3270Builder: Product Overview

3270Builder is a developer's workbench for re-engineering legacy IBM mainframe applications. Fully compliant with the NEXTSTEP Database Kit and Enterprise Object Frameworks, 3270Builder makes the process of integrating legacy resources and data with your mission critical NEXTSTEP applications as easy as dragging and dropping an Interface Builder palette object.

Utilizing advanced pattern recognition algorithms, 3270Builder automatically produces DBKit and EOF models for mainframe applications by analyzing 3270 screens and the keystrokes necessary for their traversal. Users simply assign appropriate field names to screen regions on individual or multiple screens. Transparently, 3270Builder builds a sophisticated data access tree and stores it as a DBKit or EOF model for access by Interface Builder and run-time applications.

3270Builder takes advantage of Conextions' CX3270 object to provide high-performance, concurrent access to up to 64 IBM applications on the same or different hosts. The result is a powerful, yet, easy-to-use tool for re-engineering legacy applications, building GUI front ends for existing mainframe applications and for developing new applications that can seamlessly integrate legacy data from multiple IBM mainframes with NEXTSTEP information and resources.

What is Re-Engineering?

Re-engineering is defined as the process of rearchitecting centralized mainframe-based Management Information Systems and legacy applications to a client-server architecture and a distributed computing environment. The re-engineering process is often combined with downsizing when all or part of a legacy mainframe application is moved to a smaller, often lower-cost, computer platform.

The re-engineering process involves complex political, economic, and legal restraints under which corporate computing operations must run. In addition to this, it involves the process of developing mission critical custom applications to address the rapidly changing needs of the corporate marketplace.

A crucial first step in this process is to determine what legacy applications need to be re-engineered and to define the approach that the MIS organization should take in the overall process. The early adaptors of the re-engineering methodology learned the hard way that the "all or nothing" approach, in which the entire legacy system is moved to a microcomputer based client-server model, often creates more and bigger problems than it solves.

Re-engineering applications in part, rather than entirely, proved to be the safest and the most cost effective approach. For many companies, moving vital computing chores (such as inventory management, airline reservations, customer service, etc.) too quickly can be disastrous if the new system has not undergone long and extensive field testing (and may therefore have serious bugs).

A better alternative (often called front-ending or partial re-engineering) has been recommended by firms like Booz, Allen and McKinsey to many of their clients with large mainframe systems. In this approach a front end for a legacy mainframe system is developed first. Users are able to take advantage of a newly designed Graphical User Interface (GUI) and an efficient operating environment while the legacy back end is re-engineered and moved to a smaller computing platform. Understanding the legacy system is crucially important to the success of any MIS re-engineering project.

Understanding Legacy Systems

Legacy systems are large, interconnected sets of applications that run on mainframes and provide a variety of end user services and interfaces. These legacy applications are critical to a company's existence and are designed to provide real-time information access to a centralized data repository. They will typically encompass self-contained databases, user interfaces, communication subsystems, and query facilities.

The safest candidates for step-wise re-engineering are read-only applications in which data continues to be stored on and accessed from mainframes or large minicomputers.

Another good re-engineering candidate is an application involving cooperative processing. In cooperative processing, client-server databases can be used to ensure a stable and secure operating environment. Forms based applications used for order entry, inventory management or customer service are also good candidates for re-engineering.

The key to re-engineering success lies in the complete understanding of the legacy system under consideration. Among the important issues to consider are legacy system security requirements, down-time tolerance and availability requirements, throughput and latency requirements, and overall performance considerations. Answering the questions raised by these issues will help to identify a safe and cost-effective approach to MIS downsizing and re-engineering.

IBM mainframe applications use the 3270 Data Stream format and protocol when communicating with a terminal. The data stream consists of a number of commands and orders, along with their respective parameters and attributes. The major difference between 3270 and ASCII type terminals is that 3270 terminals operate in ***block mode,*** whereas ASCII terminals operate in c***haracter mode***. In a block mode operation, changes to the screen are performed locally until the terminal user enters one of the special Attention Identifier (AID) keys (typical AID keys are PF keys, PA keys, Clear, and Enter). New host data is presented to the terminal as an entire screen or logical partition of a screen at once.

The image of the 3270 screen that is kept in memory is called a Presentation Space. You cannot see the Presentation Space, but you can see its translated display output on the 3270 screen. Each 3270 screen consists of a variety of fields with their respective attributes. A typical legacy application is a collection of screens (forms) with each screen consisting of a number of fields and their respective field attributes.

Connecting to IBM Legacy Systems

IBM legacy systems are frequently integrated into an enterprise networking environment and connectivity to them must be provided over various network topologies and protocols. De facto standard networking protocols such as NetWare, TCP/IP and SNA can operate over various communication media and low level protocols such as Ethernet, Token Ring, FDDI, T1, SDLC, and ATM. Choosing the right configuration to connect to legacy mainframe systems may be a challenge even for experienced network and system administrators.

3270Builder offers several alternatives when connecting to legacy systems and provides end users with complete transparency from network topologies, protocols and communications media. From the same 3270Builder application you can access IBM mainframe applications through TCP/IP Gateways, CoaxAccess™ Gateways and Novell™ NetWare™ for SAA Gateways.

TCP/IP Gateways: TCP/IP gateways for IBM mainframes are becoming increasingly popular in the Enterprise Network Environment. They are available from a variety of vendors, which include:

- IBM's 3172 LAN Channel Station and TCP/IP software
- OpenConnect Systems' gateway servers (formerly Mitek)
- Interlink's Access/MVS software and ACS 9310 Ethernet Controller
- Apertus Technology's DataStar Enterprise server
- Brixton's TCP/IP Gateway server.

3270Builder and 3270Adaptor allow you to connect to an IBM mainframe using any of the TCP/IP gateways that support the TN3270 protocol.

Conextions' CoaxAccess-PC Gateways: In traditional IBM environments, 3270Builder connects NEXTSTEP workstations to IBM 3x74 Terminal Controllers through Conextions' CoaxAccess-PC gateway using SNA Distributed Function Terminal (DFT) protocols.

The CoaxAccess-PC gateway may be configured to support 5, 10, 15 and 20 concurrent terminal sessions. Any NEXTSTEP workstation on the network can access the IBM mainframe through the CoaxAccess-PC gateway.

Novell's NetWare for SAA Gateways: 3270Builder provides transparent IBM mainframe access using Novell NetWare for SAA gateways.

Re-Engineering Legacy Systems with 3270Builder

The 3270Builder Workbench was specifically designed to address MIS re-engineering needs when dealing with legacy applications. 3270Builder employs a low risk step-wise approach to MIS re-engineering that is recommended by leading authorities in the field (such as Booz Allen, Andersen Consulting, and McKinsey) and represents the leading trend in Fortune 1000 companies. 3270Builder allows you to interact with a mainframe application while it transparently generates a detailed application model and stores it as a database map using NEXTSTEP's DBKit or Enterprise Objects Framework technology. The model contains information concerning the location of all data within the application and the logic necessary to access that data programmatically.

Every 3270 screen corresponds also to an entity in the database model and is represented as a node in the application map. The fields on the screen are represented as attributes of a database entity. 3270Builder automatically creates a signature of every application screen and generates an application traversal schema. With 3270Builder, the process of downsizing and re-engineering a legacy IBM application is split into three steps:

Step 1. The developer builds a database model (~*.dbmodel*) of a legacy IBM application by traversing through a legacy application and pointing and clicking at screens and fields that are to be added to the model. The *.dbmodel* is then stored as a standard database model on the NEXTSTEP computer.

Step 2. A new Graphical User Interface (GUI) front end (a client application) is built by directly accessing the dbmodel through the Interface Builder and DBKit. The new application uses the 3270Adaptor and DBKit to access the mainframe data at runtime.

Step 3. The legacy IBM application (the back end) is moved from the mainframe to a LAN database server. There are no changes to the client GUI and no disruption to the end user operating and production envi ronment.

The process outlined above is just one part of re-engineering a Management Information System. The other important part is creation of new custom applications that will increase the efficiency and productivity of the business process.

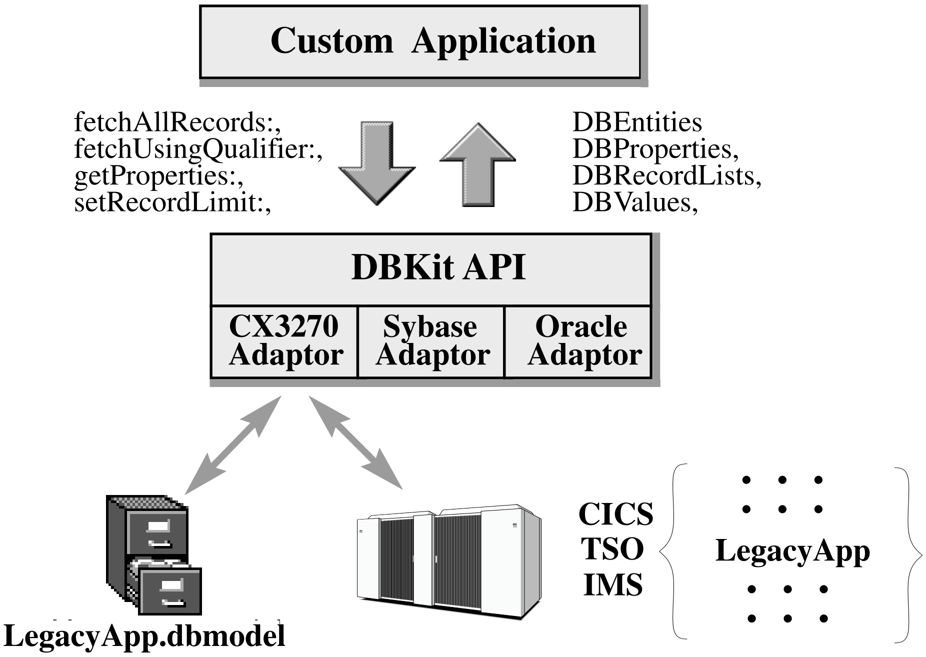
3270Builder Workbench has proven to be a very effective tool for both MIS and Business Process (BP) re-engineering. It allows early prototype development and deployment, offers a new way of manipulating and processing legacy data and cuts the development time by an order of magnitude.

Understanding the Technology

As we discussed earlier, 3270Builder allows users to interact with a mainframe application while it automatically builds a model of the application. This model contains information concerning the location of all data within the application and the logic necessary to access that data.

In order to understand the way that this model, or *.dbmodel* as it’s referred to in NEXTSTEP development circles, allows us to access information, we must first understand the components involved with the DBKit development environment. There are three major components in any application that uses the DBKit technology: the application itself, the DBKit Adaptor and the *.dbmodel* file. We will now examine each one of them

- **Custom Application.** An application is developed exactly as all other NEXTSTEP applications are developed. In order to gain access to data via the DBKit interface, the application will use standard DBKit methods such as **findDatabaseNamed:**, **connect:**, **fetchAllRecords:**, etc. These database access commands are exactly the same whether you are accessing data from an ORACLE database, a Sybase database or a legacy mainframe application using 3270Builder technology.
- **The DBKit Adaptor.** The DBKit adaptors perform all the necessary conversions of the high-level DBKit commands into the precise syntax required by the back-end data to which the adaptor is connected.
- **The Database Model.** The *.dbmodel* file contains all the information necessary to understand how all data is structured as well as any relationships or interdependences which exist with that data. In addition to this, the ~*.dbmodels* created with 3270Builder also contain a map and traversal schema for an application.



When working with traditional databases such as ORACLE and Sybase, the database server does most of the work when an application requests data. The adaptor for these databases is primarily concerned with managing the network traffic necessary to transmit a request to the server and then accept the retrieved data and pass it to the NEXTSTEP application that requested it.

The CX3270Adaptor used by 3270Builder has a much more difficult task to perform. It interprets DBKit query commands and performs a set of actions to allow the data managed by the mainframe to be accessed with the ease of a database. In addition, because the IBM mainframe world is governed by different data standards and communications protocols, the adaptor must also perform significant communications management and data conversion functions.

Fortunately, all of the activity performed by the adaptor is transparent to developers and end users. Another major benefit to this approach is the portability of the developer's application later to other data servers without having to discard large amounts of mainframe specific code - just reconnect the mainframe links to a new data server and the application can remain the same.

The 3270Builder approach of dealing with mainframe data as if it were a database provides for easy integration of legacy data into the mission critical custom applications. In addition, 3270Builder allows developers to combine data from multiple IBM applications (that could also reside on different IBM mainframes) with information stored in a local database (like Sybase or Oracle) or on a file server in a single NEXTSTEP application, thus offering tremendous power and flexibility for Business Process re-engineering.