **Data Definition Module Fields** ▪ **Text Logic Members** Items can be selected by using the left hand mouse button with a double click. *Note: For further information refer to the section Invoking the General Selection Window.*

BUTTON NAME	DESCRIPTION
Prev	Scrolls the element list to previous page. This button will be available/unavailable depending on the value specified in the LISTBOXMAX parameter in the NATENG.INI file.
More	Scrolls the element list forward one page. This button will be available/unavailable depending on the value specified in the LISTBOXMAX parameter in the NATENG.INI file.

Note: For more information on the NATENG.INI file parameter LISTBOXMAX refer to Chapter 1 in the Natural Engineer Administration Guide for Windows manual.

Invoking the General Selection Window

The General Selection screen can be invoked by using the selection button found adjacent to the input boxes, within applicable Natural Engineer screens.

The following Figure 2-7 illustrates the selection button used to invoke the General Selection screen.

Figure 2-7 Selection button used to invoke the General Selection screen

The following Figure 2-8 illustrates the selection button as it appears on the Application Preferences screen.

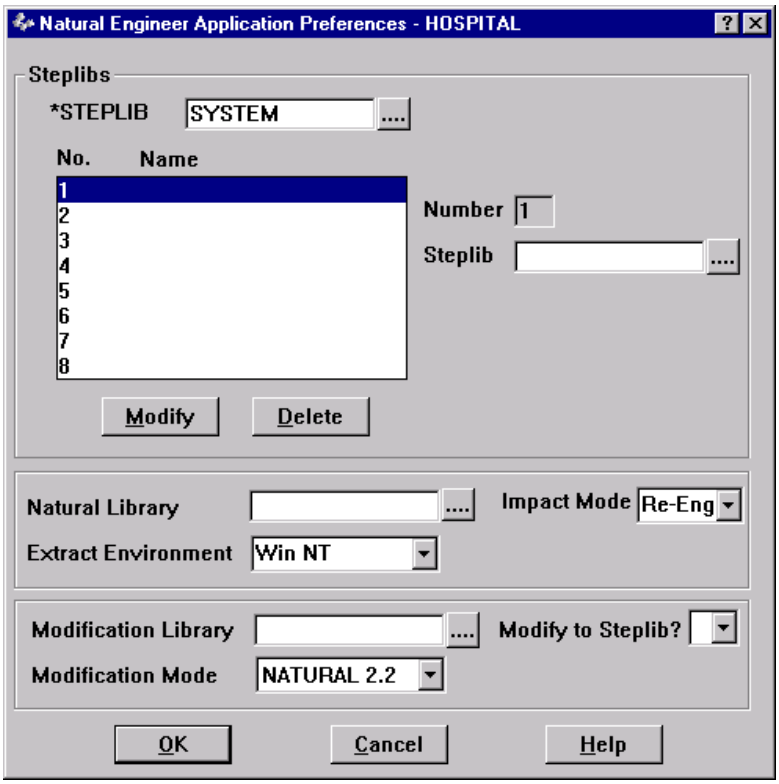

Figure 2-7 Selection button as it appears on the Application Preferences screen

The following table shows all the Natural Engineer options that can invoke the General Selection screen and the type of list that will be displayed:

Natural Engineer Process	Screen Options	Type of Items
Open Application	Select Application name	FUSER Natural Libraries.
Application Preferences	*STEPLIB	FUSER Natural Libraries.
	Steplib	FUSER Natural Libraries.
	Natural Library	FUSER Natural Libraries.
	Modification Library	FUSER Natural Libraries.
Impact Criteria	Keyword Value:	
	ADJUST OBJECT BUILDER	Objects loaded in the Repository for an application.
	DBFILE DELETE DELETE ? DELETE FROM READ READ ? READ WHERE READ WITH STORE UPDATE	DDMs loaded in the Repository for an application.
	Search Value:	
	DATAITEM	All dataitems loaded in the Repository for an application. This includes non-database fields, database fields and system variables.
	DBFILE DELETE DELETE ? DELETE FROM READ READ ? READ WHERE READ WITH STORE UPDATE	DDM fields loaded in the Repository for an application.

Natural Engineer Process	Screen Options	Type of Items
	Replace TLM	All objects with object type of 'Text' either from the Modification library for the application, or the Natural library SYSTEM.
Compare	Base Library	FUSER Natural Libraries.
	Compare 1 Library	FUSER Natural Libraries.
	Compare 2 Library	FUSER Natural Libraries.
	Base Object	Objects from the FUSER Natural Library.
	Compare 1 Object	Objects from the FUSER Natural Library
	Compare 2 Object	Objects from the FUSER Natural Library

TLM List Context Menu

When using the General Selection screen to list TLMs, a context menu is available by using the right hand mouse button with a single click.

This option provides the facility to view the contents of the selected TLM using the GenSource source code window.

The following Figure 2-9 illustrates the TLM list context menu.

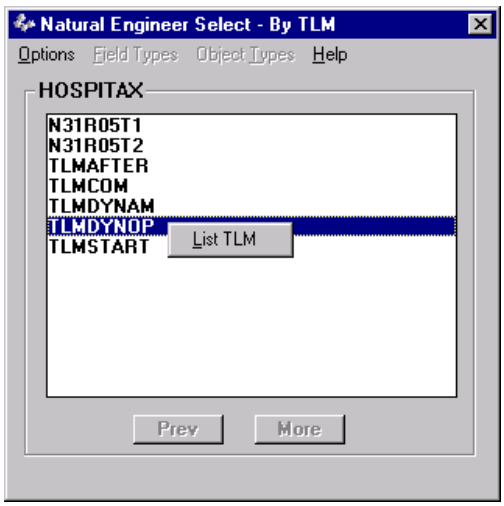

Figure 2-9 TLM list context menu

The following Figure 2-10 illustrates the GenSource source code window displaying TLM contents.

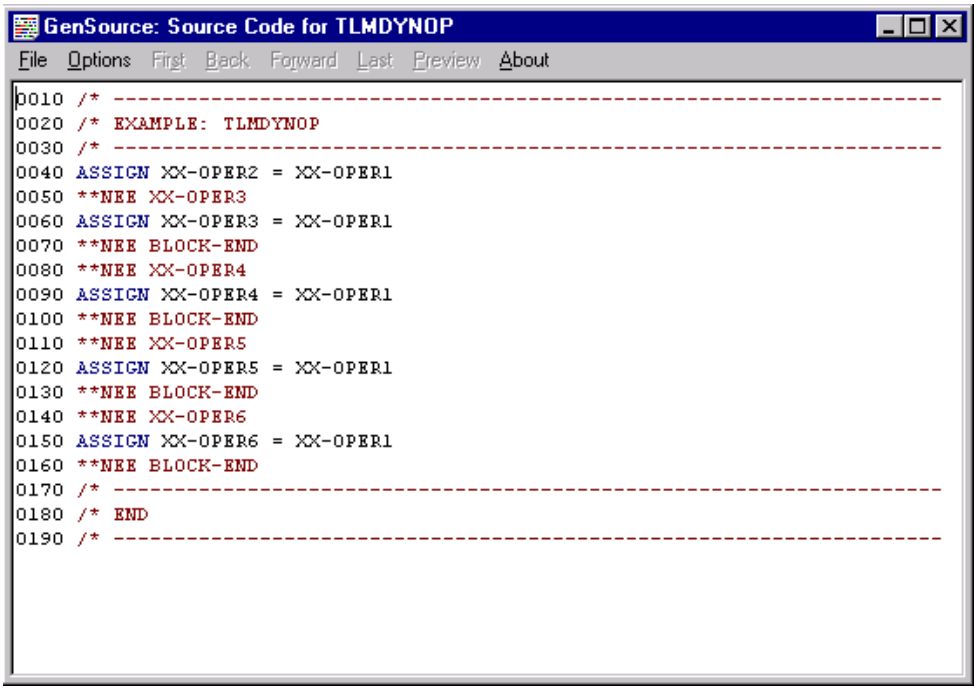

Figure 2-10 GenSource source code window displaying TLM contents

Remote Development Environments

Natural Engineer offers compatible support for remote development environments using Natural's Single Point of Development (SPoD) available with Natural version 5.1.1.

This facility provides the mechanism to access Natural source code libraries residing on remote mainframe computers, whilst retaining the full functionality and GUI of Natural Engineer for Windows.

This in-turn provides full Natural Engineer graphical functionality previously only available in Natural Engineer for Windows, to the mainframe environment.

Using the SPoD environment, Natural Engineer will read Natural source code from the remote mainframe environment. Any modifications applied using Natural Engineer's modification options will be written back to the relevant Natural libraries on the remote mainframe environment.

It also logically follows that if the SPoD environment is being used, then the Natural Engineer Repository file being used should reside on the mainframe.

Note: When using Natural Engineer in a SPoD environment, Natural Engineer version 4.3.1.1 or above must be installed in the Windows environment being used at run time. The same version of Natural Engineer must also be installed in the mainframe environment being used at run time.

Location of the Repository File

The location of the Repository file when utilizing the SPoD environment for Natural Engineer can be in any one of three locations:

1. On the mainframe

The Repository file can be located on the remote mainframe environment.

To utilize this location, the following considerations are required:

- The INI file parameter NETWORK needs to be set to N in the NATENG.INI file being used on the PC running Natural Engineer.
- The Natural Parameter file used to map the remote mainframe environment needs to have the correct settings for LFILE 96 (Repository file).

  Note: The Natural Parameter file for the Natural session being used to run Natural Engineer on a local PC will also need to have the LFILE 96 set to the same database id and file number used on the mainframe. Entire Net-Work is then used to connect from the local PC to the mainframe where the Repository file is located. This is a temporary requirement until the next Natural Engineer version release.

2. On a local PC

The Repository file can be located on a local PC.

To utilize this location, the following considerations are required:

- The INI file parameter NETWORK needs to be set to Y in the NATENG.INI file being used on the PC running Natural Engineer.
- The Natural Parameter file used for the Natural session being used to run Natural Engineer on a local PC needs to have the correct settings for LFILE 96 (Repository file).

3. On a networked server.

The Repository file can be located on a networked server.

To utilize this location, the following considerations are required:

- The INI file parameter NETWORK needs to be set to Y in the NATENG.INI file being used on the PC running Natural Engineer.
- The Natural Parameter file used for the Natural session being used to run Natural Engineer on a local PC needs to have the correct settings for LFILE 96 (Repository file).
- Entire Net-Work is then used to connect from the local PC to the networked server where the Repository file is located.

Mapping the Remote Mainframe Environment

To successfully utilize the SPoD environment for Natural Engineer, Natural parameter settings need to be made which will be invoked when using the Map Environment function in Natural version 5.1.1.

The key parameter settings for Natural Engineer are:

1. LFILE 96

   To access the mainframe Natural Engineer Repository, LFILE 96 needs to be defined with the required database id and file number.

2. LFILE 97

   To access the correct FUSER where the source code is located, LFILE 97 needs to be defined with the required FUSER database id and file number.

3. TABA1 (EBCDIC to ASCII Translation Table)

   Used to translate any special characters between the mainframe and the PC. For example:

   ```
   TABA1=(7C,40)
   ```

   This represents the '@' sign. The EBCDIC value is 7C and the ASCII value is 40.

4. TABA2 (ASCII to EBCDIC Translation Table)

   Used to translate any special characters between the PC and the mainframe.

   ```
   TABA2=(40,7C)
   ```

   This represents the '@' sign. The ASCII value is 40 and the EBCDIC value is 7C.

Note: The TABA1 and TABA2 parameters only need to be specified if the standard SPoD translation tables do not handle all special characters, for example umlauts, tilda sign exclamation mark etc. Any translation problems that may occur are related to the SPoD environment rather than Natural Engineer.

The parameter settings are applied using the 'Startup Session Parameters' specified in the Map Environment function within Natural version 5.1.1.

The session parameters can be defined in one of three ways:

1. As a dynamic individual parameter. For example

   ```
   LFILE=(00097,11177,02008)
   ```

2. As a defined profile using SYSPARM. This would then be referenced using the 'PROFILE=' parameter. For example:

   ```
   PROFILE=NEESPOD
   ```

3. Use the NTSYS macro within NATPARM. This is compiled into the Natural environment itself and will automatically invoke the correct settings. This would then be referenced using the 'SYS=' parameter. For example:

   ```
   SYS=NEESPOD
   ```

Invoking Natural Engineer under Natural Version 5.1.1

There are two ways to invoke Natural Engineer under Natural version 5.1.1:

1. **Local**

   Natural version 5.1.1 is started and the start up program "NEESTART" invoked from the Natural command line. This will invoke Natural Engineer using the local environment for the machine on which it is being executed. All Natural source code will be referenced from the local FUSER.

2. **Remote**

   Natural version 5.1.1 is started and the remote mainframe environment needs to be mapped (i.e., logged onto from within Natural version 5.1.1). The start up program "NEESTART" is then invoked from the Natural command line. This will invoke Natural Engineer using the remote environment that has been mapped. All Natural source code will be referenced from the remote environment FUSER.

To assist the User as to which mode (local or remote) is being utilized there is new status bar pane available on the main Natural Engineer screen. This will show 'Local' for the local FUSER environment, and 'Remote' for the remote environment FUSER.

Note: This will only appear if Natural version 5.1.1 has been installed.

The following Figure 2-11 illustrates Natural Engineer running under Natural version 5.1.1 using the local environment.

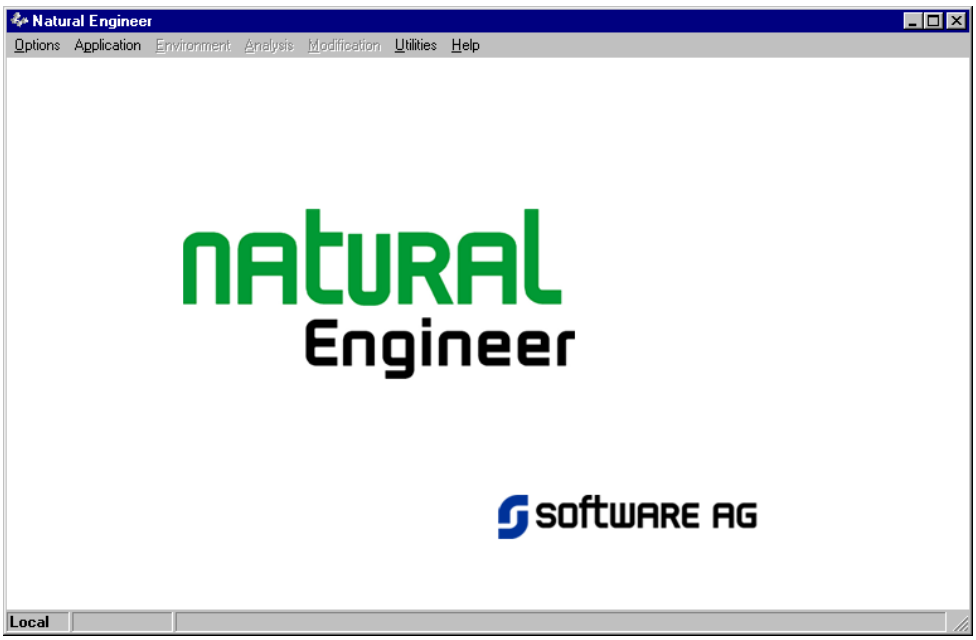

Figure 2-11 Natural Engineer running under Natural version 5.1.1 using local environment

The following Figure 2-12 illustrates Natural Engineer running under Natural version 5.1.1 using the remote environment.

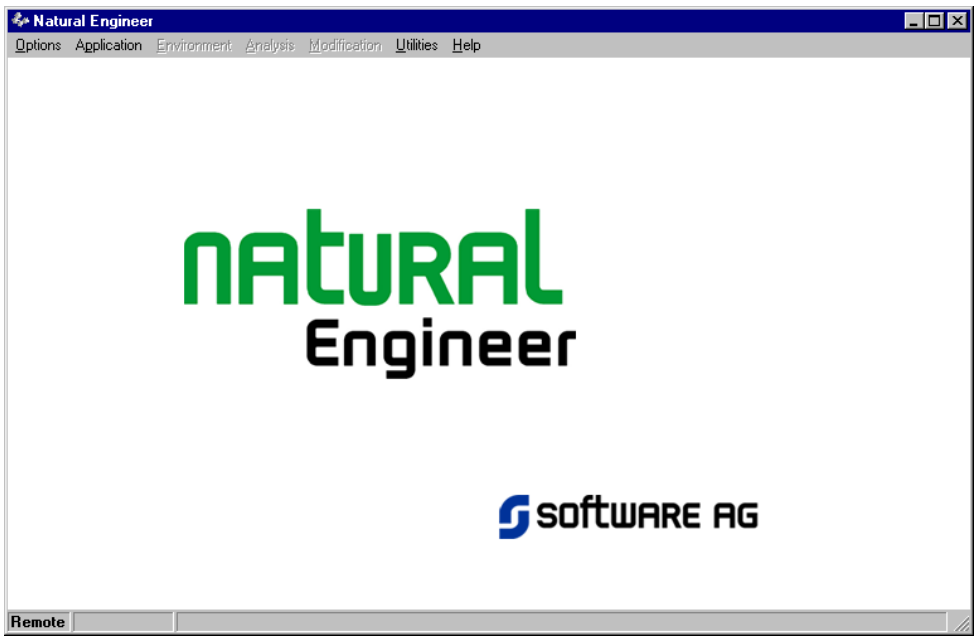

Figure 2-12 Natural Engineer running under Natural version 5.1.1 using remote environment

Note: For more information on Natural's Single Point of Development for remote development on a mainframe computer refer to the SPoD specific documentation on the Natural documentation CD.

INDEX