

CORE A Guided Tour



CORE®: Product & Process Engineering Solutions

Copyright © 1993-2000 Vitech Corporation. All rights reserved. No part of this document may be reproduced in any form, including, but not limited to, photocopying, translation into another language, or storage in a data retrieval system, without prior written consent of Vitech Corporation.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Vitech Corporation 2070 Chain Bridge Road, Suite 320 Vienna, Virginia 22182-2536

CORE[®] is a registered trademark of Vitech Corporation. Other product names mentioned herein are used for identification purposes only, and may be trademarks of their respective companies.

Last revision date: December 2000

Table of Contents

INTRODUCTION	5
EXAMINING CORE	7
AN OVERVIEW OF PRODUCT & PROCESS ENGINEERING	8
THE SAMPLE PROBLEM	9
The Collection Management System	9
GETTING STARTED WITH CORE - OPENING CORE TRIAL	. 10
CORE CONTROL PANEL	. 10
CORE Control Panel Window	10
DATABASE BROWSER	. 11
Database Browser – Empty	11
IMPORTING CORE DATA	. 11
OPENING THE DATABASE BROWSER	. 13
Database Browser – Loaded	13
Opening a Text View	. 14
Text View of Perform Collection Management	14
VIEWING CLASS ELEMENTS	. 15
Views Pull-down Menu	15
VIEWING ER AND ERA DIAGRAMS	. 15
VIEWING ER AND ERA DIAGRAMS	. 16
Element Relationship Attribute (ERA) View	17
OPENING THE DATABASE EDITOR	. 18
Database Editor	18
SAVING CORE DATA	. 19
BUILDING A CORE DATABASE	. 21
CAPTURING THE PROBLEM AND THE ORIGINATING REQUIREMENTS	. 22
CAPTURING THE DOCUMENT ELEMENT	. 23
Element Extractor Window	23
SAVING THE DOCUMENT ELEMENT IN THE DESIGN REPOSITORY	. 24
Element Extractor Window with Document Type Set to Originating Requirements	24
EXTRACTING ORIGINATING REQUIREMENTS.	. 25
Element Extractor Window with Reset Attributes Selected	25
EXTRACTING THE TOP-LEVEL REQUIREMENT	. 26
General Requirements – ORD.1	26
DEFINING A RELATIONSHIP	. 27
EXTRACTING THE CHILD-LEVEL ORIGINATING REQUIREMENTS	. 29
EXTRACTING THE CHILD LEVEL ORIGINATING REQUIREMENTS	. 30
VIEWING THE HIERARCHY IN THE BROWSER	31
VIEWING A TRACEABILITY HIERARCHY	. 32
VIEWING A TRACEABILITY HIERARCHY DIAGRAM	. 33
Adding Elements in a Traceability Hierarchy	. 34
VIEWING A TRACEABILITY HIERARCHY	. 35
ENHANCING THE SYSTEM DEFINITION WITH ISSUES	36
DEFINING THE SYSTEM	. 39
DEFINING THE SYSTEM AND ITS BOUNDARIES	. 40



CREATING EXTERNAL SYSTEMS	41
DEFINING THE UNIVERSE COMPONENT	42
ADDING DETAIL TO THE EXTERNAL SYSTEMS	43
VIEWING A PHYSICAL HIERARCHY	44
CREATING FUNCTION ELEMENTS	45
CREATING FUNCTION ELEMENTS	45
CREATE ROOT FUNCTIONS FOR THE SYSTEM/EXTERNAL SYSTEMS	46
BUILDING A FUNCTIONAL MODEL	47
A NOTE ABOUT INSERTION POINTS AND SELECTING OBJECTS AND BRANCHES	48
INSERTING A PARALLEL STRUCTURE	49
BUILDING A FUNCTIONAL MODEL	50
ADDING INPUTS AND OUTPUTS	51
THE N2 CHART SHOWS THE INTERFACES FOR OUR SYSTEM	54
DERIVING THE FUNCTIONAL (BEHAVIOR) MODEL FOR OUR SYSTEM	55
ADDING INPUTS AND OUTPUTS (N2 CHARTS)	60
DERIVING THE ENHANCED FUNCTIONAL (BEHAVIOR) MODEL	61
REVISITING/EXTENDING TRACEABILITY	63
EXTENDING TRACEABILITY	64
EXTENDING THE COMPONENT (PHYSICAL) HIERARCHY	65
EXTENDING THE COMPONENT (PHYSICAL) HIERARCHY	66
ALLOCATING THE FUNCTIONS	66
ALLOCATING THE FUNCTIONS	67
IMPACT ANALYSIS	67
IMPACT ANALYSIS	68
CAPABILITY OF CORE	69
GENERATING A REPORT	70
CONGRATULATIONS	78
CORE 3.0 TRIAL LIMITATIONS	79
CORE 3.0 PRODUCT FAMILY	80

Introduction

Thank you for your interest in CORE - *the* product and process engineering tool for the PC. CORE is an affordable tool that runs under the Windows OS to support system engineers during the initial phases of system definition, analysis, and design. Using the Windows interface enables CORE to be easier to learn and easier to use than its predecessors. The user-friendly interface allows efficient manipulation and representation of the system definition data. These interfaces include a *database browser*, a *system database editor*, an *element extractor*, and *engineering views* - *text*, *hierarchy*, *entity-relationship* (*ER*), *entity-relationship-attribute* (*ERA*), *function flow block diagram* (*FFBD*), *enhanced function flow block diagram* (*EFFBD*), and *N2* (*interface*) *charts*.

- This trial version* of **CORE** includes a sample database, **Collection Management System**, as it has been captured in the tool. The sample solution is presented in the data (ASCII) file: *CollMgt.rdo* in the SAMPLES directory.
- You can use this guide to recreate the solution yourself from scratch as we go along (starting on page 10) in **CORE**.
- This example is not intended to demonstrate the full power and flexibility of CORE. Instead, it serves as a simple, *structured walkthrough* of a sample product and process engineering problem in order to introduce you to the basic concepts and capabilities of CORE.
- In addition to top-down system engineering, CORE supports reverse (bottom up) and middle-out (incorporating legacy elements) engineering processes/paradigms. Some of CORE's most noteworthy applications have involved middle-out and reverse engineering solutions.
- While **CORE** supports the complete system engineering process in both *model-driven* and *document-driven environments*, relaxing the behavior modeling (from executable down to just hierarchies) provides a very powerful *Requirements Management* (*RM*) tool.
- To get a quick, but comprehensive, understanding of **CORE**, Vitech recommends our three-day introductory class. Please contact Vitech for more information on our training classes.
- We will be more than happy to provide additional information regarding our quick start and evaluation versions of CORE, product demonstrations, and our classes on product and process engineering and the application of CORE.

Vitech Corporation

2070 Chain Bridge Road, Suite 320 Vienna, Virginia 22182-2536 (703) 883-2270 FAX: (703) 883-1860 Email: info@vtcorp.com Support: support@vtcorp.com WWW: http://www.vtcorp.com

*The trial version is a limited 3.0 version of CORE. The current distributed version offers many enhancements and features not demonstrated in this manual.



This Page Intentionally Left Blank.





Examining CORE

In this section, you get a feel for the CORE Tool

- See how CORE applies to System Engineering
- Launch CORE •
 - Look at basic CORE Menus and Windows
- Import a data file into CORE Trial ٠ ٠
 - Export (Save) a data file from CORE Trial

•



CORE: The Complete Process

The diagram above represents the **CORE** paradigm for product and process engineering. It serves as a foundation for our walkthrough.

- 1. We will capture the *source document*, which is the starting point for top-down engineering.
- 2. We will capture the *originating requirements* from the source documentation.
- 3. We will define our system and its boundaries.
- 4. We will derive the system behavior (functional) model while extending the physical architecture and allocating all behavior onto the physical architecture.

By doing this we will establish and maintain traceability between and among the relevant design elements, identify and resolve critical issues, and provide documentation as we progress.

CORE®

The Sample Problem

In this guide, we examine a sample database for a **Collection Management System**. This system can be thought of as a simple library or resource center. The context diagram below provides a high-level view of the system we will use throughout this guide. You may find it helpful to refer back to this diagram as you build the system structure.

We have a database of collected information.



CORE®

The Collection Management System

The **Collection Management System** controls the inflow and outflow of this collection. **Customers** request information from the collection. The **Collection Management System** processes the customer requests and either returns information to them or provides a status of when they can expect a response. When the requested information is not in the system, the **Collection Management System** tasks **Collectors** to locate the information. (We can think of *Collectors* as Reporters or Researchers.) When the *Collectors* locate information, they pass the data to the *Collection Management System*, which in turn passes it on to satisfy the **Customer** request.

Getting Started with CORE - Opening CORE Trial

Once you have installed the CORE Trial, launch the CORE 3.0 Trial application.

- Click the **START** button. •
- Select the Programs Menu, proceed to the CORE 3.0 Trial menu, and click CORE 3.0 Trial.



CORE Control Panel

CORE opens with the CORE Control Panel. The CORE Control Panel provides pull-down menu and toolbar access to key user interface windows and access to the main menu commands. It is from this window that you navigate your CORE database.

New in CORE 3.0 is the addition of *tabbed sheets*, *toolbars*, and *icons*. The CORE Control Panel contains tabbed sheets grouping common user commands. Click on a tab to access the icon button commands for that tab. Toolbars are present in all CORE windows and contain icons to access frequently used commands.



CORE Control Panel Window

Database Browser

In general, the **DATABASE BROWSER** command opens a window to view the data structure, the **DATABASE EDITOR** command opens a window to view the data structure and to make changes, and the **ELEMENT EXTRACTOR** command opens a window allowing you to extract data from an electronic document text file into the database.

If you click the **DATABASE BROWSER** icon at this point, you see the **CORE** data structure, but there is no data in the database.



Database Browser – Empty

• Close the Database Browser Window

Importing CORE Data

To introduce you to **CORE**, we begin by importing some data. We will use a data file that was created in **CORE** and then exported to the Samples directory of our file system. In general, this import/export capability of **CORE** allows you to transfer a **CORE** database from one computer to another or to make a backup copy of the data. Here, we want to import data so we can see the **CORE** application with data.

To Import a CORE Data File:

 From the CORE Control Panel, select Database > Import
 Database.
 (or you can click the DB MGMT tabbed sheet and click Import
 Database button.)

🌘 C	ORE Control Pane	l (Trial Version)	_ 🗆 🗙
<u>F</u> ile	<u>D</u> atabase <u>S</u> chema	<u>Project</u> Scripts <u>Utilities</u> Pa <u>n</u> el <u>Options</u>	<u>W</u> indow
F Proje	<u>B</u> rowser <u>E</u> ditor Element E <u>x</u> tractor	Mode: • Database) Schema
Ger	Import Database	t Schema Utilities	
D.	Export Bridges	Lement	
	Elase		
	Set <u>F</u> acility		
	Script Editor Run Script	Exit	
RWD	A Database Mod	de T	11:20 AM

This opens the Import Database File dialog.

CORE®

Database Browser (cont.)

Note

By default, the Import Database File dialog directs you to the Data directory since this is typically where data files are stored. However, in the Trial, we have data in a Samples directory.

- Navigate to the CORE30 Trial/Samples directory and highlight the file named Collection Management.rdo.
- Click **OPEN**



The *Loading Database* dialog indicates activity while importing a file.

Loading Database	
	Loading database from file

Opening the Database Browser

Now that we have imported the file *Collection Management.rdo*, let us look at what we have in the database.

• From the CORE Control Panel, select Database > Browser to open a Database Browser window.

The left pane lists the *classes* that are used to describe the system.

Database Browser for System Engin	eering (Trial Version)	×
Pile Fojder Element views options	122	
Classes	Elements	
System Engineering	All Elements	
Category CompletionCriterion (4/4) Component (5/5) Constraint (1/1) Constraint (1/1) Constraint (2/2) Constraint (2/2) Constraint (2/2) Constraint (1/14) Co	0 Perform Collection Management 1 Accept & Format Request 2 Acquire Product 2.1 Check Product Inventory 2.2 Prioritize Request, Determine Coll 2.3 Accept Data From Collectors 2.4 Process Data And Put Product In II 3 Retrieve Product From Inventory 4 Provide Products To Customer 5 Compare Product To Request 6 Generate Discrepancy and Recomm c.1 Perform Collector Functions c.2 Perform Customer Functions u 1 Context (Reot Functions] 🏙 🖂 🍽 F F F A 🛛 🖬 🗗 🕂
Link (5/5) OriginatingRequirement (10, PerformanceIndex (1/1) Resource		
	Sort: Numeric 11:33 A	× • M //

Database Browser – Loaded

Notice the data that was added when we imported the *Collection Management.rdo* file. We have selected the *Function* class to list the elements in the Function class. Double-clicking on an element name will instantly open a *Text View* window of that element.

• From the Database Browser, select Function from the Classes pane.

Note

You can adjust the size of the Classes pane and Elements pane.

A yellow folder preceding the class name indicates that at least one element (instance) of that class has been defined.

The numbers in parenthesis indicate how many elements have been defined for that class and how many total elements in all subcategories of the class).

Opening a Text View

A **Text** *View* window provides the complete definition of a given element in the database by displaying all the *attribute values* and *relationship settings*. You can open a **Text** *View* of any database element to *view*, *add*, or *make changes* to the *attributes* and *relationships* of the displayed element. The attributes and their values are displayed in the upper portion of the window. The *relationships* and *targets* that complete the element definition are displayed in the lower portion of the window. Use the *scrollbars* on the right to view the complete list of *attributes* and *relationships*, respectively.

The list of *attributes* and *relationships* differs depending on the *class* of the element displayed. Here we are looking at an *Element* of the class named *Function*, so these attributes and relationships pertain to *Functions*.

• Double-click **Perform Collection Management** from the list of Function elements to open a **Text** *View* window of this element.

🔡 Perform Collecti	on Management asText (Trial Version) 📃 🗖	⊐ ×
<u>F</u> ile <u>E</u> dit <u>E</u> lement	<u>I</u> arget <u>V</u> iews <u>O</u> ptions	
🖪 🎒 🛄 В	/ 및 🛕 · 씁 급 Ճ 급 흡 🎜 🗸	
Name:	Perform Collection Management	
Number:	0	₽
Abbreviation:		Ĥ
	This function is the integrated logic for the	
Description:	system. It is not yet complete, and it does	
	inot sausty an me originaling requirements	
Duration:	Edit	 ₽
Timeout:		
Execution Level:	Follow Decomposition	
Relationship	s Targets & Attributes	r[²]r
💼 allocated to	System SYS.1 Collection Management Sy	-Fi-
annotated by		122
augmented by		强
captures categorized by	Sort: Numeric by class	* •
RWDA Friday, Ju	ly 07, 1995 at 11:12:53 PM 06:45 A	λM //

Text View of Perform Collection Management

Scroll through the Attributes

Scroll through the list of possible Relationships

• A ± indicates a *target attribute*. Click the ± to expand the list and view the attributes of the target.

Note You can adjust the size of the *Relationships* and *Targets & Attributes* panes.

Viewing Class Elements

We viewed the *Perform Collection Management* element attributes by double-clicking the element name in the **Database Browser** window and opening a **Text View** of the selected element.

For other views of the data, select the desired view from the **Views** menu or use the **Database Views** toolbar icons. The views listed in the **Views** pull-down menu are the same as those displayed in the **Database Views** toolbar.

V	iews P	ull-down Menu				
Perform Collection	on Man	agement asText (Ti	rial Versio	n) 💶 [⊐×	
<u>File E</u> dit <u>E</u> lement	<u>T</u> arget	<u>Views</u> <u>Options</u>				
🗄 🎒 🛄 В	ī II	<u>I</u> ext View	Ê	≠⊒ .		
Name:	Perfo	E <u>R</u> Diagram ER <u>A</u> View				
Number:	0	<u>H</u> ierarchy				
Abbreviation:		<u>P</u> hysical Block			Ĥ	
Description:	This f logic compl	FF <u>B</u> D Enhanced FFBD <u>N</u> 2	;	•		Database Views Toolbar
Duration:		1000		Edit	≣ª	, consu
Timeout:		Inspector L	Jtrl+₩		<u> </u>	
Execution Level:	Follow	/ Decomposition		-	151	
Priority:				•	чр.	
Relationship	s	Targets &	Attributes		÷	
constrained by		Function 1 Accept	t & Format	Requ 🔺	Ē	
consumes	·	Function 2 Acquire	e Product class	• •		
RWDA Friday, Ju	ıly 07, 19	95 at 11:12:53 PM		11:52 A	M //	

CORE®

Viewing ER and ERA Diagrams

<u>ER View</u>

- Select Perform Collector Functions from the list of elements in the Function class
- Click ER to open an ER View

The **ER View** is a graphical representation of a selected element. The element and its targets are represented as icons linked by relations.

e.2 c.1 Perform allocated to Collectors Collector Func... Function Component u.1 Context (Root decomposes Function For ... Function í.3 inputs tasking Item i.4 outputs data Item owned by System Engineer Engineer Date: Author: Friday, July 07, 1995 System Engineer Number: Name: c.1 (Trial) Perform Collector Functions

Element Relationship (ER) View

Close ER View

ERA View

The **ERA View** provides a composite of the **ER** and **Text Views**. The **ERA View** completely specifies an element by incorporating the strength of the graphical relation representation with the value of the tabular attribute representation. By having the views together, the user can manipulate *relationships* and *attributes* for the selected element.

- Click ERA to open an ERA View. You may find you need to resize or scroll through the window to see all the data.
- Explore more of the database by selecting other *Classes* and *Elements* and opening their respective views.

瞷	Perform Collector Function	ns asERA (Trial	Version)					
<u>F</u> ile	Edit <u>D</u> iagram E <u>x</u> tras <u>L</u> a	ayout <u>V</u> iews <u>O</u> p	tions		# 6. 0 . 0			
Ш		≗ * E´ C´ ×	a D 5 as A	· • •			a	
Ι.					Name:	Perform Collector Functions		
	(1)				Number:	c.1		
	Perform	allocated to	C.2		Abbreviation:		Ĥ	
	Collector Func					·	=	
	Function	Component				1		
			u.1 Context (Root Function For Function uts (1.3 tasking Item					
	2	lecomposes			옯			
						扇		
				Description:				
	L.	nputs				4		
						B		
						100 1001		
		outputs	eteh			- -		
		lten		Itom		Duration:	Edit	-
					Timeout:	<u></u>		
	l	wned by	System		Execution Level:	Follow Decomposition	Í	
	~		Engineer		Priority:			
			Engineer		Log Message:		1	
	Date:	Author:			Query Criterion:	Edit		
	Friday, July U7, 1995 Number:	Friday, July 07, 1995 System Engineer			Creator:	System Engineer		
	c.1 (Trial) Perform Collector Functions		Created:	Friday, July 07, 1995 at 12:00:00 /				
					Last Modified:	Friday, July 07, 1995 at 05:29:44 I		
RWDA Friday, July 07, 1995 at 05:29:44 PM 09:32 AM								

Element Relationship Attribute (ERA) View

• Close ERA View

CORE®

Opening the Database Editor

Another way to view and make changes to the data in the **CORE** database is with the **Database Editor**. The **Database Editor** combines the **Browser** and **Text View** windows into one— allowing you to view the structure of the data and make updates to the selected element in the same window.

- From the **CORE Control Panel**, click **DATABASE EDITOR** to open a *Database Editor* window.
- Take a moment or two to familiarize yourself with the layout of the Database Editor window.



Saving CORE Data

In the CORE Trial Version, we import and export the database with an .RDO extension.

Note

The actual CORE version imports and exports the CORE database with an .RDT file extension. The full version of CORE can also save an image with a .COR file extension allowing it to load faster. This feature is not accessible in the Trial version.

🌘 C	ORE Control Pane	
<u>F</u> ile	Database Schema	<u>Project</u> S <u>c</u> ripts <u>U</u> tilities Pa <u>n</u> el <u>O</u> ptions <u>W</u> indow
Proje Ger D. B	Browser Editor Element Extractor Open Simulator Import Database Export Bridges Erase Set Eacility Editor	Mode: O Database O Schema t Schema Utilities Export Database Export Changes Exit
RWD.	A Database Mo	de 03:08 PM

To Save Your CORE Trial Data to an .RDO File:

• Select Database > Export > Export Database from the CORE Control Panel menus to export CORE data.

An *Export Database* dialog prompts you for an *RDO* file name in the *Data* directory.

- We will name the export file Collection Management.rdo
- Click SAVE.

The *Saving Project* dialog indicates activity while exporting a file.

aving Project





Refer to this section to save/export and load/import your data as you proceed through the rest of the guide. When you need a break, or need to exit the CORE Trial, you will want to export you data to an RDO file and then import that RDO data file when you return to CORE in order to continue where you left.

Saving project to file..



This Page Intentionally Left Blank.





Building a CORE Database

In this section, you build a CORE database from scratch

- Extract a System Document into CORE
- Extract Originating Requirements into CORE
- Define Relationships
- View the Requirement Hierarchy

Capturing the Problem and the Originating Requirements

Now that you have seen how to import a *database file* and *view data*, we will see how to *build a database from scratch*.

Before proceeding with this section, be sure to START WITH AN EMPTY DATABASE. (CORE Trial is empty each time you launch the application.)

- Close any open **CORE** windows (except the **CORE Control Panel**).
- From the CORE Control Panel menus, select Database > Erase > Erase Database.
- Answer **Yes** to the warning.

We will start by capturing the *source document*. We want to put our entire source document into **CORE**. The **Element Extractor** is an easy way to transfer text from a text-based file into the **CORE** database structure. New with **CORE 3.0**, the **Element Extractor** now accommodates formatted text in either *RTF*, *HTML*, or *TXT* formats.

To accomplish this, we use **CORE's Element Extractor**.

• From the CORE Control Panel > General tabbed sheet, click the ELEMENT EXTRACTOR button.

For extracting, you need to load your source document. The source document must be DOC, RTF, HTML, HTM, or TXT file types.

• Select File > Load Document.

We will use the file *Collmgt.txt*, located in the *Samples* folder of the **CORE Trial** directory.

- Navigate to the Samples subdirectory
- Change the default Files of Type to Text Files (*.txt)
- Highlight Collmgt.txt.
- Click OPEN.

The *Collmgt.txt* file displays in the left pane while the element being extracted/created is built and defined in the right pane. The *attributes* and *relationship* fields vary depending on which *class* is selected from the *class pull-down selection list*.

Ele	ement Extra	ctor Window			$\overline{\mathbb{Q}}$
Element Extractor (Trial Version)					g
<u>File Edit Extractor Target Views Option</u>	ns				Ţ
🖪 🖨 🛄 🛩 в и ц 🛆 - с	🖞 Reset Attributes Cle	ear Relationships 🖏 🖕			~
Collingt.txt		Element Definition			H
COLLECTION	Class: Document	•			\cap
MANAGEMENT SYSTEM	Folder: Document	V	🔒		\mathcal{L}
1.0 SCOPE	Name:	COLLECTION MANAGEMENT SYSTEM		Transfer	\bigcirc
This source specification	Number:			 Buttons 	
establishes the performance,	Abbreviation:				
requirements for an intelligence		COLLECTION MANAGEMENT SYSTEM 🔺			
collection management system.	Descriptions		# 		
The mission of the Collection	Description:	This source specification establishes the	151		
Management System is to		performance, design, development, and test	r[r]r		
provide management of a system	Document Type:	nil 💌	$\mathbb{P}_{[\frac{k}{2}]}^{[\frac{k}{2}]}\mathbb{P}$		
of intelligence collectors, from	Document Number:		12		
the acceptance of user requests, through scheduling the collectors.	Document Date:		强		
to delivery of the collection	Govt. Category:	nil			
products to the users.	lon-Govt. Category	nil	-		
2.0 APPLICABLE	Relationship	s Targets & Attributes	1		
DOCUMENTS	annotated by				
The applicable documents for the Collection Management System	assigned to augmented by				
program are:	categorized by				
Collection	causes documents				
Management System source	generates	Sort: Numeric by class	1		
This element has not been created.	I OWNED by		M		
		Jondon			

The Element Extractor can extract text into any class of elements.

Select the **Document** class from the **Class** pull-down selection list. •

Many of the attributes for an Element can come directly from the source document. In general, to move text from the left pane to the element attribute fields, highlight the desired text in the left pane and press the transfer button corresponding to the desired attribute.

To add a *Description* to the Document

element:

- Resize the Element Extractor window in preparation for reviewing the source • material and loading it into the system design database.
- Highlight the entire contents of the *Collmgt.txt* file (Ctrl+A) ٠
- Click the **DESCRIPTION** transfer button in the right pane to transfer the selected • text to the Description attribute field.

To add a *Name* to the Document element:

In the Name field, type **COLLECTION MANAGEMENT SYSTEM**, or highlight the text COLLECTION MANAGEMENT SYSTEM in the left pane and click the NAME transfer *button* in the right pane to transfer the text to that attribute field.

Saving the Document Element in the Design Repository

The *Document Type* attribute is set by choosing from a predefined list of possible values (an enumerated list).

• Click on the *down arrow* next to the *Document Type* field, and choose **Originating Requirements** to reflect the source of this information.

Element Extractor Window with Document Type Set to Originating Requirements

Element Extractor (Trial Version)		Create	>
<u>File Edit Extractor Target Views Option</u>	20	Element	
🖫 🖨 🛄 😂 B Z U 🔬 🌔 🛙	🕯 🔚 teset Attributes Clear	Relationships	
Collmgt.txt		Element Definition	
COLLECTION	Class: Document		T
MANAGEMENT SYSTEM	Folder: Document		
	Name:	COLLECTION MANAGEMENT SYSTEM	
1.0 SCOPE This source specification	Number:	, 	
establishes the performance,	Abbroviation:	J [
design, development, and test	ANDI EVIALION.		ľ
requirements for an intelligence		SVSTEM	🗕 д
collection management system.	Description:		Ter I
The mission of the Collection		1.0 SCOPE	
Management System is to		This source specification establishes th	.e 🔟 🗐
provide management of a system	Document Type:	Originating Requirements	 4
of intelligence collectors, from	Document Number:		12
through scheduling the collectors,	Document Date:		14
to delivery of the collection	Govt. Category:	nil	-
products to the users.	Non-Govt. Category:	nil	_
2.0 APPLICABLE	Relationships	Targets & Attributes	
DOCUMENTS The applicable documents for the Collection Management System	annotated by assigned to augmented by categorized by		
Collection	causes		
Management System source	documents		
specification (this document)	owned by	Sort: Numeric by class	_
OLLECTION MANAGEMENT SYSTEM create	ed at 07:40:21 AM 🛛 🦟		07:40 AM
		Status	

Once all desired attributes are defined, we need to enter the *Document Element* into the design repository (database).

• From the Element Extractor window, click the CREATE ELEMENT icon on the Element Extractor toolbar to enter the *Element* in the database (or select Extractor > Create Element). The status line reflects when an element has been saved to the database.

We have finished defining the *attributes* of the *Document* element we named *Collection Management System.* We will talk about defining *relationships* between *elements* in the next few pages when we extract the *Originating Requirements Element.*

Extracting Originating Requirements

Our next step is to extract and define our *Originating Requirements* elements from the source document, *Collmgt.txt*. Notice that after we saved the Document element to the database (selected **Create Element**), the *attribute data* is still displayed in the fields. This saves you from having to re-enter it if you want to make use of some of the same data. In our case, at this point, we won't so we will want to clear the attribute fields before proceeding.

- Click the **Reset Attributes** command on the toolbar (or **Extractor > Reset Attributes** from the menus) to clear previous data from the fields.
- From the *Document Type* selection list select **Originating Requirements**.

🙀 Element Extractor (Trial Version)			×
<u>File Edit Extractor Target Views Option</u>	s constant of the second se		
🖪 🖨 🛄 🚄 B Z 🗉 🗛 🖻	Reset Attributes Clear	Relationships	
Collmgt.txt		Element Definition	
COLLECTION	Class: Document		ון
MANAGEMENT SYSTEM	Folder: Document		à
10 SCOPE	Name:		
This source specification	Number:		
establishes the performance,	Abbreviation:		
design, development, and test			5
collection management system.		Reset	
	Description:	Attributes	1
The mission of the Collection			ie I
Management System is to	Document Type:	Originating Requirements	i-
of intelligence collectors, from	Document Number:		2
the acceptance of user requests,	Document Number.		е Г
through scheduling the collectors,	Document Date:		ŧ.
nroducts to the users	Govt. Category:		ř.
	Non-Govt. Category:	l nii 🔄 🔄	•
2.0 APPLICABLE	Relationships	Targets & Attributes	
DOCUMENTS The applicable decuments for the	annotated by		
Collection Management System	assigned to augmented by		
program are:	categorized by		
Collection	causes documents		
Management System source	generates	Sort: Numeric by class	
Ispecification (this document)	l owned by		
This element has not been created.		07:41 AM	11.

Element Extractor Window with Reset Attributes Selected

The *attribute* and *relationship* fields will reflect attributes and relationships for an *Originating Requirement* element.



Extracting the Top-Level Requirement

During this extraction process, we want to establish a *hierarchy of requirements*. In our case, the *General Requirements* element incorporates a number of subordinate requirements. We will first establish a parent, or top-level, *Originating Requirement* from the GENERAL REQUIREMENTS section of the *Collmgt.txt* file. You may need to scroll down in the left pane to see this section of the text file.

- Highlight the words **GENERAL REQUIREMENTS** from the text in Collmgt.txt.
- Click the **NAME** *transfer button* to insert the text into the *Name* field, which saves you from having to type the name in the field.
- Type **ORD.1** in the *Number* field. (There was no short way to enter this data.)
- Highlight the text in *Collmgt.txt* within the heading **GENERAL REQUIREMENTS**, as shown in the figure below.
- Click the **DESCRIPTION** *transfer button* to insert the text into the *Description* field.

To be thorough, you can enter data in the other attribute fields as appropriate.

🛱 Element Extractor (Trial Version)) ×		
<u>File Edit Extractor Target Views Options</u>					
🔚 🎒 🛄 😂 🖪 🛛 🔲 🖆 💼 Reset Attributes Clear Relationships 🎼 🖕					
Collmgt.txt		Element Definition			
	Class: Originating	Requirement			
3.1 GENERAL	Folder: Originating	Requirement	r B		
1. The system shall accept	Name:	GENERAL REQUIREMENTS			
intelligence data collection requests	Number:	ORD.1			
from the certified users.	Abbreviation:		9 -55		
 The system shall retain an inventory of previously collected data/products and provide them to users, if appropriate. The system shall control multiple sensors and multiple types of sensors. The command center shall be staffed at a maximum of 12 personnel 	Description:	 The system shall accept intelligence data collection requests from the certified users. The system shall retain an inventory of previously collected data/products and provide them to users, if appropriate. The system shall control multiple sensors and multiple types of sensors. 	n + + 12 + -		
on any shift.	Weight Factor:		5761		
on the user's request within twenty	Paragraph Title:		150		
four hours.	^p aragraph Number				
 The system shall provide a means of prioritizing the user's requests 	Line Number:				
7. The system shall monitor and	Relationship	s Targets & Attributes			
assess its own performance.	documented by generates incorporated in incorporates	Sort: Numeric by class			
This element has not been created.		10:00 AM			

General Requirements – ORD.1

Defining a Relationship

For *traceability*, we want to establish that this **Originating Requirement** element (**ORD.1**) is documented by the Document element named *Collection Management System*. The <u>documented</u> <u>by</u> relationship identifies the source document which specifies and/or enhances the definition of the element.

• Double-click the <u>documented by</u> relationship in the Relationship pane (or select **Target > Edit Target** from the menus) to open a *Target Dialog* for the relationship.

"documented by" for GENERAL REG	QUIREMENTS (Trial Version)	
Target Classes	Targets	Add
🗅 Document (1/1)	All Elements	New
		Remove
Possible Targets		<u>0</u> K
All Elements		
Sort: Numeric	Sort: Numeric by class	

The **Edit Target** command is used to add one or more targets to the selected relationship. It *automatically creates the complimentary relation* linking the target with the selected elements.

Defining a Relationship (cont.)

The *Target Dialog* allows you to add or modify *relationships*. It lists the allowable target classes for the specified relation, which in this case is <u>documented by</u>. With a *Target Class* selected, its possible targets are listed. To make a *Possible Target* a *Target*, select a possible target and click **ADD** (or double-click on the possible target). As soon as the target is added, the new target is added to the *target* list. You can also click the **NEW** button to create a new target, if you want one that does not already exist.

Here, we will use the one we have.

- From the Target Classes, select Document
- From the Possible Targets, select Collection Management System
- Click the **ADD** button to add the document element to the targets list.
- Click **OK** to close the *Target Dialog*.



Returning to the **Element Extractor**, notice the target added to the <u>documented by</u> relationship. We are now ready to enter this *Originating Requirement* in the repository.

	Elle Edit Extractor Larget Views	Uptens Faset Attribute	s Clear Relationships 🔀 🗸	
	Collingt.txt		Element Definition	
	3.1 GENERAL	Class: OriginatingF	lequirement	
	REQUIREMENTS:	Folder: OriginatingF	lequirement	
	intelligence data collection	Name:	GENERAL REQUIREMENTS	
	requests from the certified	Number:	ORD.1	
-	users.	Abbreviation:		- E
eate ment con	 The system shall retain an inventory of previously collected data/products and provide them to users, if appropriate. The system shall control multiple sensors and multiple 	Description:	 The system shall accept intelligence data collection requests from the certified users. The system shall retain an inventory of previously collected data/products and provide them to users, if appropriate. The system shall control multiple sensors and 	F = F
	types of sensors. 4 The command center shall	Weight Factor:		Tet
	be staffed at a maximum of	Paragraph Title:		
	12 personnel on any shift.	Paragraph Number		<u> </u>
	5. The system shall provide	Line Number:		_
	request within twenty four	Relationships	Targets & Attributes	
	hours. 6. The system shall provide a means of prioritizing the user's requests. 7. The system shall monitor and assess its own performance.	categorized by causes documented by generates incorporated in incorporated in incorporates owned by traces to	Document COLLECTION MANAGEMENT SYSTE Notice the Fold indicating that a t been adde Sort: Numeric by class	er icon arget has ed.

• From the Extractor pull-down menu, click CREATE ELEMENT (or Ctrl+E)

Note: You can also click the Create Element icon.

Extracting the Child-Level Originating Requirements

Now that we have defined the top-level **Originating Requirement**, we will break out each of the seven General Requirements to create seven individual, child-level *Originating Requirements*. These requirements are <u>incorporated in</u> their parent, which we named **GENERAL REQUIREMENTS**. *Traceability* back to the source document, *Collection Management System*, is achieved through the parent (as defined with the documented by relationship).

To begin, we took the entire *General Requirements* section of our *Collmgt.txt* and placed it in an *Originating Requirement* class element that we numbered ORD.1; this will serve as our parent level. Now we will break up the section, placing each numbered requirement in its own element of the *Originating Requirements* class; this will serve as our child level.

We will give each of the seven requirements a number and name, such as:

ORD.1.1Accept RequestsORD.1.2Retain InventoryORD.1.3Control Multiple SensorsORD.1.4Maximum StaffORD.1.5Provide FeedbackORD.1.6Prioritize RequestsORD.1.7Monitor and Assess

Since we are creating child-level elements to the same class (*Originating Requirements*), we don't need to select another class from the selection list. All we need to do is reset the *attributes* and *relationships*.

- Click the Reset Attributes button to clear all the attribute fields.
- Click the Clear Relationships button to clear all the relationship fields.

Now we have a clean slate to use to begin transferring text and thus creating the first of our seven child elements.



Extracting the Child Level Originating Requirements

1 Type a name in the *Name* field. Using the list on the previous page, the name of the first child element is **Accept Requests**.

2 Establish a hierarchical numbering system for the child elements. Number the first element **ORD.1.1**. (The second will be **ORD.1.2** etc. as shown in the list on the previous page.)

3 Highlight the text for the description from the *Collmgt.txt* file. For the first child element, we highlight the text in the first *General Requirement*.

4 Click the **DESCRIPTION** *transfer button* to place the text in the *Description* field.

Establish the incorporated in relationship to its parent, **GENERAL REQUIREMENTS**.

5 Double-click <u>incorporated in</u> from the *Relationships* pane to open the *Target* Dialog.

6 Highlight **OriginatingRequirement** from the list of *Target Classes*.

7 Double-click **ORD.1 GENERAL REQUIREMENTS** from the *Possible Targets* to add it as a target.

8 Click **OK** to close the *Target Dialog.*

9 From the **Element Extractor**, click the **CREATE ELEMENT** *icon* to store this element definition.



• Click the **Reset Attributes** to clear the attributes values, but **DO NOT** clear the relationships. The relationship you established for the first child element can be used for the subsequent elements.

	-		
	Reset Attributes Clear H	elationships 70	
Collmgt.txt		Element Definition	🗉
	Class: Originating	Requirement	_ ₽
3.1 GENERAL	Folder: Originating	Requirement	-
The system shall accent	Name:	Accept Requests	
ntelligence data collection requests	Number:	0RD.1.1	
from the certified users.	Abbreviation:		—
 The system shall retarn an nventory of previously collected data/products and provide them to users, if appropriate. The system shall control multiple sensors and multiple types of sensors. The command center shall be 	Description:	 The system shall accept intelligence data collection requests from the certified users. 	
staffed at a maximum of 12 personnel	Weight Factor:		Ę
on any shift. S. The system shall provide feedback	Paragraph Title:		F
on the user's request within twenty	Paragraph Numbe	r	_
four hours.	Line Number:		- 1
5. The system shall provide a means	Belationshi	ns Targets & Attributes	
7. The system shall monitor and assess its own performance.	incorporated in incorporates owned by traces to verified by	OriginatingRequirement ORD.1 GENERAL REQUI	REN •

Create the other six child elements using the *Numbers* and *Names* from page 28. You can skip steps 5-8 each time through since you didn't clear the relationships.

Viewing the Hierarchy in the Browser

Let's check what we have done by looking at the elements we created from a Database Browser window.

We can close the **Element Extractor** window; we are done with it for now.

- Using the Views *pull-down menu*, open a **Database Browser** window.
- From a Database Browser, select the Originating Requirements class.

Notice the Elements we created.

🛄 Database Browser for System Enginee	ring (Trial Version)	
<u>File Folder Element Views Options</u>		
🖪 🖧 🛄 🍣 🗞 🖬 🗗 🗗 🍯	ÉĆ.	
Classes	Elements	
System Engineering	All Elements	
🗖 🗅 Category 📃	ORD.1 GENERAL REQUIREMENTS	. 🖿
🗅 CompletionCriterion	ORD.1.1 Accept Requests	
🗅 Component	ORD.1.2 Retain Inventory ORD.1.3 Control Multiple Sensors	
🗅 Constraint	ORD.1.4 Maximum Staff	FE
📥 Document (1/1)	ORD.1.5 Provide Feedback	
DomainSet	ORD.1.6 Prioritize Requests ORD.1.7 Monitor and Assess	
	OND.1.7 Monitor and Assess	- A
Giossary		नि
C Itom		r[]=
		- Fia
OriginatingRequirement (8(8)		1753
PerformanceIndex		5-1
	1	
Risk 🗾	Sort: Numeric	* -
	10:21	AM

Viewing a Traceability Hierarchy

Now that the seven child-level *Originating Requirement* elements have been created, let us view a *Traceability Hierarchy Diagram* from our source document. Recall that our **Originating Requirements ORD.1** is <u>documented by</u> our source Document named Collection Management System. This means that our Document <u>documents</u> our Originating Requirements. Let's see a diagram of this.

- From the **Database Browser**, select **Document** from the *Classes* pane.
- Select **COLLECTION MANAGEMENT SYSTEM** from the *Elements* pane.

🛄 Database Browser for System Enginee	ering (Trial Version)	
<u>File Folder Element Views Options</u>		
🖪 🖨 🛄 🍣 🗞 🗅 🖶 🗗 🗙		
Classes	Elements	
System Engineering 🗾	All Elements	
👜 🗅 Category 📃	COLLECTION MANAGEMENT SYSTEM	
🗅 CompletionCriterion		
🗅 Component		Ŧ
🗅 Constraint		
🖴 Document (1/1)		
🗅 DomainSet		
- C Function		4
- 🗅 Glossary	X	E
🗅 Interface		5
🗅 Issue		
Calltern	Hierarchy	-E
🗅 Link	icon	-9
OriginatingRequirement (8/8)		112
- PerformanceIndex		恒
Resource	Sert: Numeric	1
		*
RWDA Friday, December 01, 2000 at 10:03	3:59 AM [10:237	AM
Click the HIERARCHY icon from from the menus to open a Hi er	n the Database Views <i>toolbar</i> or select Views > Hi erarchy dialog.	era
Definitions:		
	Cancel	
Save	Delete	

The *Hierarchy Definition Dialog* provides selections for building various hierarchy diagrams. We will select a *Definition* from the enumerated list of stored definitions.

- Click on the *drop down arrow* to show the list of stored definitions and select **Traceability**
- Click OK.

CStored Defin	nitions		
Definitions:	Traceability	-	1 ∟ ≞~
	,		
	Save	Delete	

Viewing a Traceability Hierarchy Diagram

Your diagram should look similar to the one shown below. Since the **Collection Management System** document was the source from which we extracted the specific requirements, we really have traceability from the source *Document* through the third tier (child-level) *Originating Requirements*.



Notice that **ORD.1.7 Monitor and Assess** Originating Requirement can be broken down further since it combines *monitor and assess*. So let's add another level of Originating Requirements by adding *targets* for the *incorporates* relationship of **ORD.1.7 Monitor and Assess**. We will add these elements directly from the **Hierarchy** diagram.



Adding Elements in a Traceability Hierarchy

CORE allows us to create elements from a diagram. Using the **Traceability Hierarchy** diagram, we will create two additional originating requirements incorporated in **ORD.1.7** to break down the requirement to its lowest possible level.

- From the **Traceability Hierarchy** diagram, double-click the **ORD.1.7 Monitor and Assess** object to open a *Text View* of this element.
- In the **Text View** window doubleclick the <u>incorporates</u> relationship to open the **Target Dialog**
- In the Target Dialog, select Originating Requirement from the list of *Target Classes*.

III Monitor and Ass	ess asText (Trial Version)	
<u>File E</u> dit <u>E</u> lement	Target Views Options	
🖪 🎒 🛄 🖪	〃 亚 △ - 凿 凿 凿 凿 🖆 🕶 🖡 、	•
Name:	Monitor and Assess	
Number:	ORD.1.7	
Abbreviation:		Ĥ
Description:	7. The system shall monitor and assess its own performance.	
Weight Factor:		≣ª
Paragraph Title:		
D	ii	臣
Relationships	Targets & Attributes	die 1
documented by generates		r (j) r
incorporated in		屈
🗅 owned by		强
verified by	▼ Sort: Numeric by class ▼	* -
RWDA Friday, De	cember 01, 2000 at 10:19:34 AM 10:32 /	AM //

Instead of selecting a target from the list of Possible Targets, we will create two new ones.

- Click NEW.
- Type a name for the target. We will name one **Assess Self Performance.** Click **OK** and the new target is listed in the *Targets* pane.
- From the **Target Dialog**, click the **NEW** button again to create another target.

New Origina	tingRequirement	×
Enter the nam	e for the new element.	
Assess Self F	Performance	
<u></u>		
OK	Cancel	

- We will name this one **Monitor Self Performance.** Click **OK** and the new target is listed in the *Targets* pane.
- Click **OK** to close the **Target Dialog**.

New OriginatingRequirement	x
Enter the name for the new element.	
Monitor Self Performance	
OK Cancel	

Viewing a Traceability Hierarchy

By default, hierarchy diagrams display 3 levels. In order to see these added elements in our **Traceability Hierarchy** diagram we need to change the diagram to display 4 levels.

Hierarchy Diagram Options

Options Icon Scale

✓ Store View Settings

M Show Frame

- From the Hierarchy window, select Options > Local Diagram Options.
- Type **4** in the Levels field of the **Options** tabbed view of the **Hierarchy Diagram Options** dialog.

Notice these new elements are not numbered.

- Right-click on the Monitor and Access element and select **Renumber Selection**.
- ORD.1.7 appears by default; click OK.
- Click **OK** when prompted for a sort block type.
- ✓ Use Compact Placement

 Levels:
 ■

 Renumber Element and Descendents
 ×

 Enter the new number for the selected element.
 ●

 ORD:1.7
 ●

 OK
 Cancel

 OE
 Select Sort Block

 Select the sort block to use in sorting children for the renumber operation.
 ▼

<u>0</u>K

Cancel

Font Line

<u>A</u> -

🗾 - 🖄 -

Fill

CORE®

<u>0</u>K

<u>C</u>ancel

• Click **Yes** when the warning appears.



Your diagram should look similar to the one shown below.



Enhancing the System Definition with Issues

Now that we have extracted the *Originating Requirements*, the requirement analysis begins. As a System Engineer, you want to identify problems encountered during system engineering such as poorly stated or conflicting requirements. In **CORE**, these problems can be captured as *Issues*. An *Issue* identifies a problem (as well as a resolution) with an element in the system design or specification. The primary application is documenting problems with requirements.

Let's add an *Issue* to ask for clarification on our Originating Requirement.

- Close any open windows, except for the **Database Browser** and **CORE Control Panel**.
- From the **Database Browser**, double-click **Issue** in the *Classes* pane to create an *Issue* element.
- Name the Issue Media of Requests and press OK to close the dialog.
 - From the **Database Browser**, double-click the new **Media of Requests** element to open a Text View window.

We link an *Issue* to the element that generated the problem via the <u>generated by</u> relationship.

- Double-click <u>generated by</u> in the *Relationships* pane.
- Add the Originating Requirement, ORD.1.1, Accept Requests as the target.
- Click **OK**.

•

New Issue		×
Enter the nam	e for the new element.	
Media of Red	quests	_
·		
OK	Cancel	

謳 Media of Reque	sts asText (Trial Version)	_ 🗆 ×
<u>F</u> ile <u>E</u> dit <u>E</u> lement	<u>T</u> arget ⊻iews <u>O</u> ptions	
🗄 🎒 🛄 🖪 🛛	/ U 🛆 - 📥 🗗 🏕 🖨 🖆 🖻 :	×a .
Name:	Media of Requests	
Number:		
Abbreviation:		Ē
	<u>k</u>	
Description:	-	
Severity:	Critical	
Status:	Open	
		크그 (원)
Relationships	Targets & Attributes	- fr
documented by		-Fi-
generates	- I	100
cowned by		5741
verified by		
	Sort: Numeric by class	▼ * ▼
RWDA Friday, De	cember 01, 2000 at 10:56:14 AM 1	0:56 AM //

"generated by" for Media of Requests (Trial Version)		
Target Classes	Targets	Add
Item 🔺	All Elements 💽 🔽	
Link	OriginatingRequirement_ORD.1.1 Ac	New
OriginatingRequirement (10/10)		Remove
Possible Targets		<u>0</u> K
All Elements		
ORD.1 GENERAL REQUIREMEN		
ORD.1.1 Accept Requests		
ORD 1.3 Control Multiple Sensor		
ORD 1.4 Maximum Staff		
Sort: Numeric	Sort: Numeric by class	
Enhancing the System Definition: Issues (cont.)

📲 Media of Requests asERA (Trial Version)

CORE allows you to capture both the issue and the decisions that resolve the issue. You can document your decision, your alternatives, and your rationale. In this way, **CORE** serves as a repository for the project design history- capturing the why of system engineering decisions. We will complete the **Media of Requests** Issue element to capture the analysis of the problem and its resolution. Although you can use the **Text View** to fill in the attributes, let's use and **ERA View** this time. With an **ERA View**, you can see a diagram of the *relationships* and a **Text View** of the *attributes* of the selected element.

- From the Database Browser, select the Media of Requests Issue element.
- Click the **ERA** button to open an **ERA View** of the selected element.

ERA Diagram

- <u>File Edit Diagram Extras Layout Views Options</u> 🖪 🎒 🛄 B Z 😐 🛕 • 🖆 🖆 🞜 👂 🛊 <u>A</u> • <u>J</u> • 🕭 • Originating Requirement ORD.1.1 states that the system shall -말 accept intelligence data collection requests from the certified users. Description: What are the media that the system must be able to accomodate? Ĥ -• Critical Severity: 围 ORD.1.1 Accept Requests • Open Status: ۳ OriginatingReq.. **•** None Issumptions Å Trial User -Engineer 1)Hardcopy forms; 2)Verbal; 3)Phone-verbal; 4)Phone-electronic 🔺 file; 5) PC diskette-electronic file; 6) all of the above. Alternatives Date Author ridau. December 01.20. Trial User -Vumber e: (Trial) Media of Requests CH The system shall accept requests in any of the following formats: 1) 📕 Hardcopy forms, 2) Verbal, 3) Phone-verbal, 4) Phone-electronic Decision: file, or 5) PC diskette-electronic file. -Issue × -RWDA Friday, December 01, 2000 at 11:16:18 AM CCC 11:16 AM
 - Type text in the *Description* and *Alternatives* fields. You can use the text as shown above, or come up with your own.

Scroll through the list of Issue attributes to see the other fields applicable to an Issue. As a System Engineer, it is important to be as thorough as possible. **CORE** provides the fields you will need to maintain complete requirements.

• Close all **CORE** windows except the **CORE Control Panel**.



_ 🗆 ×



This Page Intentionally Left Blank.





Defining the System

In this section, you define your system

- Create System Elements
- Define System Boundaries
- View Physical Hierarchy
- Define High Level Functions

Defining the System and its Boundaries

For defining the system boundary, we want to identify the top-level components (physical elements) their top-level (root) functions and any top-level inputs and outputs. To begin, we need to define the overall system context to determine what is inside and outside of our system. The *System* element is intended to identify the system and capture the system-level mission, I/O, functions, performance, and components. We will create the System element with the **ELEMENT EXTRACTOR** because we can get the text from the *CollMgt.txt* file and save some typing.

Open an Extractor Window:

- From the CORE Control Panel, click the ELEMENT EXTRACTOR button.
- Navigate to and load Samples/Collmgt.txt and double-click (or click OPEN.)
- From the *Select Element Class* dialog, select the **System** class as the destination class.

Define the attributes for the Element:

• Fill in the Name, Number (SYS.1), Description, and Mission fields.

Next, we define a relationship by establishing that the System Element is <u>documented by</u> the Document Element, which we had named *Collection Management System*.

Establish a System Relationship:

- Double-click the documented by relationship to open the Target Dialog
- Select **Document** from the *Target Classes* pane.
- Double-click **Collection Management System** from the list of *Possible Targets* to add it to the *Targets* pane.
- Click **OK** to close the Target Dialog.

Save the Element Attributes in the Database:

• From the **Element Extractor** window, click the **CREATE** button to store this *System Element* definition.

Close the Element Extractor window:

• From the Element Extractor window, click the X, or choose File > Close Window to close the Element Extractor window.

Element Extractor (Trial Version)					
Eile Edit Extractor Target Views Uptions					
Collingt.txt Element Definition					
SYSTEM	Class: System				
	Folder: System		고 🚡		
1.0 SCOPE	Name:	COLLECTION MANAGEMENT SYSTEM			
establishes the	Number:	SYS.1			
performance, design,	Abbreviation:		(F		
development, and test requirements for an intelligence collection management system. The mission of the	Description:	This Collection Management System is intended to serve as a means to demonstrate the use of automated system engineering support tools. As defined, this demonstration system accepts requests for intelligence data, determines the best way for the the system to respond to the request, and then provides the requested data to the requestor, if possible. In the process of acquiring the requested data, the system may generate tasking orders for a set of intelligence data coll ectors, which may include humans.			
Collection Management System is to provide management of a system of intelligence collectors, from the acceptance of user remests through	Mission:	The mission of the Collection Management System is to provide management of a system of intelligence collectors, from the acceptance of user requests, through scheduling the collectors, to delivery of the collection products to the users.			
scheduling the	Cost:	☐ □ve	rride		
collectors, to delivery of the collection products to the users.	Relationship contained by documented by	Targets & Attributes			
2.0	▲	Sort: Numeric by class	•		
This element has not been created.		្រា	:53 PM 🥢		

Creating External Systems

External systems interact with the System element. External systems are represented in the system description database by the *Component* class. A *Component* is an abstract term that represents the physical or logical element that performs a specific function or functions. We will create two physical Component class elements, *Collectors* and *Customers*. We will also create a logical Component element called *Universe* to provide a context as to how our system interacts with other systems to achieve its objective.

We will use the **Database Browser** to create these elements since this data does not come from the source document--e.g.; there is no data to get from the **Element Extractor**.

Create three new Component Elements:

- From the CORE Control Panel, open a Database Browser window.
- Double-click **Component** from the list of *Classes* to create a *Component* element. The *New Component* dialog prompts for the element name.
- Name the element **Collectors** and press the Enter key (or click OK.)
- Double-click **Component** again to create an element named **Customers**.

New Component	×
Enter the name for the new element.	
Collectors	

• Double-click **Component** again to create an element named **Universe**.

Database Browser for System	Engineering (Trial Version)	
Ele Fojder Element views option	a a a a	
Classes	Elements	
System Engineering	All Elements 💽 🍸	
Category	Collectors	
CompletionCriterion	Customers	
🛁 📥 Component (3/3)	Onwerse	
Constraint		FE
Document (1/1)		
DomainSet		
		产
		厨
□ Issue (1/1)		_Fi_
- G Item		-1-1-
🛛 📖 🗂 Link 👘 📶		바
📖 🗅 OriginatingRequireme		15
🗅 PerformanceIndex		龝
Resource		
	Sort: Numeric	* •
	02:09	РМ //

Note

Pressing the Insert key also allows the creation of a new element.

Defining the Universe Component

Now we need to define the attributes and relationships of these elements. We capture the external systems as Components with *ComponentType* set to *External System*. Hierarchy of Components can be constructed using the <u>built from</u> relationship. We want to show that the *Universe* is <u>built from</u> the *Collection Management System* and the two external systems, *Collectors* and *Customers*.

Define Attributes and Relationships:

- Open a Text View by double-clicking Universe.
- Type the attribute information as shown below.

🔡 Universe asTex	t (Trial Version)	. 🗆 ×
<u>F</u> ile <u>E</u> dit <u>E</u> lement	<u>I</u> arget <u>V</u> iews <u>O</u> ptions	
🖪 🖨 🛄 🖪	〃 亚 △ - 🗃 者 者 🖀 📲 📲 🖕	
Name:	Universe .	▲ 💷
Number:	e.0	
Abbreviation:		Ĥ
Description:	The universe is a supersystem to show the relationships between our system (Collection Management System) and the external systems (customers and collectors).	
Component Type:	External System	
Cost:	🗖 Override	
Creator:	Trial User	F
Created:	Friday, December 01, 2000 at 02:09:21 PM	• ^{-[]} *
Relationship	s Targets & Attributes	_ = ^[] =
annotated by assigned to augmented by	Sort: Numeric by class	₩ ₩ • • • •
RWDA Friday, D	ecember 01, 2000 at 02:11:07 PM 02:1	зрм

Adding Detail to the External Systems

(Continuing)

To be a thorough System Engineer, you will want to complete the attribute fields for the Collectors and Customers Component Elements.

- From the **Database Browser**, open a **Text View** for each of these Component Elements
- Fill in values for the *Number* and *Description*, and set the Component Type to **External System** from the pull-down list.
- Close all the **Text View** windows when you are done.

🔡 Customers asTex	rt (Trial Version)	×
<u>File E</u> dit <u>E</u> lement	Target ⊻iews Options	
🖪 🎒 🛄 В .		
Name:	Customers 🗕	
Number:	e.1	
Abbreviation:		ĩ
Description:		
		12
Component Type:	External System	
Cost:	🗖 Override	<u>-</u>
· · · · · · ·	Trial Jack	묍
Relationships	Targets & Attributes	in i
built from	Component e.0 Universe	i.
categorized by		л- -
causes	12	24
connected thru		봂
connected to	Sort: Numeric by class	Ţ
RWDA Friday, De	cember 01, 2000 at 02:20:15 PM 02:23 PM	

Collectors asTer	xt (Trial Version)	⊐×		
<u>File Edit Element Iarget Views Options</u>				
🖪 🎒 🛄 🖪	I ∐ ▲ - 凿 凿 凿 凿 = = = -			
Name:	Collectors			
Number:	e.2			
Abbreviation:		Ĥ		
Description:	The external system, collectors, is responsible for acquiring intelligence data that is requested but does not reside in the inventory. There are multiple collectors, of different types, which may include humans.	100 C		
Component Type:	External System	± ∎'a		
Cost:	🔽 Override			
C	Tviattlaav	191		
Relationships	s Targets & Attributes	die		
built from built in categorized by causes connected thru connected to	Component e.0 Universe	。 中 国		
•	Sort: Numeric by class	× -		
RWDA Friday, De	ecember 01, 2000 at 02:20:16 PM 02:21 F	м		

• CORE®

Viewing a Physical Hierarchy

To see the structure of the Universe Component Element, we can view a Physical Hierarchy diagram.

- From the Database Browser, highlight the Component Universe element.
- Click the **Hierarchy** button.
- Select **Physical** from the enumerated list of stored definitions.
- Click **OK**.



You will get a Physical Hierarchy diagram representing the context of our system.

- Choose **Options > Show Relationships** to annotate the diagram.
- Close all open windows, except for the CORE Control Panel.



Creating Function Elements

We have defined our *Document*, our *Originating Requirements*, our *System*, and some *Components*. Now we need to add some *Functions* to describe what our system is to do. A *Function* is a transformation that accepts one or more inputs and transforms them into outputs. Let us create the root (top-level) function for our system and each of the external systems. With any system, a root-level function drives the other functions. This time we will use the **Database Editor** to create these elements. The **Database Editor** allows us to create elements and define the attributes in the same window. It is good practice to fill in the applicable attributes of elements as you create them, especially the *Description* since it represents the requirement statement for the element.

To Create Function Elements:

- From the CORE Control Panel, click the DATABASE EDITOR button.
- In the **Classes** pane, double-click **Function** to open the *New Function* dialog.
- Type the name Perform Collection Management
- Type a description for this function in the *Description* field.



Create three additional Functions named as follows:

- Context (Root Function For Universe) number u.1,
- Perform Collector Functions number c.1, and
- Perform Customer Functions number c.2.

Create Root Functions for the System/External Systems

We will now allocate the root functions to the components that perform them. This means establishing an *allocated to* relationship.

Establish an allocated to Relationship:

- From the Database Editor highlight the Function Element Context (Root Function For Universe).
- Double-click the relationship, <u>allocated to</u> to open a Target Dialog.
- Highlight the target class *Component*, and then double-click the **Possible Target**, **E.0 Universe** to add it to the *Targets* pane.
- Click the **OK** button to close the **Target Dialog**.
- Double click on Behavior Type: Atomic.
- From the Edit Behavior Type Attribute dialog, select Integrated (Root) from the enumerated list as the new *Behavior Type* and click **OK**.

You are now finished allocating the Context function.

"allocated to" for Context (Root Function For Universe) (Trial Version)				
Target Classes	Targets	Add		
💬 🖆 Component (3/3)	All Elements			
📖 🖆 System (1/1)	Component e.0 Universe	New		
		Remove		
Possible Targets		<u>0</u> K		
All Elements 🗾 🔽				
e.0 Universe				
e.1 Customers				
e.2 Collectors				
	1			
Sort: Numeric	Sort: Numeric by class			



Allocate the other two external root functions (**Perform Collector Functions** and **Perform Customer Functions**) to their respective Component elements the same way.

Also, allocate **Perform Collection Management** to the System element and identify its *Behavior Type* as *Integrated (Root)*.

Note

Make sure the model has a System and a corresponding root Function (behavior type Integrated Root) for the system being developed. Even during requirements analysis, this is important as it provides a standard starting point for analysis and review.

After finishing, close the Database Editor.

Building a Functional Model

Next, we begin the behavior analysis of our system. We can do this graphically with a functional model of our context function, **Context(Root Function For Universe)**.

Open Database Browser



CORE®

Open an FFBD:

- Highlight the Function Element, Context (Root Function For Universe).
- Click the **FFBD** icon to open a functional flow block diagram of the selected element.
- Select/highlight the branch between the two reference blocks. This is where we will insert the functional behavior for the context function, the root function for Universe.



A Note About Insertion Points and Selecting Objects and Branches



Inserting a Parallel Structure

Recall, we have three functions that we want to include in our diagram:

- Perform Collection Management,
- Perform Collector Functions, and
- Perform Customer Functions.

Each of these functions works in parallel. To incorporate these functions into our diagram, we will create parallel branches.

Creating a Parallel Diagram Structure:

From the FFBD window, choose Insert > Parallel menu command to insert a concurrency structure.
 Context (Root Function For Universe) asFFBD (Trial Version)



CORE prompts for the number of branches. We will create a concurrency with 3 branches since we have 3 functions.

• Type 3, then OK

Now we insert one function on each branch. To insert an element on a branch, you click on the branch to highlight it, and you select the menu commands **Diagram > Insert > Element**.

Adding Functions to a Diagram:

- Highlight the top branch.
- In the FFBD window, choose Insert > Element command to open an Insert Element dialog.

Insert Element (Trial Version)				
Classes	Selections	Add		
🖿 Function (4/4)	*			
		New		
		<u>B</u> emove		
Elements		<u>0</u> K		
All Elements		<u>C</u> ancel		
	-			
Sort: Numeric	T E			

Building a Functional Model

The **Insert Element** dialog works like the Target Dialog; it shows elements that can be inserted. You can select one of those to add to the Selections pane, or you can create a new element by clicking the **NEW** button. Click **OK** and all the elements listed in the Selections pane are added to the highlighted spot in the diagram, in the order they are listed in the Selections pane.

Inserting Elements:

- From the list of *Classes*, highlight **Function**
- From the list of *Elements*, highlight **Perform Collection** Management.
- Click **ADD** to add the element to the *Selections* pane.
- Click **OK**.

Classes	Selections	Add
Function (4/4)	Function Perform Collection Management	New
		<u>R</u> emove
Elements		<u>0</u> K
All Elements Perform Collection Management .1 Perform Collector Functions .2 Perform Customer Functions J.1 Context (Root Function For Unive		<u>C</u> ancel
↓		

The Function Element is added to our FFBD.

In the same manner, insert **Perform Customer Functions** on the second branch and **Perform Collector Functions** on the third branch.



Adding Inputs and Outputs

Now let's add inputs and outputs to the root function (and thereby to the Components that they are <u>allocated to</u>) to show the activities of the Functions. We will do this by using the **N2** (interface) view. In a later exercise, we will show how to add data flows from the **EFFBD** view.

Open an N2 Diagram:

- Click on the diagram background to ensure no icons are selected. We don't want any icons or branches selected.
- Choose Views > N2 to open an N2 Diagram of this function, Context (Root Function For Universe).



An **N2** *Diagram* represents the data flow for a system or system segment. It displays the data dimension of the behavior model, whereas the **FFBD** displays the control dimensions of the integrated behavior model. Used in conjunction with an **FFBD**, you can use the **N2** diagram to help capture and analyze the functional behavior of a system.

Note
Editing the diagrams is another way of entering data into the CORE repository.

Adding Inputs and Outputs (cont.)

Item Elements represent flows with and between Functions Elements. An *Item* is an input to or an output from a Function. We will create an Item Element named **Tasking**. Tasking will be an *output from* **Perform Collection Management** and an *input to* **Perform Collector Functions**.

Opening a Connection Dialog:

- On the N2 Diagram, click to select the icon Perform Collection Management.
- Now, while holding down the shift-key click to select the **Perform Collector Functions** icon.
- Right-click to open a pop-up menu and choose the **Connect via Data**... command to open a **Connection Dialog**.

EIE Diagram Extras Layout Views Option	sN2 (Trial Version) s		
R 🖨 🛄 🖅 🗄 🖙 🌭 💲 🔎	🗄 🗇 <u>A</u> • 📶 • 🕹	> - Show External Item	s 🗸
			Ĥ
			-15
0			ff
Perform			基
Management			151
			-Cj-
Perform	<u>R</u> ename Element		- E-
Collector	Delete Selections		153
	Insert <u>N</u> ode		醖
	Edit Inputs		1 €
	Edit <u>I</u> riggers Edit Outouts		_
	Connect via Data		
Date: Au	Connect via Trigger		
Monday, December 04, 2000	Layout		
	<u>⊻</u> iews	n For Universe)	
RWDA Monday, December 04, 2000 at 09:24	4:53 AM	09:29	AM //



Copyright[©] 1993-2000 Vitech Corporation. All rights reserved.

The N2 Chart Shows the Interfaces for Our System

We'll now add three additional data items in the same manner.

Using the chart as a guide for the desired information flow between our system and the two external systems, complete the model for the three other items: *product status, requests,* and *data.*

Context (Root Function For Universe) Interfaces				
Source (Output)	Input	Item		
Perform Collection Management	Perform Collector Functions	tasking		
Perform Collection Management	Perform Customer Function	product status		
Perform Customer Functions	Perform Collection Management	requests		
Perform Collector Functions	Perform Collection Management	data		

Below is the completed N2 view showing the interfaces (Item flows) between the three functions.



When you are done, close the diagram window.

Now we will decompose the Functions, breaking them down into leaf-level Functions. We begin with the Function **Perform Collection Management**.

- From the Database Browser, select the Function Element Perform Collection Management.
- Click FFBD to open an FFBD view.
- Select/highlight the branch between the reference blocks at the insertion point.
- From the menus, choose Insert > Element.
- From the Insert Element dialog, create two new Functions: Accept and Format Request and Acquire Products.
- Click **OK** to add these functions to the diagram serially in the order they were added to the target list.

• ()• Pe	Perform Collection Management asFFBD (Trial Version)	_ 🗆 🗙
<u>F</u> ile	<u>E</u> dit <u>D</u> iagram Insert E <u>x</u> tras Layout <u>V</u> iews <u>O</u> ptions	
8		
±‡		H
E	Accent and	
ţ,	Ref. Accept and Acquire Ref. Ref.	
ជ្រ		
ι Γ Ι	Date: Author:	æ
RP-	Monday, December 04, 2000 Trial User	151
Ð	Number: Name: 0 (Trial) Perform Collection Management	
⊏¢,		r [j]r
•		15
		▶ _{* *}
		49 AM 💋

Now while the branch is still selected, from the Diagram menu choose the **Insert > Parallel** command and accept the default of **2** branches. Then, select/highlight the top branch of the concurrency (the parallel construct). From the Diagram menu, select the **Insert > Element** command.

When CORE opens the dialog box, create two functions: Retrieve Product From Inventory and Provide Products to Customer. Click the OK button.

Insert Element (Trial Version)				
Classes	Selections	Add		
🖴 Function (8/8)	Function Retrieve Product From Inventory			
	Function Provide Products to Customer	New		
		<u>R</u> emove		
Elements		<u>U</u> K		
All Elements V		Connect		
0 Perform Collection Managemer		Lancei		
c.1 Perform Collector Functions				
u 1 Context (Root Function For L)				
Agent and Format Requests				
Sort: Numeric				

Select/highlight the bottom branch of the concurrency. From the Diagram menu, select the **Insert** > **Element** command.

When **CORE** opens the dialog box, create one function: **Compare Product To Request**. Click the **OK** button.



To define this function as having multiple exit conditions (paths), select/highlight the function Compare Product To Request. Select the Insert >> Add Exit Condition command from the Diagram menu.

• When **CORE** opens the dialog box, select **Completion Criteria** from the *Classes* pane, then click on the **New** button and create two exit conditions (represented as *Completion Criteria*): **OK** and **Discrepancies**.



• Click the **OK** button when finished.





From the **Diagram** menu, select the **Insert > Element** command. When **CORE** opens the dialog box, click on the **New** button and create one Function: **Generate Discrepancy and Recommendations Report**. Click the **OK** button.

Insert Element (Trial Version)				
Classes	Selections	<u>A</u> dd		
🖴 Function (10/10)	Function Generate Discrepancy and Recommendations Report			
		<u>N</u> ew		
		<u>R</u> emove		
Elements		<u>0</u> K		
		<u>C</u> ancel		

Next, select/highlight the Function Acquire Product. From the Diagram menu, select the Insert > Iterate command. When CORE opens the dialog box, click on the New button and create the *DomainSet* for all products in request. Click the OK button.



- Ensure that the function **Acquire Product** is selected/highlighted.
- From the Edit pull-down menu, select the Cut command (or use the Microsoft Windows shortcut Ctrl+X).

CORE®

Highlight the main branch inside the Iterate. From the **Edit** menu, select the **Paste** command. This logic assumes that a single request may ask for more than one product.



CORE®

As a final step, let us renumber the functions on our **FFBD**. **Be sure that nothing is selected/highlighted**. From the **Diagram** menu, select the **Renumber Element** command. When **CORE** prompts, enter 0 (zero) as the number for the parent function. Click **YES** at the next warning prompt. This will cause the functions on the diagram to be numbered with integers.





Adding Inputs and Outputs (N2 Charts)

Remembering the N2 Charts:

While we have the **FFBD** open let's define the Items which are inputs and outputs to the functions as shown in the **N2** chart and the table of the **Perform Collection Management** root function. We will do this using an **N2** chart, as we did earlier for the **Context (Root Function for Universe)** function. From the **Views** menu, select the **N2** command.



Selecting a single function allows you to define inputs, outputs, and triggers. Selecting multiple functions allows you to connect the function via data or triggers. **Remember to hold down the shift key when selecting from-to connections.**

Deriving the Enhanced Functional (Behavior) Model

Enhanced FFBD:

Close all windows but the Palette and reopen the Database Browser window. Now, let's decompose Function 2, Acquire Products, to add another level of detail. For this exercise, we will use the EFFBD (an FFBD with data).

• Select/highlight the Function Acquire Products.

From the Views menu, select the Enhanced FFBD command to open an EFFBD of Acquire Products. From the Database Browser you can select the function and click the EFFBD button.





Highlight the main branch (between the reference block on the left and the concurrency (AND) for the reference blocks on the right).

Generate the functions and their control constructs (e.g., the FFBD subset of the EFFBD) as shown, using the methods presented earlier for decomposing the Perform Collection Management function.



Deriving the Enhanced Functional (Behavior) Model (cont.)

Inputs and outputs can be added to functions in the EFFBD in the same manner as we used with the N2 chart earlier. Flows between functions can be defined by highlighting the functions in order of from-to and using the Edit > Connect via ... commands in the Diagram menu.

Individual function inputs and outputs can be added with the **Edit > Inputs** (or **Outputs**, or **Triggers**) command in the **Diagram** menu. Note that triggers differ from data items in that they are required by a Function before it can begin execution.



The completed EFFBD and N2 chart are shown.

Function	Trigger(s)	Input(s)	Output(s)
Check product Inventory		formatted request	collector request
Prioritize Request	collector request		collector tasking
Accept Data From Collectors		collector data	process data
Process Data and Put in Inventory		process data	inventory product



CURER

Revisiting/Extending Traceability

Traceability:

Now that we have defined our functional (behavior) model, let us go back and extend the traceability to include these new elements by simply relating these functions to the requirements they fulfill. We could form these relationships in the **Text** views, the **ER** views, the **ERA** views, or the **Database Editor**.

In the Database Editor, in consecutive panes, select class Originating Requirement, element Accept Requests, and relationship traces to. Then select the Add Target command from the *Target* menu or click the Edit Targets button.

In the Target dialog box, highlight target class Function. Select possible target Accept and Format Requests and click the Add button. Click the Done button to close the dialog box.

🔡 Database Editor for System E	ngineering (Trial Version)		Ι×	
<u>File Edit</u> Folder <u>Element Target Views(Element)</u> Vie <u>w</u> s(Target) <u>O</u> ptions				
🗄 🖨 🛄 в и ц 🛆 -	- 😂 🎭 🖬 🖆 🗂 🖆 🗂	e 12 -		
Classes	Elements	Relationships		
System Engineering	All Elements	incorporates		
Document (1/1)	ORD.1 GENERAL REQUIREME	owned by		
DomainSet (2/2)	ORD.1.1 Accept Requests	verified by		
Function (14/14)	ORD.1.3 Control Multiple Sensc	_	围	
Glossary	ORD.1.4 Maximum Staff	Targets & Attributes	- E	
Interface	ORD.1.5 Provide Feedback	Function 1 Accept and Format Reques		
Item (15(15)	ORD.1.7 Monitor and Assess			
Link	ORD.1.7.1 Assess Self Perform 🚽		<u>a</u>	
OriginatingRequirement (10/			臣	
	Sort: Numeric	Sort: Numeric by class	e[]e	
	,,	,	-Ei-	
Name: Accept Requests				
Number ODD 1.1	Loot Modified	Mondou December 04, 2000 et 10:41:01	192	
		Monday, December 04, 2000 at 10.41.01	饇	
Description - 1. The syste	m shall accept intelligence data c	ollection requests from the certified 🛛 🛋		
users.			-	
RWDA Monday, December 04, 2	2000 at 10:41:01 AM	10:41 A	м //	

Go to the **Database Editor** again, from the **Views (Element)** pull-down menu, select **Hierarchy** and choose **Traceability** from Definitions. The results should be as shown.



Extending Traceability

Continue by extending the traceability for the issue **Media of Requests**. In consecutive panes in the **Database Editor**, select class Issue, element **Media of Requests**, and relationships traces to. In the **Target** dialog box, highlight target class Function. Select possible target **Accept and Format Request** and click the **Add** button. Click the **Done** button to close dialog.

University of the second secon	Engineering (Trial Version)	-	□×
	, v 😂 🗞 🖬 🏙 🗂 🏠 🗃		
Classes	Elements	Relationships	
System Engineering	All Elements	generates	
Function (14/14)	CI.1 Media of Requests	Contractions to	
Glossary		verified by	
Interface		_	
Issue (1/1)	1	Targets & Attributes	-6
Item (15/15)		- Function 1 Accept and Format Requests	
LINK OriginatingRequirement (10)			
PerformanceIndex	J		- a
Resource		() () () () () () () () () ()	臣
	Sort: Numeric	Sort: Numeric by class	- (j:
			-
Name: Media of Requests			
Number VIII	Last Modified	Monday December 04, 2000 at 10:45:54	1 521
Description 🔄 Originating	g Requirement ORD.1.1 states that	t the system shall accept intelligence 🛛 🔺	1
data collec	tion requests from the certified us	ers. What are the media that the	
system mu	st be able to accomodate?	•	
			1
RWDA Monday, December 04	, 2000 at 10:45:54 AM	10:47	AM

Now in the **Database Editor**, again select the class **Originating Requirement** and the element **Accept Request**, then from the **Views (Element)** pull-down menu, select **Hierarchy** and choose **Traceability** from the show list button.

You see the traceability of the **Originating Requirement** down to the Function level.

NOTE

If any icons in a hierarchy diagram have a black dot in the upper-left corner, that icon can be further expanded. From the Settings menu, select the Diagram Options command and set the number of levels to a number greater than 3, which is the global default. A black dot in the upper-right hand corner of an icon means that icon is repeated somewhere else on that diagram. To see where else it appears, highlight it, and select the Highlight Matching Nodes command in the Diagram menu.



Extending Traceability (cont.)

Use the same technique to extend traceability for the *Originating Requirements* Retain Inventory, Control Multiple Sensors, Monitor and Assess and Prioritize Requests. Use the diagrams below to guide you. While creating the Monitor and Assess diagram notice the function Compare Product to Request is traced from Assess Self Performance and the function Generate Discrepancy and Recommendations is traced from Monitor Self Performance.

The Originating Requirements Maximum Staff and Provide Feedback do not trace to any Functions in our model at this time.







Allocating the Functions

Allocating the Functions	
Let us use the Database Editor to establish the relationship that these (leaf-level) components perform the (leaf-level) functions as shown in the two <i>custom</i> hierarchies below (See page 58). (<u>performs</u> is the inverse of the <u>allocated to</u> relation.)	Image: Analyst/Command Center Name: Analyst/Command Center <t< td=""></t<>
Hierarchy Definition Dialog Stored Definitions Definitions Definitions Save Delete Less << Properties Labet Save Delete Less << Properties Labet Save Delete Less << Properties Labet Selections Add Remove Provides definitions for Provides definitions Provides definitions Provides definitions Provides definitions Candidates Candidates Selections Add Remove Candidates Add Remove Candidates Add Remove Candidates Candi	Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of the system Image: speed of
performs performs 3 5 Retrieve Compare Product From Product to Re Function Function	C.2 Analyst/ Workstation Component performs performs performs performs Accept Data Check Product Inventory Prioritize Request Process data and Put in Inv Function Function Function Function Date: Monday, December 04, 2000 Author: Trial User Trial User Number: c.2 (Trial) Analyst/Workstation



Impact Analysis

ERA diagram:

To show some of the power of our **ERA** design repository, let us postulate a typical system engineering example.

Assume that the customer wants to know the impact of exchanging with the workstation in the **Analyst/Workstation** component.

Select the component **Analyst/Workstation** and create a custom hierarchy with the <u>performs</u>, <u>inputs</u>, and <u>outputs</u> relations and the targets, Functions and Items.

Hint: With the cursor highlighting any item in the candidate list, pressing the first letter of the desired candidate jumps to the first candidate beginning with the entered letter; example – pressing P within the candidate list jumps to performs

Hierarchy Definition Dialog				
Stored Definitions Definitions: Save	Delete	<u>O</u> K <u>C</u> ancel Less <<		
Properties Label:	Show Relationships	Essential		
Relations Candidates member of output from outputs owned by owns performs produced by	Selections inputs outputs performs	Add Remove		
Target Classes Candidates ImplementationUnit InformationUnit Interface Issue Leader Link MureberdElement	Selections Function Item	Add		

The diagram quickly shows which functions and which I/O (inputs and outputs) may be impacted.



Capability of CORE

CORE Provides Full Traceability from Source Document to Physical Architecture

The Hierarchy diagram below displays the document, Collection Management System. In the Database Browser select the class Document then select ORD.1 Information Management System. Click on the Hierarchy button and choose the stored definition Traceability. In the diagram, go to the Settings Menu > Diagram Options to set the level to seven and the scale to fit the window.



Using relationships for traceability makes it easy to detect unfulfilled requirements and unresolved issues. For example, Originating Requirements, **ORD.1.4** and **ORD.1.5** do not trace to functions on design elements.

Note

The Unallocated Leaf-level Requirements script may also be executed to produce an automated indication of which requirements do not trace to anything.



Generating a Report

An added benefit of **CORE** is its report writing capability. Reports in **CORE** can range from a simple database query (e.g. show me a list of all open issues) to complex, formatted reports (e.g. the System Description Document). **CORE** contains a number of built-in reports that can be executed directly in **CORE**. Also included in **CORE** is the **COREscript** language providing the capability to develop custom reports, queries, and interfaces to other tools. The built-in reports are written in the **COREscript** language and were created to satisfy some common database requests. Reports in **CORE** can be generated in any ASCII-based text file format. Most reports are generated as a file in Rich Text Format (RTF) (a standard publication file format) that can be imported into word processors such as Microsoft Word for preview, editing, and publication purposes.

The report generator enables you to extract system specification material from the **CORE** database repository and present it in virtually any desirable format. Reports allow you to view the database in different ways. The structure of a report is controlled by a report script that instructs the report generator where to go in the database to get data and how to format the information for each section of the report.

Generating a Report:

• From the CORE Control Panel, select Scripts > Run Script

CORE Control Panel (Trial Version)	ersion)	_ 🗆 ×		
<u>File Database Schema Project</u>	Scripts Utilities Panel Optic	ons <u>W</u> indow		
🖪 🖨 🛄 🗉	<u>N</u> ew Script Ctrl+N			
	<u>O</u> pen Script Ctrl+0			
Project: Default 🔄 🔄	<u>R</u> un Script Ctrl+R abas	se 🔿 Schema		
General Diagrams DB Mgmt So	chema Utilities			
Database Eleme	nt			
Script Editor Run Script	Exit			
RWDA Database Mode		02:53 PM		

Generating a Report (cont.)

This opens a dialog prompt for selecting the desired report.

- Use the *pull-down arrow* to select **System Description Document**.
- Press **OK**.



A prompt will appear. Answer **YES** to running the report with a *top-level System/Component*. Next, you need to select the name of the **System** or **Component** in the repository for the report.

- Select System in the Classes pane
- Select Collection Management System in the Candidates pane
- Click **OK** button.

Select the Primary System/Component (Trial Version)				
Classes	Candidates	New		
💼 Component (5/5)	All Elements			
📖 🖆 System (1/1)	SYS.1 COLLECTION MANAGEMENT SYSTEM	<u>0</u> K.		
		<u>C</u> ancel		
	Sort: Numeric			

CORE®



Generating a Report (cont.)

When creating the System Description Document (SDD), you can select the desired sections/parts of the SDD that you would like included. Select/ highlight the sections that you want in your report and click Add. If you want the (entire) default document, click the Add All button. We will include all the sections.

• Click Add All.



• Click OK.

Select the Type of Diagram	s to be Output	×
Options	Selections	
Enhanced FFBD FFBD N2 Diagram IDEF0 Diagram	Enhanced FFBD FFBD N2 Diagram IDEF0 Diagram	<u>Add</u> Add All <u>B</u> emove <u>O</u> K <u>C</u> ancel

• Select Add All to include all diagram types.
Generating a Report (cont.)

CORE displays a default Report Settings dialog box. These are the settings as defined in the Report Preferences. Update the settings from the Preferences menu it you'd like to have some names and addresses stored for later use.

- Edit the settings, if des •
- Press OK. •

Once you select which report you' walks you through a series of pror what you like to query.

ired.	Report Settings 🛛 🗙
	Prepared by
	Name: ABU Corporation
	Address: 123 Main Street
d like. CORE	Aliywileie, 03A 30703
npts to establish	
	
	,
	Prepared for
	Name: XYZ Company
	Address: 321 Main Street
	v
	Publication Information
	Override Creator With: System Engineer
	Uverride Date with: Tuesday, September 19,
	OK
report name,	
t SSD.	
	1
	? ×
- 🖻 🖻	



Collection Management System SSD rtf

Press Save. •

Save in:

File <u>name</u>:

Save as type: All Files (*.*)

Select, or Name, Report Output File 🛅 Output

CORE generates the report as an RTF (Rich Text Format) file. Import the RTF file into Microsoft Word (or your favorite publication tool or word processor) for previewing and printing.

٠

<u>S</u>ave

Cancel

• CORE®

Generating a Report (cont.)

The document will be completely formatted except for the Table of Contents, List of Figures, and List of Tables.



Highlight the identifier (**Body of Table of Contents**) and press the **F9** *function key* on your keyboard for each identifier. Word will automatically format and paginate for each table/list.

The System Description Report for this example is approximately 40 pages long.

Generating a Report (cont.)

Another report available in **CORE** converts the System Engineering Database to HTML format and generates a Homepage for anyone with a Web browser to access.

- To run this report, select Scripts > Run Script
- From the Select Script pull-down select the HTML Report

Select Script		
Select the desired script. Liste the tria	d below is the subset of so I version of CORE.	cripts provided in
HTML Report		
	ОК	Cancel

Do you want to show blank attributes?

No.



- When prompted to **Select a location for the Report Home Page** to be saved, click **Save** to accept the default filename (**0_homepage.htm**) in the default directory (...**CORE 30 Trial\Output)**
- When prompted to *show blank attributes*, click **No**
- When prompted to *sort the elements*, click **Yes**
- When prompted to *select the diagram types*, click Add All
- Click **OK**



Yes

Information



When the report generation is complete, the **Execution Completed** window appears

• Click **OK**



Generating a Report (cont.)

• Navigate to \CORE 30 Trial\Output and double-click the file 0_homepage.htm

Your page should look similar to this image.

🖉 CORE Da	tabase	Homepage	e - Micro	osoft In	ternet Explorer				- D ×
File Edit	View	Favorites	Tools	Help					1
Back	Ŧ	→ Forward	~	区 Stop	💣 Refresh	කී Home	Q Search	😹 Favorites	»
🛛 Address 餐	D:\COR	E 30 Trial\O	utput\0_	homepag	ge.htm			💌 🤗 Go	Links >>
Selec	ct one o	CO of the follo (Note:	RE wing D Classe	Databas es witho	atabas e Class links t out hyperlinks	e Hoi o get a list o do not have	nepago of elements in a any elements.)	e particular class	
				A	ctivity nnotation				
				C	ategory				
				C	ompletionCrite	rion			
				C	omponent				
				C	onstraint				
				D	ocument				•
🥙 Done								🖳 My Computer	

Generating a Report (cont.) . 🗆 🛛 🙆 Ele ent Page - Mi Clicking on Function bring you to a list ORE® File Edit View Favorites Tools Help of all elements in the Function class. 수 Back \Rightarrow Stop ڑچ Refresh 쑵 tom () History -* Any function can then be selected to Address 🙋 D:\CORE 30 Trial\Output\35.htm ∂Go Links » • view its text view. • **Function: 2 Acquire Product** View eFFBD Page View FFBD Page View N2 Page View IDEF0 Page Created Friday, July 07, 1995 at 12:00:00 AM Graphical System Engineer Creator Links The system shall acquire the requested information, either from Description the system inventory or by tasking external data collection resources. Execution Level Follow Decomposition Last Modified Friday, September 08, 2000 at 03:05:34 PM Number 2 Function 2.1 Check Product Inventory decomposed Function 2.2 Prioritize Request, Determine Collector Mix, and by Task Collectors Function 2.3 Accept Data From Collectors Function 2.4 Process Data And Put Product In Inventory **Text View** Item i.4.1 collector data Links inputs Item formatted request owned by Engineer System Engineer relates to DomainSet for all tasked collectors

If the selected function has a diagram associated with it, that view is accessible by clicking on the related link at the top of the text view as shown above.

(ک



Once in the **Functional** view, you will notice that the graphical icons are also hyperlinked to their respective textual view. 📃 🖳 My Computer

CONGRATULATIONS

You have completed your first system design using **CORE**. Now, with this tour as a reference, experiment by designing your own system. Remember that **CORE** is far more powerful and flexible than we have shown in this simple example. Experiment with the other features and capabilities to get a better idea of what you can do. In fact, you will discover that with **CORE**, system engineering involves less perspiration (leaving more time for inspiration).

System engineering should be <u>productive</u> and <u>fun</u>. We believe that using CORE[®] is both.

Vitech Corporation

2070 Chain Bridge Road, Suite 320 Vienna, Virginia 22182-2536 (703) 883-2270 FAX: (703) 883-1860 Email: info@vtcorp.com Support: support@vtcorp.com WWW: http://www.vtcorp.com

CORE 3.0 Trial Limitations

The Trial version of CORE is identical to the commercial version with the following exceptions:

- The capability to save an image file, which stores the data with the program in order to eliminate the import/export cycle, has been disabled.
- The **Trial** will import/export **CORE** database files in binary format instead of *ASCII format*. This format does not permit merging of databases.
- The capability to export only the database changes has been disabled.
- Users are limited to a single project.
- Only a subset of the available reports has been provided. In addition, the capability to create and modify report scripts has been disabled.
- The capability to maintain a recovery log has been disabled.
- The bridge between CORE and RDD-100 has been disabled.
- The User/Group Tool, which allows multiple users and groups to be created within the CORE environment, has been disabled.
- An artificial limit on the number of elements in each class has been added. The user is limited to a total of 275 elements. Each class is limited to 5 elements with the following exceptions:

50 OriginatingRequirements	35 Functions
75 Items	20 Components
15 PerformanceIndex	40 Constraints
20 Links	25 CompletionCriteria
10 DomainSets	10 Interface

3 in each Program Management (Unique) Class, which include:

Activity	Leader
Product	Program
Project	Work Unit (alias: Task)
Work Package	

An artificial limit on the number of schema extensions has been added. The user is limited to:

2 additional classes 4 additional relations 1 additional facility			
	additional classes	4 additional relations	1 additional facility

However, the user is free to define as many attributes and relationships as desired.

This version of **CORE** is provided under special agreement for evaluation or academic use only. If you wish to use **CORE** for commercial or other purposes, please contact **Vitech Corporation** at (703) 883-2270 or via e-mail at info@vtcorp.com. **Vitech** has an official price list, which provides for indefinite and evaluation licenses for **CORE** as well as system engineering training classes and purchased technical services.

CORE 3.0 Product Family

Products

CORE

An integrated engineering approach for developing and conveying global process and product solutions on your desktop. **CORE** enforces consistency, interactively deriving and associating system behavior models with originating requirements and physical architectures.

CORE Enterprise

Server based application for concurrent multi-user access. It incorporates a client-server architecture and commercial object-oriented database to provide full **CORE** support to the Integrated Product Team environment.

CORE2net

CORE2net provides access to all information and models contained in a **CORE** database via Internet Browser. A separately licensed component of the CORE Enterprise Server, CORE2net turns your **CORE** Enterprise system into a Web server. CORE2net is a "CORE viewer" that does not require any special software to be installed on a user's workstation.

COREsim

A discrete event simulator option, which executes the process and physical model to provide an assessment of system performance and to verify the dynamic integrity of the conceptual design.