DB2 Package Binds

Session 670

Laura Heist, Texas Instruments
Louise McCrorie, Belk Stores Services

The Purpose

To acquaint the audience with how one Composer client integrated the MVS Host Implementation Toolset (IT) process into their DB2 environment

The Assumptions

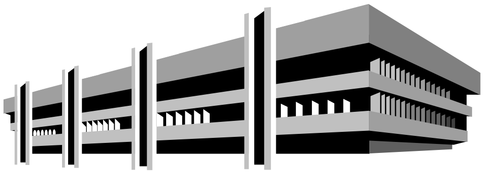
- Familiarity with IBM’s DB2 relational database concepts
- Knowledge of Composer client/server development

Warning: Presentation can become rather technical

The Plan

- The Setting
- The Players
- The Dilemma
- The Solution
- The Implementation
- The Result
- The Future

The Setting

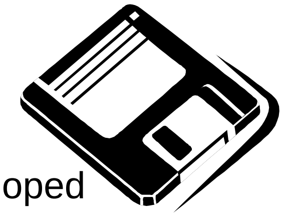


- Belk Stores Services
  - Private, family-owned retail department store chain
  - Based in Charlotte, NC
  - Over 300 stores in 14 southeast U.S. states
  - In operation since 1888

The Players

- Composer software
- Other software
- The application

Composer Software



- Version 5.3.0 installed April 1995
- Microsoft Windows 3.11 client environment
- No Host Construction; code developed and generated on OS/2 platform
- Composer 3 Windows NT CommBridge using TCP/IP from client and LU6.2 to MVS
- MVS Host IT
- Host Encyclopedia
- MVS CICS cooperative servers
- MVS batch processes

Other Software

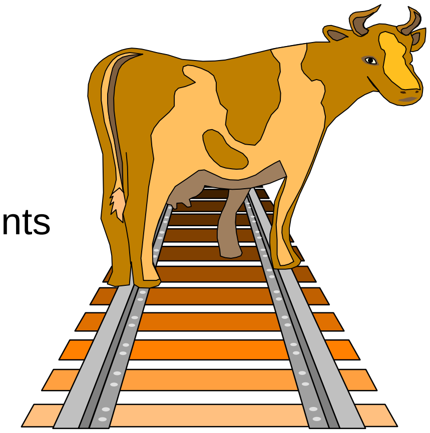
- CICS 3.3
- MVS DB2 3.1 and OS/2 DB2/2 1.0
- COBOL/370
- Separate DB2 subsystems for Production, QA, and Development environments

The Application–
Inventory Management System

- Client/Server Windows/MVS
- On-line team of 3-4 analysts/developers
- Batch team of 2 analysts/developers
- 20 on-line windows
- Batch interface to Data Warehouse (Unix Informix)
- 12 batch programs

The Dilemma

- Belk Standards
- MVS Host IT Requirements
- DB2 Primer



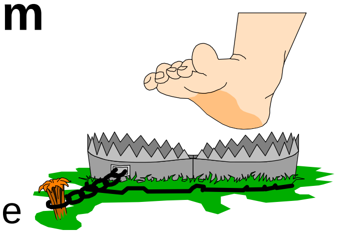
Belk Standards

- Belk standards
  - Legacy
    - One plan per program/DBRM per plan where possible in on-line
    - Dynamic linking of programs and called programs
  - Composer
    - Reuse as much code as possible
    - Package application into one server module per client module

MVS Host IT

- Default is to prepare DB2 applications binding all DBRMs in a load module to one plan
- Possible to dynamically link Common Action Blocks (CABs) and procedure steps

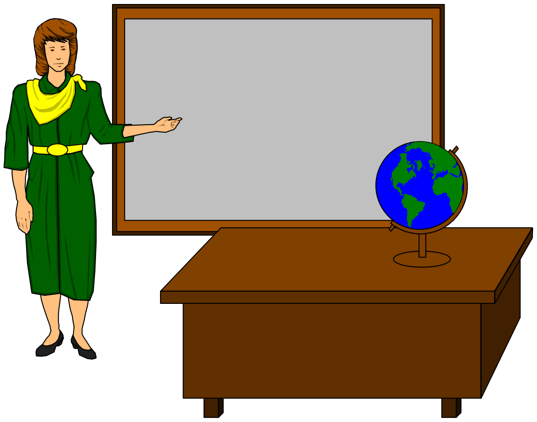
The Problem



- Maximized reusability
  - Many shared CABs
- Dynamic linking ensures executing the most recent code
- But...static binding (binding all CABs into each server module)
  - Changing one server module CAB causes DBRMs in other servers to be changed
  - Must rebind all servers using CABs or get -818 SQL code
- Brownstone—DB2 repository tool for legacy; manual process for Composer

A Brief Aside–A DB2 Primer

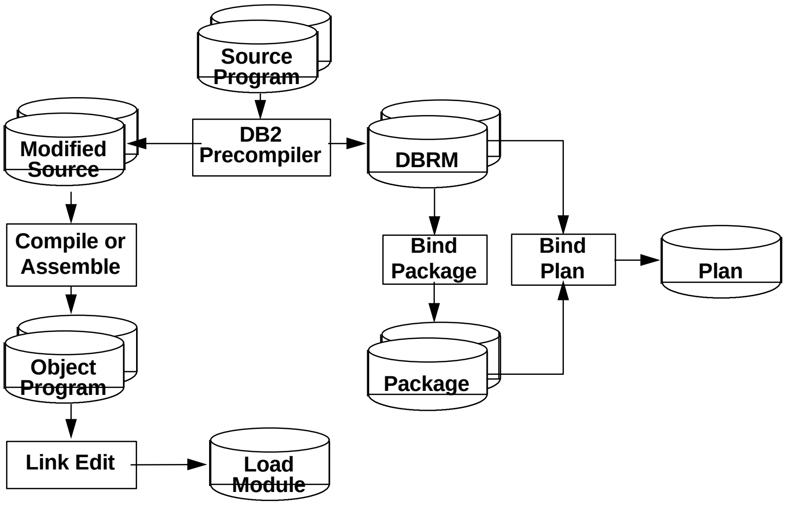
- What is binding?
- What is a DBRM?
- What is a plan?
- Why do I care?



DB2 Bind

- DB2—IBM’s relational MVS database product
- DB2 programs include Structured Query Language (SQL) to manipulate and access data
- Cannot compile program until SQL statement(s) converted into language recognized by compiler

Preparing an Application Program to Run



Source Program

DB2 Precompiler

Modified Source

DBRM

Compile or Assemble

Bind Package

Bind Plan

Plan

Object Program

Package

Link Edit

Load Module

DB2 Precompiler

- Validates SQL statements
- Replaces SQL statements in program with compilable code
  - CALL statements and entry point into language and environment-specific attachment program
  - Attachment program handles all communication between MVS and DB2 resources
- Creates a Database Request Module (DBRM) to communicate SQL requests to DB2 during the bind process
  - Contains all extracted SQL statements from a program in DB2 internal format
  - Located in external library on MVS accessed during bind process

Binding is ...

- After precompile DB2 statements, compile resulting code into a load module or executable
- BUT... DB2 only executes plans
- SO... Binding creates DB2 plan or package
  - Converts the DBRM SQL structures into DB2 run-time structures

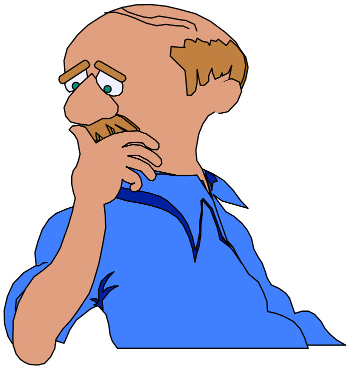
Plans and Packages are ...

- Structures = plans or packages
  - Composed of DB2 runtime structures with information on
    - » Where data lives
    - » How to get there (access path)
    - » What to do once there

Which Structure Type?

Plan ...

- Bind DBRMs once to a plan
- If any DBRM changed, must rebind entire plan (<u>all</u> members)



... or Package?

- One DBRM per package means program changes only require ONE package to be rebound
- Can add packages to a Collection (group of packages) after plan is bound AND no rebind of the plan is required!
- Can bind one DBRM to multiple packages in different collections and locations
- Can have multiple versions of a package
- User-dedicated DB2 access thread with large plan ⟶ enhanced performance
- But...must have DB2 2.3 or higher (3.1 for package level statistics)

Packages are Deceiving ...

- A DB2 package is not the same as a Composer package
  - Composer package**:** a method to group several objects into one load module
  - DB2 package**:** a method to create multiple SQL runtime objects for a single DB2 plan

Three Plan Bind Options

1. Bind DBRMs directly to a plan

- BIND PLAN(XXXX) MEMBER(AAAA, BB, CCC)
- Small applications
- Applications unlikely to change often
- If all resources must be acquired when a plan is allocated

Second Bind Option

2. Bind DBRMs using a package list

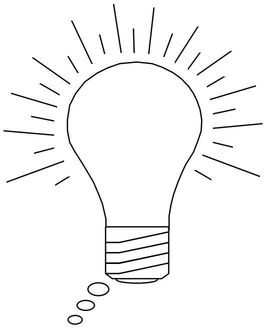
- BIND PLAN(XXXX) PKLIST(GROUP1.*)
- Include all packages in entire collection (GROUP1)
- Use if application will change significantly or if in early development stage
- Can specify different options per package or default to plan options

Third Bind Option

3. Bind DBRMs directly to a plan and use a package list

- BIND PLAN(XXXX) MEMBER(AAA, BB, CC) PKLIST(GROUP1.*)
- Maintain existing applications
- Can specify plan options AND package options for specific packages

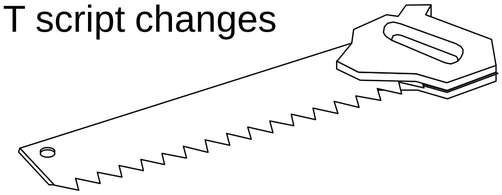
The Solution: DB2 Package Binds !

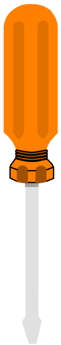


- Can incrementally develop application without having to rebind the plan each time
- Can recompile and link dynamically linked CABs contained within one server module without having to rebind other plans containing those same CABs!
- Can take advantage of using dedicated DB2 threads to enhance performance

The Implementation

- Develop new naming conventions and revise standards
- CICS/DB2 changes
- MVS Host IT configuration changes
- MVS Host IT script changes





Before Binding Packages

- Develop a naming convention and strategy for the most efficient and effective use of your packages/plans
  - Determine when your application requires that resources be acquired
  - Determine if the program will ever be used in a release of DB2 earlier than release 3
  - Maximize performance by having large plans if DB2 release 2.3 or later

Verify DB2 Installation Options

- Average number of:
  - Packages in DB2 system
  - SQL statements per package
  - Package lists per plan

Belk’s Strategy

- Single DB2 plan per major application system
- Create plans in advance with package lists using Collection ID
- Each module containing DB2 SQL statements will be bound as a package referencing the Collection ID; this automatically includes the package in plan
- ‘1 Plan = 1 DBRM = 1 Program’ standard for legacy applications only
- Each application system considered for its own dedicated DB2 thread

Belk’s Implementation

- Plan name = Application System Name
- Different DB2 subsystem for Prod, Test, QA
- Collection name = Application System Name
- Each plan bound by DBA at system start-up
- Commonly called legacy programs
  - At first, bind directly to plan to prevent having to free plan, bind again as package and rebind ALL referencing legacy programs
  - Eventually will have own Collection and be included in multiple package lists
- RI Triggers bound directly to plans because of batch limitation; but generated seldom

Example Implementation

- BIND PLAN(XXX) PKLIST(XXX.*) MEMBER(SECURITY, RI1, RI2, RI3...)
- BIND PACKAGE(XXX) MEMBER(PROG1)
- If call internal security, eventually will be:
  - BIND PLAN(SECURITY) PKLIST(SECURITY.*)
  - BIND PLAN(XXX) PKLIST(SECURITY.*, XXX.*)

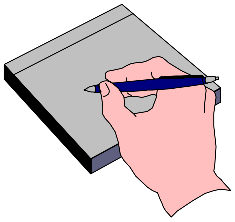
CICS/DB2 Changes

- Do not use Composer plan exit TIRC$EXT:
  - Returns plan name = program name
- All server transaction codes have own Resource Control Table (RCT) entry associating them to plan

MVS Host IT Changes

- Minor changes:
  - Changed Composer default script file
  - Loaded and included in CICS target environment

TIXMVSLM Script Changes



- Currently call CLIST to prompt for DB2 OWNER ID, QUALIFIER ID, and COLLECTION ID bind parameter values
  - When interface Composer to Brownstone this will be automated
- Change call to Bind Procedure to pass DB2 bind parameters read in from called CLIST
- Change bind section of script to bind each DBRM as separate package using input DB2 values

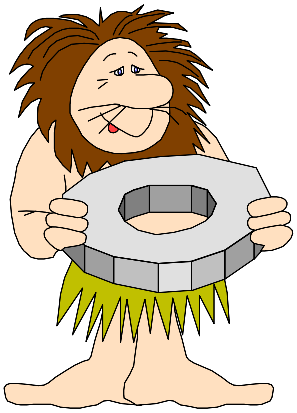
The Result

- Minimize -818 abends
- Can change reusable code without having to bind additional plans
- Always execute current code
- Maximize performance by using dedicated DB2 thread
- Binds less costly
- Less stress on DB2 catalog
- Programmer more productive since bind of packages is faster

Warnings

- Load modules and DBRMs must be kept in sync (security)
- Must coordinate changes to reusable code with other routines calling shared CABs
- CICS RCT exit must be maintained when add new servers (transaction codes)
- Must migrate script file customizations to new releases of Composer

What about Batch? and Why?



- Use default Composer script file (DBRMs bound to one plan per load module)
- Static linking
- Different target environment to handle different script and library files

The Future

Proposed Features

- Option in Host Construction and MVS IT to bind one DBRM per package
- Package is part of one or more Collection(s)
- Plan is bound with Collection in separate step
- RI Triggers, Procedure Steps, Action Blocks, External Action Blocks are eligible for package processing
- Modifiable user exit will be supplied
- Load module name used as plan name
- Plan name cannot be changed in Composer
- Must bind multiple packages per load module into a single plan outside the toolset

In Summary...

Why?

- Can change reusable code without having to rebind other plans
- Minimize DB2 catalog contention and bind time
- Always execute most current copy code
- Maximize DB2 performance by using dedicated DB2 thread where warranted

---

How?

- Minimum changes to MVS Host IT (easiest part of process!)
- Manually maintain CICS RCT entries
- Minor DB2 installation changes
- Development coordination important!

DB2 Package Binds

Session 670

Laura Heist, Texas Instruments
Louise McCrorie, Belk Stores Services