



**PathPoint Software**



# Dynamic Discovery for Mainframe Applications

---

## Technical White Paper

---

PathPoint Software, Inc.  
101 North Plains Industrial Road  
Wallingford, CT 06492  
Phone 203-741-0400 ext. 11  
[www.pathpointsoftware.com](http://www.pathpointsoftware.com)

## Breakthrough Dynamic Technology

Without any change to application source code, PathPoint *dynamically* captures the application-processing path followed by a specified, *user-entered*, business transaction(s) as it executes on the mainframe. In minutes, application information is captured on the mainframe and downloaded to a PC relational warehouse. Users are provided with factual information about the application, including, in the actual sequence of execution:

- Input and output screens
- Programs called
- Call statements issued
- Tables/files accessed
- How tables/files are accessed

Access to accurate information provides an unmatched understanding of mainframe applications – knowledge based upon *dynamic* execution-based fact.

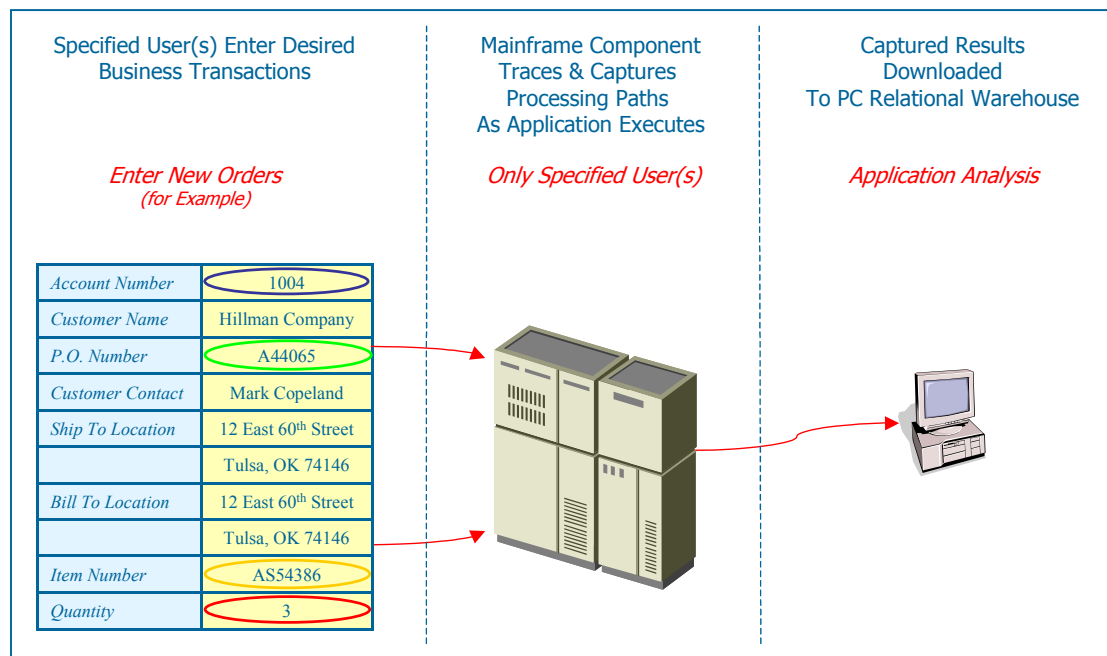
### PathPoint is Easy to Use

#### Step 1 – Enter Business Transactions

A specified (by user-id) mainframe user simply enters a desired business transaction (for example, entering an order) in either the test or production environment. The PathPoint Mainframe Component traces and captures the application processing paths followed by the business transactions entered by the specified mainframe application user. Captured information is downloaded to the PathPoint PC Analyst Workstation relational warehouse.

#### Step 2 - Analyze Captured Information

The PathPoint PC Analyst Workstation, with its relational database, enables users to have rapid point-and-click access to mainframe application processing information – users now understand how the application really works. Information is organized hierarchically in application and business terms, simplifying access and understanding:



## How Does PathPoint Work?

### Mainframe Component

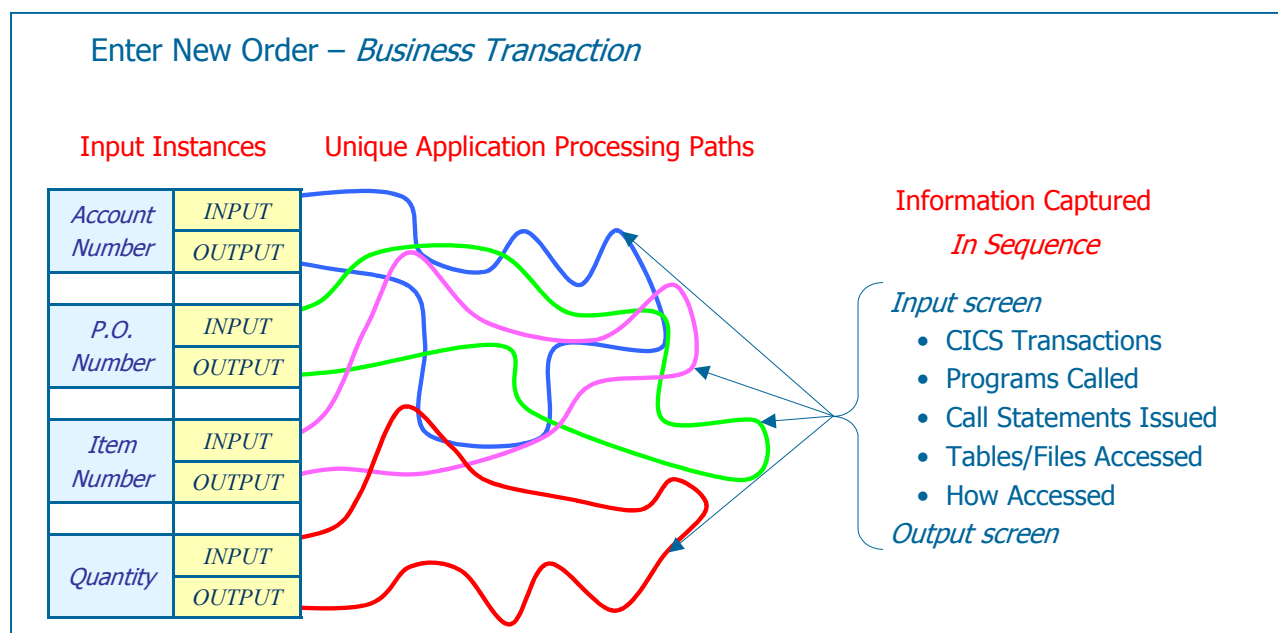
The mainframe component is comprised of two facilities: the Collection Facility and the Post Processor Facility.

### Collection Facility

The Collection Facility collects, in real-time, application activity for business transactions entered by end users. For example, when a specified user enters a business transaction (e.g., enters a new order), PathPoint will dynamically capture the application activity for each input instance (e.g. account number) for the business transaction that is entered. Information is captured in sequence, and includes: input and output screens, CICS transactions, programs called, call statements issued, tables/files accessed and how they were accessed.

When a Collection Session is started, a user ID is specified for the user who enters the business transactions of interest (up to 8 user IDs may be used). The Collection Facility then captures the actual application information for the business transactions entered by those users only.

### Conceptual Overview



### PathPoint Environment Table

In order to use the Collection Facility, you must ensure that the environment that your application operates in is defined in the PathPoint Environment Table. For example, the Order Processing application may initiate workloads in "CICSA" and "DB2A", while the Billing application may initiate workloads on "CICSP" and "DB2P". When you start a Collection Session, you simply select an environment by name in the PathPoint Environment Table, and PathPoint will initiate system level tasks to capture the relevant information from the appropriate subsystems.

## **Post-Processor Facility**

The Post-processor Facility organizes the information captured by the Collection Facility, retrieves additional data from various system catalogs about tables and files accessed by your application and stores this information in a file for downloading to the PC Component. The Post-Processor Facility is initiated by running a PathPoint generated batch job.

The following is a brief description of inputs to the Post-Processor Facility. The ones actually invoked for a particular Collection Session are determined by the Environment Table entry specified when a Collection Session was started. The Post Processing Facility software accesses all input sources for read-only processing.

- DB2 Interface
  - Collects information about DB2 calls using DB2's Instrumentation Facility Interface (IFI).
  - Reads DB2 table and, optionally, access path information from the DB2 catalog.
- IMS Interface
  - Reads the Online Log Data Set (OLDS) in the IMS/DC environment.
  - Reads the IMS Log for IMS batch environments.
  - Reads DBD information from IMS DBDLIB
- CICS Interface
  - Collects transaction, program control, temporary storage, transient data and call information for both terminal and non-terminal CICS transactions via user exits in the CICS system.
- VSAM Interface
  - Collects information about VSAM clusters from the ICF (i.e., Integrated Catalog Facility) catalog.

The file created by the Post-Processor Facility is downloaded to the PC Component using any of the commercially available mainframe utilities (e.g., "IND\$FILE" and "WS FTP Pro") for transferring files to the PC Component.

## ***PC Component***

The PC Component imports the file created by the Post-Processor Facility into an application warehouse database. It contains all the information needed to analyze your application.

The PC Component uses a relational database to store and retrieve information about your application. This technology provides flexible and powerful features to report on information about your applications.

The PC component is where the relationship between the business and technology is established and reported. By analyzing this information, one is able to understand their business applications like never before.

## Organizing Captured Information

Captured information is organized hierarchically in the PC Analyst Workstation, in application and business terms, simplifying access and understanding. PathPoint uses the following terms:

### Application

An Application is the name of the application you are analyzing, e.g., an Order Processing application. It consists of one or more Business Functions.

### Business Function

A Business Function is a major processing area within the application, e.g., Order Entry. It consists of one or more Scenarios.

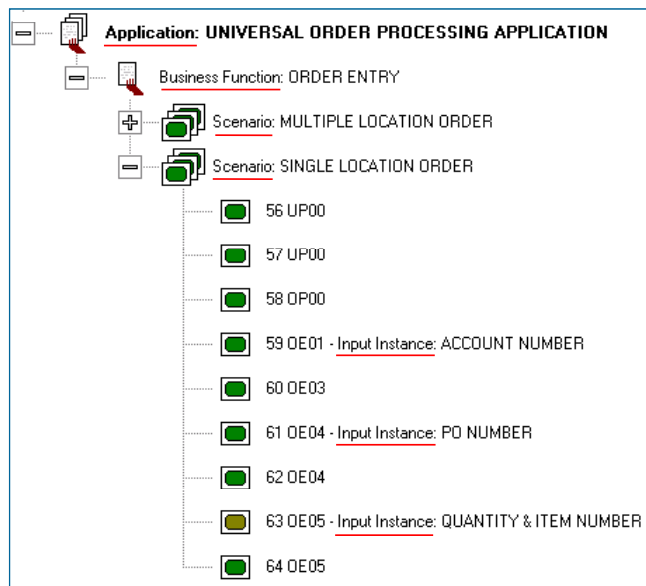
### Scenario

A Scenario is a one way that a business transaction can be entered by a user, e.g., "enter a new order" and "enter a 'blanket' or recurring order". Typically, there are many ways that a business transaction can be completed. Thus, there are often multiple Scenarios within a Business Function. A Scenario, i.e., business transaction, consists of one or more Input Instances.

### Input Instance

When entering a business transaction, a user interacts (typically fills out information on a screen and presses Enter) with the application's Host computer several times. An Input Instance is defined as each time the end user presses either Enter (or other program function key, e.g., PF3) and a response is received at the user's terminal. PathPoint captures the application activity for this Input Instance. If there are multiple Input Instances for a business transaction, PathPoint captures the application activity for all Input Instances in that Scenario.

### *PC Analyst Workstation Hierarchy of Captured Information*



## Technology

When PathPoint relates business and technology, “technology” refers to the application processes invoked on the mainframe and the databases it accesses when the end user presses Enter. It includes:

- Process Information
  - Transaction/job step attributes and execution statistics
  - Program attributes and execution statistics
  - IO and program call statement attributes and execution statistics
  - Terminal messages and message format definitions (e.g., CICS map)
- Data Access Information
  - Access path attributes for DB2 (optionally), IMS and VSAM calls
- Data Store Information
  - DB2 table information from the DB2 catalog
  - IMS DBD information from the DBDLIB
  - VSAM cluster information from the ICF catalog
  - Column/field attributes from the DB2 catalog or record definition copybook (if supplied)

## Information Retrieval

A business application’s primary function is to manipulate data using business rules. Business rules are represented in programs in the form of executable statements. An insight into the underlying application design and program logic that implements the desired business rules is in seeing the sequence in which transactions and programs are invoked and the actual sequence of IO calls made by those programs. This chronological list of selected program control and IO statements performed for a business transaction is the basic unit of information collected and reported by PathPoint. Similar to the gate flags in a downhill skiing competition, PathPoint guides you through the complex set of business rules that make up your application.

Simply showing selected statements is useful but is often too detailed when analyzing an entire application. Therefore, PathPoint summarizes program and data access activity in the form of a “CRUD” matrix. (“CRUD” is an acronym for create, read, update and delete.) The matrix shows the number of IO calls that were performed to access data stores for each Input Instance.

Note that not all statements recorded by PathPoint actually result in accessing data stores. For example, “OPEN” and “CLOSE” statements are common file operations but they do not access application data. Also, program CALL statements show program flow but have nothing to do with input and output operations. Thus, such statements, although captured by PathPoint, are not included in CRUD matrix figures.

In the case of DB2 tables, CRUD matrices only show activity against base tables. A program may reference a DB2 VIEW definition in a DB2 SELECT statement. PathPoint automatically resolves which base tables and columns were actually accessed, and it shows those base tables and base columns, as appropriate, in the CRUD matrix.

There are three forms of the “CRUD” matrix in PathPoint. The first form is the entity CRUD matrix that shows the number of program statements that create, read, update and delete records/rows in files/base tables. The second form is the transaction CRUD matrix that further breaks out file/base table activity by transaction and program within transaction. A third form is the column CRUD matrix. It summarizes the IO calls by the base columns referenced in the program statements.

The powerful relational database used by PathPoint provides the flexible information storage and retrieval capabilities needed to quickly analyze complex relationships between the business objects and the corresponding application objects that were invoked. The application objects include transactions, programs, IO statements, DB2 tables, DB2 columns, files and fields.

## **Application Analysis and Design**

The classical orientation of application analysts and designers is the need to understand "what" an application does and "how" it does it in terms of the business activities it supports. The business perspective is important because that is how the end user relates to the application. The communications with end users improve when analysts can relate their business to transactions, programs, files, etc. that support it.

A common question that is asked for a change request is "Which files are accessed?" and "How are they accessed?". To answer that question, PathPoint provides the entity CRUD matrix by Application, Business Function, Scenario, Input Instance, transaction/job step, program and Call Path. A Call Path is a user-defined group of statements that represents logic of interest to the analyst, e.g., functions to be offloaded to a network server. The entity CRUD diagram shows what is happening by showing the tables/files that are accessed and how often.

To address the "how" question, PathPoint provides a selected list of statements in the sequence that they were executed by the application. Of particular value is the "where used" statement list which shows the statements that access a particular data store. The entity CRUD diagram shows file and table activity by Scenario, Input Instance and transaction/job step. By simply double-clicking on the frequency cell in the CRUD diagram, PathPoint lists the "where used" statements along with the program making each call and the transaction being executed.

For example, suppose that the "enter out-of-stock order" Scenario entity CRUD diagram shows that the DB2 CUSTOMER base table was updated on three occasions. By simply double-clicking on the cell that contains "3" under the U column, PathPoint shows the actual "UPDATE" statements. In addition, PathPoint can show which DB2 base table columns are updated for those statements.

Other inquiries show transactions and programs executed by Application, Business Function, Scenario, Input Instance, and table/file definitions.

Altogether, PathPoint enables the analysts and designers to quickly "get their arms around" an application by simply pointing and clicking.

## **Programmers**

A primary orientation of programmers is in writing and testing program logic used to support the business rules of an application. When making a change to an application, the basic questions asked by programmers are "Which programs need to be changed and where in the programs do I look?" Furthermore, if a change is made in those programs, what is the impact on other parts of the system?

PathPoint is an invaluable tool to determine which programs, where in the programs and perform impact analysis based on what other programs access the same data stores. There is no guesswork -- the information provided by PathPoint is based on what the application actually does for the business activities affected.

PathPoint is also helpful to test program changes. The programmer can compare the difference in the IO call path before and after a program change as a means to validate expected results.

In addition, PathPoint reports provide permanent documentation on how the programs work.

## Systems Testing

Test scripts form the basis for regression testing an application. PathPoint captures and reports screen images along with data store activity. Reports from PathPoint can be used as test scripts, and they show actual application activity to confirm expected results.

## Performance Analysis

The underlying measure of application performance is time, e.g., how long does it take to do a function and is it within the time needed by an end user? PathPoint reports time by Scenario, Input Instance, transaction/job step, call path (i.e., list of statements), individual statement and access path within a statement. Also, for DB2 activity, PathPoint shows wait times, DASD service times, buffer pool activity and various call statistics by DB2 subsystem. Thus, the application performance analyst and DBA have a thorough picture of where time is being spent in terms of the end user initiated workloads.

If a DB2 statement execution time is long, then the user can simply generate the DB2 Performance Analysis report. It shows a complete picture of time spent processing a DB2 query including optionally access path used, table definitions, open statement reference, rows accessed, rows qualified, rows returned and time spent processing each table/index. Using this report, most application performance issues can be identified and resolved quickly.

Also, the CRUD diagrams by Application, Business Function and Scenario provide a complete picture of how table/files are accessed so that table/file organization and data store placement decisions can be optimized for projected workloads.

## Information Reporting

The query capabilities of PathPoint are extensive and organized to view your applications from both the business and technical viewpoints. Screen navigation in PathPoint is intuitive, and needed information is simply a "point and click" away.

The screenshot displays the PathPoint interface with several key components:

- Business Transaction(s):** A table showing scenarios like 'MULTIPLE LOCATION ORDER' and 'SINGLE LOCATION ORDER'.
- Input Instances:** A table listing instances with columns: User Seq #, Inst #, and Description.
 

User Seq #	Inst #	Description
1	56	Account Number
2	57	PO Number
3	58	Item Number
4	59	Quantity
5	60	Complete Order
- Menu Options:** Two panels of navigation options. The left panel includes 'Show Input Instances' and 'Show Tables/Files'. The right panel includes 'Show Tables/Files', 'Show Columns', and 'Show Transactions/Job Steps'.
- Call Statements Issued on Call Path:** A table of SQL statements.
 

Pgm/Pkg	Stmt #	Statement
OPC0005C	1642	SELECT FRONTLIST_QUANTITY INTO :HV-DA-TOPDISCOUN
OPC0005C	1655	INSERT INTO TOPDISCOUNTACCUM ( ORDER_NUMBER , G
OPC0005C	1786	SELECT COMBIN_BL_NEW_IND , DISC_ACCUM_CODE FR
OPC0005C	1804	INSERT INTO TOPDISCOUNTACCUM ( ORDER_NUMBER , G
OPC0005C	1937	UPDATE TOPDISCOUNTACCUM SET FRONTLIST_QUANTITY
OPC0005C	1967	UPDATE TOPDISCOUNTACCUM SET FRONTLIST_QUANTITY
- Tables/Files Accessed by Statements on Call Path:** A table showing table access statistics.
 

Table/File Name	Identifier	C	R	U	D	Total
TOPACCOUNTSCHEDULE	DVT2		1			1
TOPBOOKORDHEADER	DVT2		1			1
TOPCOTIMPRINTPRFL	DVT2		1			1
TOPDISCOUNTACCUM	DVT2	2	2	2		6
TOPDISCOUNTMATRIX	DVT2		1			1
TOPFACILITYID	DVT2		1			1
- Columns Referenced by Statements Accessing File:** A table showing column access statistics.
 

Name	Table/File	File Type	Len	C	R	U	D	All Strms
ACCOUNT_NUMBER	TOPDISCOUNTACCUM	DB2	4	2				2
BACKLIST_QUANTITY	TOPDISCOUNTACCUM	DB2	4	1	1	1		3
COMBIN_BL_NEW_IND	TOPDISCOUNTACCUM	DB2	1	1	1			2
DISC_ACCUM_CODE	TOPDISCOUNTACCUM	DB2	2	2	1			3
DISC_SCHEDULE_CODE	TOPDISCOUNTACCUM	DB2	2	2				2
FRONTLIST_QUANTITY	TOPDISCOUNTACCUM	DB2	4	2	2	2		6
GRUOP_NUMBER	IUPDISCOUNTACCUM	DB2	2	2				2
ORDER_NUMBER	TOPDISCOUNTACCUM	DB2	4	2			0	2

## Hard Copy Reporting

If a hard copy report is helpful to complete your analysis or supplement existing documentation, then the reports of PathPoint and the flexible screen reporting features provide a tangible result.

Reports generated by PathPoint include:

### Application Objects

This report lists the application objects in an Application or Business Function. It includes transactions/job steps, screens' map set and map name, programs and tables/files.

This report is the inventory of application objects that support the piece of the business being analyzed.

### Application Summary

This report summarizes the application objects that support an Application or Business Function. The report includes the Scenarios that make up the Application and/or Business Function, the number of transactions/job steps, screens, programs, tables/files, Input Instances, total calls and elapsed time.

This report is useful for documenting the size of a business transaction and planning work assignments.

### Scenario Overview

This report provides a complete picture of what an application did and how it did it to support a business transaction. It includes a summary of all transactions and programs invoked, tables/files accessed in CRUD format, summary of table/file activity by data manager, formatted input and output screen messages and the actual program IO statements executed in chronological sequence.

This report is useful for understanding program flow, identifying mainframe subsystems that were involved, documenting detailed program logic and addressing application performance issues.

### Input Instance Overview

This report is similar to the Scenario Overview report except that it contains information for a single Input Instance.

### Scenario Screens

This report shows the human/computer interactions, i.e., Input Instances, as seen by the end user. It includes Input Instance number, formatted input and output screens, end user time and response time.

This report is useful for training, workload planning, systems testing and other activities where seeing the actual end user screens is important.

### DB2 Performance Assistant

This report provides detailed information about the execution of a single DB2 statement including its DML, its access path, table/index scan statistics, sort statistics table definitions, column

definitions, index definitions and tablespace definitions. Also, it includes run statistics about the Input Instance and transaction/job step in which the statement was executed.

This report consolidates information useful to a DBA to identify and correct performance issues associated with a single DB2 call.

### **Screen Reports**

Most information is displayed in tabular form. A useful feature is the ability to print the contents of a window. To print a window, the user simply clicks the File:Print menu option or clicks the printer icon on the menu bar.

### ***PathPoint Platform Support***

- Operating Systems
  - MVS (z/OS and OS/390)
- TP Monitors
  - CICS/TM
  - IMS/TM
- Data/File Access
  - DB2
  - IMS
  - VSAM
  - Temp Storage
  - Transient Data
- Batch Environments
  - DB2
  - IMS
  - VSAM/Non-VSAM (*coming soon*)