Automated Modernisation of AllFusion Gen systems: new approaches to rejuvenation and maintenance

Dave Tomkins

Jumar Solutions

Session: 24

Tuesday, June 13

EDGE Europe Presentation Abstract

“Do you wish to modernise, upgrade or re-platform your existing Gen applications in a quicker and more cost effective way, without tying up your valuable development resources? This presentation will demonstrate new automation software solutions that will dramatically change the way you approach this type of project, and deliver results far quicker than following manual coding methods. Perhaps you would like to extract your valuable business rules and expose them as reusable Web Services, components or service based assets? Maybe your customer is asking for a new Web User Interface, or you need to re-engineer your client server application from a ‘fat’ to a ‘thin’ client model, or split your 3270 procedures into clients and servers for flexible deployment.”

Jumar Brands

Jumar:Links - The brand for Tool Integration

Jumar:Xtras – The brand for Productivity

Project Phoenix - The brand for AllFusion Gen Application modernisation software and services.

Why Consider Modernisation?

- Pressure to lower system maintenance costs
  - Reduced duplication
  - Increase reuse
  - Enhanced standardisation
- Business enhancements wanted faster
- Customer pressure – e.g. new UI
- Wholesale replacement – too high cost or risk
- Enhanced performance required
- Increasing integration necessary. More sharing of business rules
- New standards set – new target technical architectures

Modernisation and Re-factoring Brings Significant Challenges

- Problematic to get sponsorship as business benefit not always obvious or immediate
- Time consuming and resource intensive
- Competes with business functionality enhancement plans
- Risky – changing what works well in production
- Complexity of task depends on nature of current/legacy systems
  - Standards adopted
  - Diversity of templates and development approaches. For example:
    - Block Mode: Presentation logic and business rules are mixed together
    - GUI Fat-Client: Business rules split between client and server leading to poor scalability
    - GUI Thin Client: A better starting position but still Gen specific validation, exit state usage etc.

Modernisation – Benefits

- Incremental enhancement, no big bang replacement needed, keep what works well
- Open systems and integration of business rules, extend system usage with better returns on Investment
- More re-use, less redundancy
- Asset based development – more open architectures
- Ride with technology changes – eg downsizing, new environments
- Small investment can give impression of big changes – re-architecting does not have to be too expensive
- Opportunities to ‘tidy up the house’ – standardisation
- Maintenance productivity improvements – reduce total cost of system ownership

Modernisation - Key Considerations

- Manage risk of any transition
- Manage transition quality
- Co-exist with ‘business as usual’ changes
- Speed – disruption minimised, business benefits faster
- Cost – competitive with onshore and offshore manual alternatives

The answer is a new way to approach this type of project – where bulk changes are executed automatically using Project Phoenix rejuvenation software.

Modernising and upgrading AllFusion Gen Applications with Project Phoenix

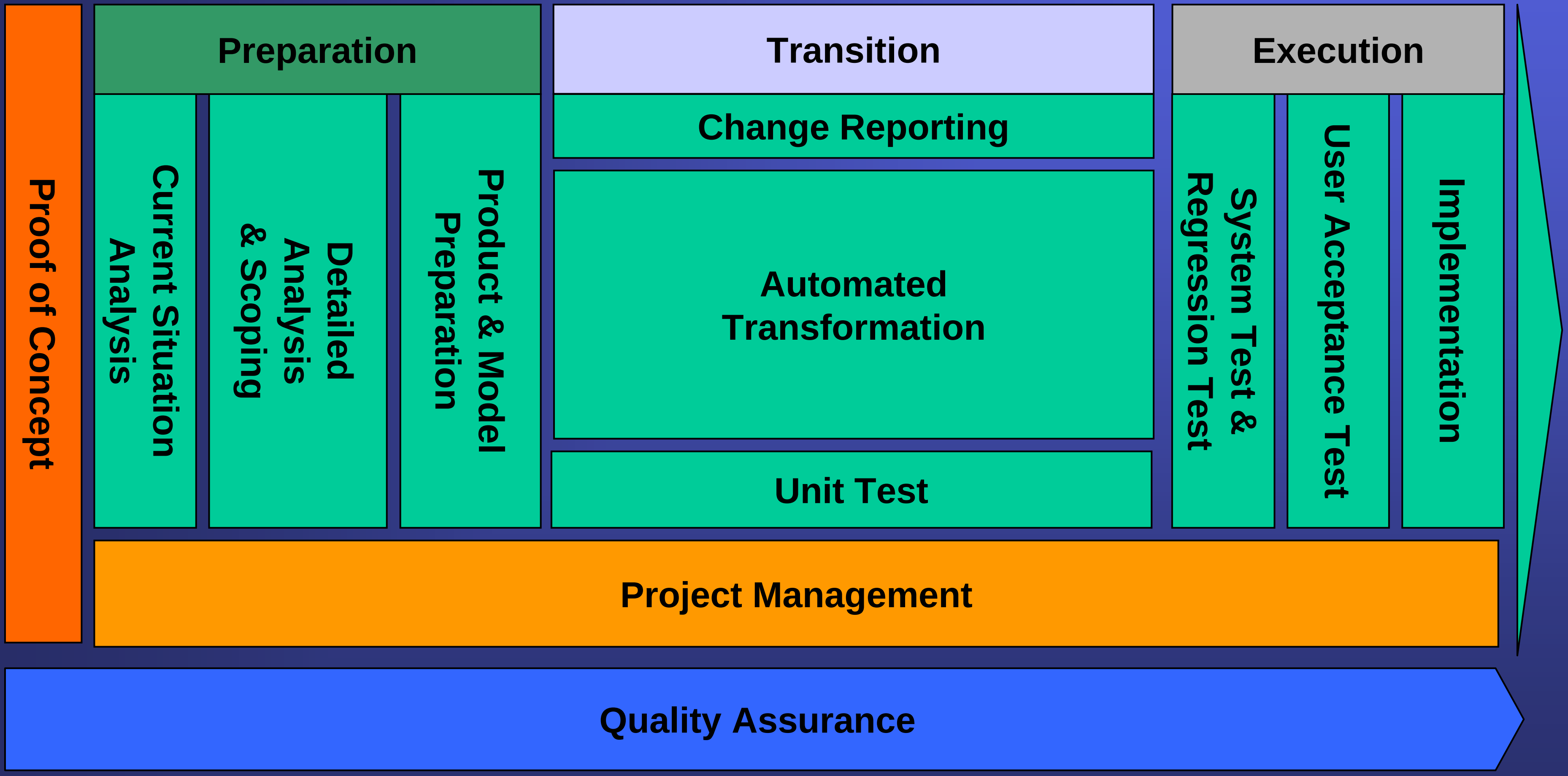
Project Phoenix:
The Automated Modernisation Solution

- A unique combination of consultancy and flexible automation tools to rejuvenate/modernise your Gen applications
- Highly cost effective through the use of automation – the result is 30-40% of the cost of completing manually
- Potential for low impact on release and resource planning due to shortened lifecycle
- A favourable alternative to buying packages or pursuing a higher risk offshore option
- Success depends on preparation, analysis and taking advantage of automation at all stages

Project Phoenix:
The Automated Modernisation Solution

- Project Phoenix software specifically designed to automate repetitive manual analysis and re-factoring tasks
- Existing business rules are kept intact
- Different ‘flavours’ of re-factoring available - configurable flexible solutions
- ‘One stop’ automation workbench, tailored to needs
- High % automation achieved based on repeatable processes and patterns
- Full lifecycle automation support
- High quality – fewer errors in automated repeatable process
- Deliverables - highly maintainable in Gen, well documented
- Replaces costly coding effort and reduces unit test and cycles
- Batch automation reduces transition time and overall lifecycle lapse time – benefits delivered faster
- Re-factoring can now become a viable business proposition

Project Phoenix: Modernisation Framework



Proof of Concept

Preparation

Current Situation Analysis

Detailed Analysis & Scoping

Product & Model Preparation

Transition

Change Reporting

Automated Transformation

Unit Test

Execution

System Test & Regression Test

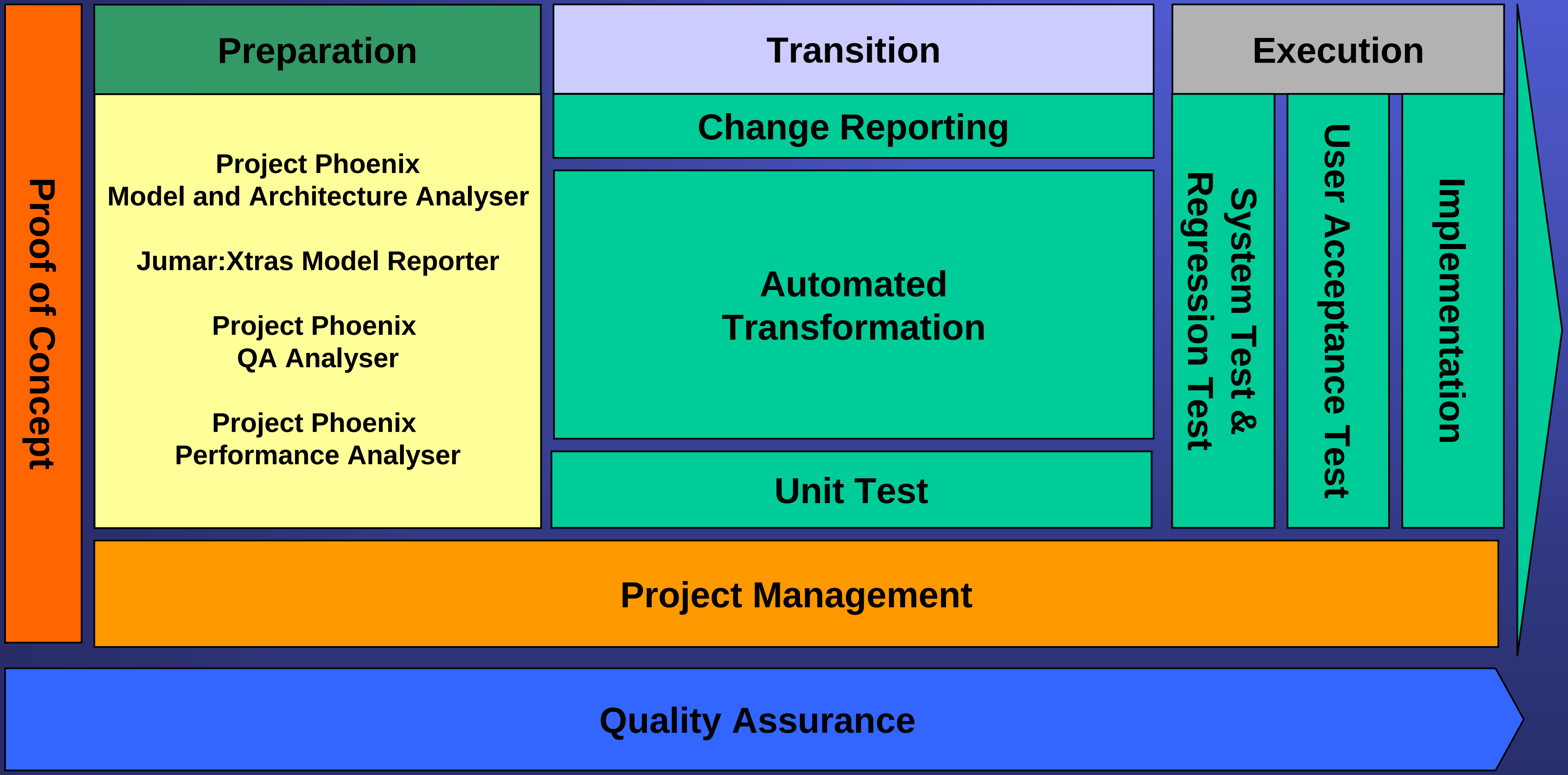
User Acceptance Test

Implementation

Project Management

Quality Assurance

Preparation Phase
Tool Support



Proof of Concept

Preparation

Project Phoenix
Model and Architecture Analyser

Jumar:Xtras Model Reporter

Project Phoenix
QA Analyser

Project Phoenix
Performance Analyser

Transition

Change Reporting

Automated Transformation

Unit Test

Execution

System Test & Regression Test

User Acceptance Test

Implementation

Project Management

Quality Assurance

Project Phoenix:
Preparation Phase

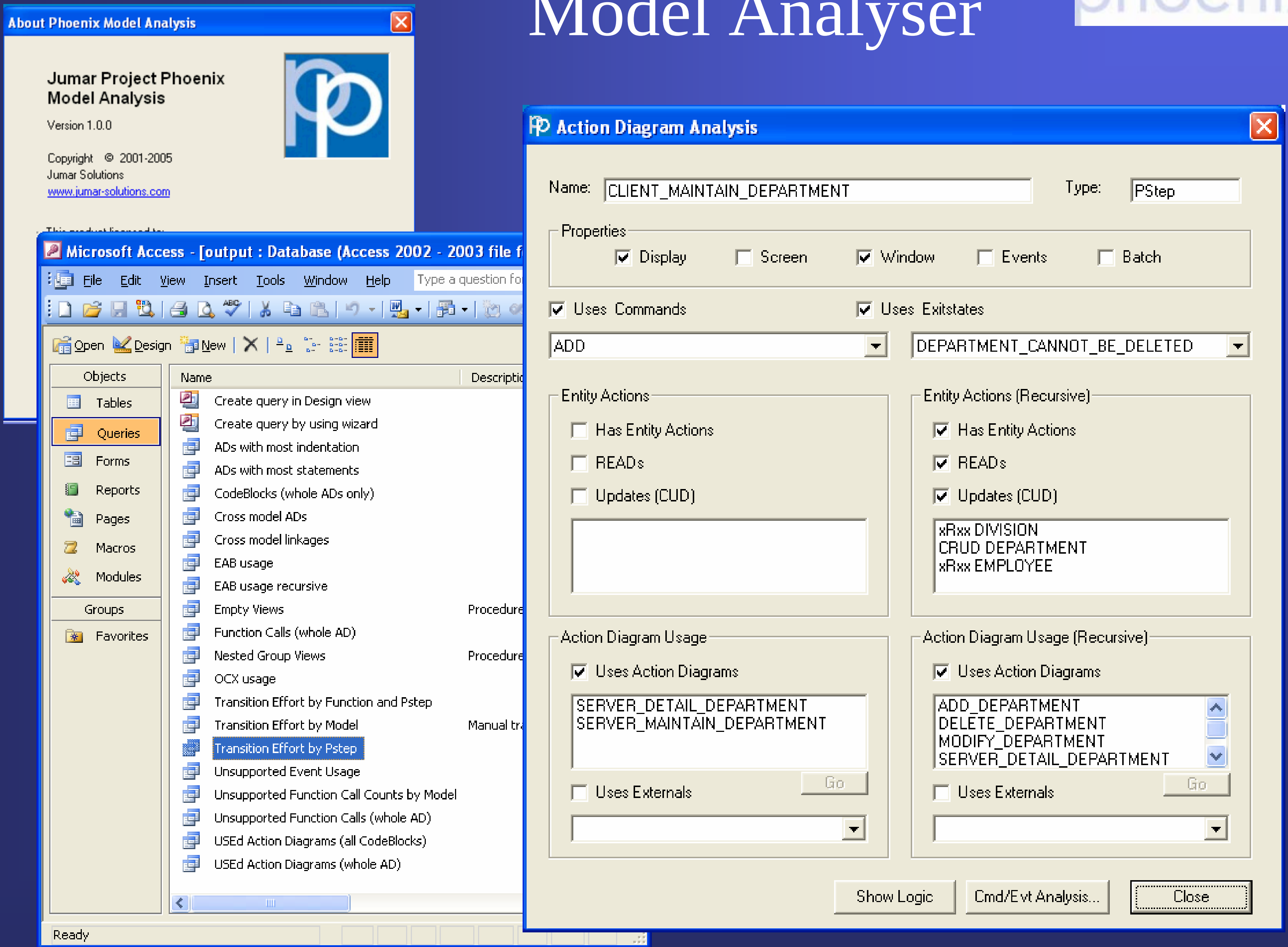
- Effective analysis of current Gen systems is essential to:
  - understand size and complexity of transitioning required
  - identify degree of standardisation
  - identify site-specific traits that may require automation to be configured
  - Identify preparation and pre transition amendments

- The Jumar Project Phoenix analysis suite of software:
  - works across multiple Gen models
  - reports on ALL objects in Gen and identifies coding patterns
  - provides full management information to support scoping, estimation & planning of the modernisation initiative
  - Deepest insight yet into existing systems, and essential for identifying services, performing UI transitions and re-architecting
- Briefly review some features of the software

Project Phoenix:
Model and Architecture Analyser Benefits

- Fast Investigation of detailed content of a model or collection of associated models
  - Specific transition issues
  - Complexity - size numbers, level of nesting, dependencies etc
  - Projected work effort (customised metrics are possible)
  - Degree of automation achievable (with and without tailoring)
- Detailed analysis of Action Blocks and PSteps
- Quick overview of the complexity of single AB/PS within the tool or whole model(s)
- Cross model dependencies identified
- Access database based tool - more flexibility for bespoke reporting

Model Analyser



About Phoenix Model Analysis

Jumar Project Phoenix
Model Analysis

Version 1.0.0

Copyright © 2001-2005
Jumar Solutions
www.jumar-solutions.com

Microsoft Access - [output : Database (Access 2002 - 2003 file f

Name	Descriptio
Create query in Design view	
Create query by using wizard	
ADs with most indentation	
ADs with most statements	
CodeBlocks (whole ADs only)	
Cross model ADs	
Cross model linkages	
EAB usage	
EAB usage recursive	
Empty Views	Procedure
Function Calls (whole AD)	
Nested Group Views	Procedure
OCX usage	
Transition Effort by Function and Pstep	
Transition Effort by Model	Manual tr
Transition Effort by Pstep	
Unsupported Event Usage	
Unsupported Function Call Counts by Model	
Unsupported Function Calls (whole AD)	
USEd Action Diagrams (all CodeBlocks)	
USEd Action Diagrams (whole AD)	

Action Diagram Analysis

Name: CLIENT_MAINTAIN_DEPARTMENT

Type: PStep

Properties

☑ Display ☐ Screen ☑ Window ☐ Events ☐ Batch

☑ Uses Commands: ADD

☑ Uses Exitstates: DEPARTMENT_CANNOT_BE_DELETED

Entity Actions

☐ Has Entity Actions
☐ READs
☐ Updates (CUD)

Entity Actions (Recursive)

☑ Has Entity Actions
☑ READs
☑ Updates (CUD)

xRxx DIVISION
CRUD DEPARTMENT
xRxx EMPLOYEE

Action Diagram Usage

☑ Uses Action Diagrams

SERVER_DETAIL_DEPARTMENT
SERVER_MAINTAIN_DEPARTMENT

☐ Uses Externals

Action Diagram Usage (Recursive)

☑ Uses Action Diagrams

ADD_DEPARTMENT
DELETE_DEPARTMENT
MODIFY_DEPARTMENT
SERVER_DETAIL_DEPARTMENT

☐ Uses Externals

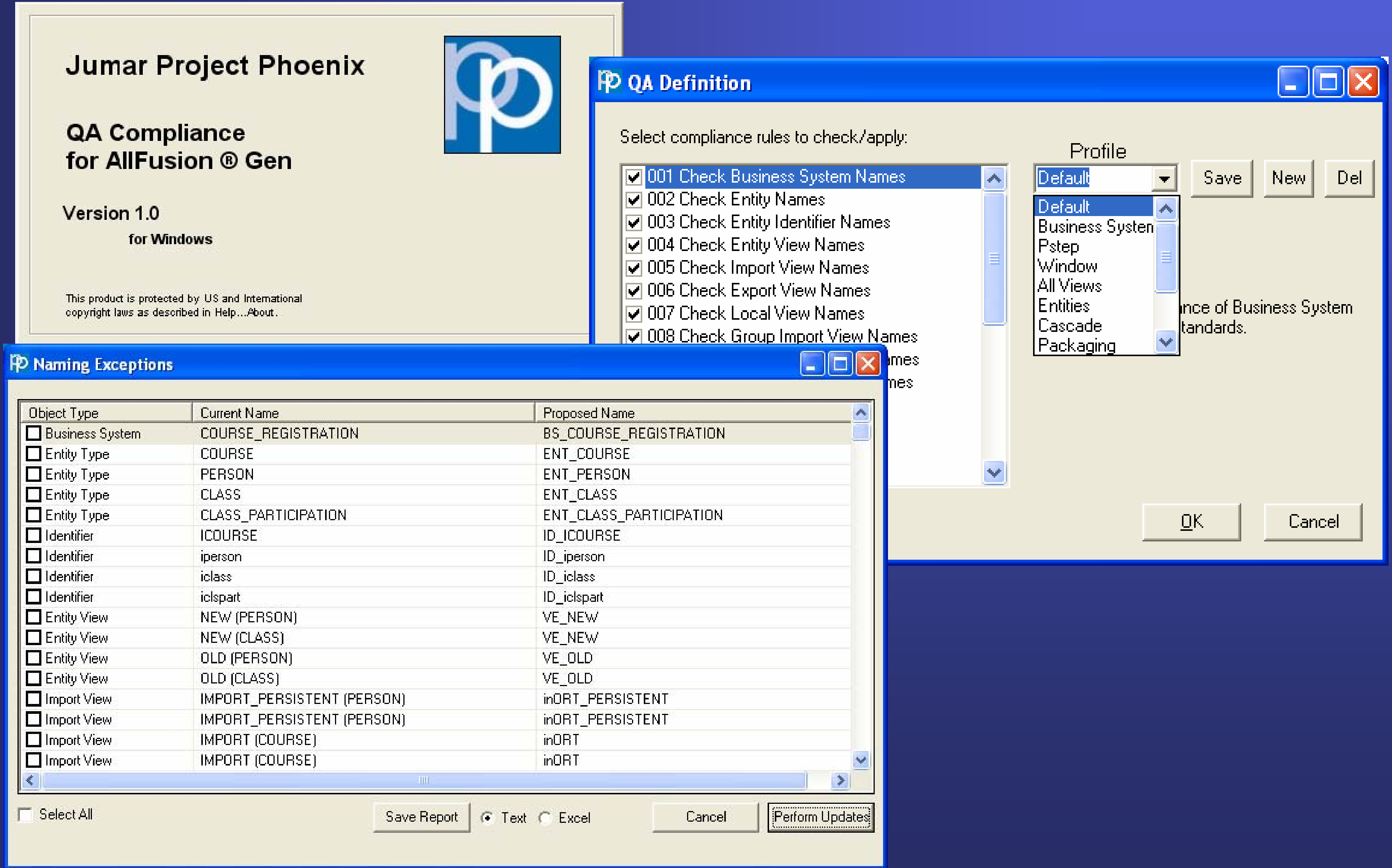
Project Phoenix: QA Analyser
QA Compliance Summary

- Ensures full conformance to naming standards
- Detailed reporting of deviations and (proposed) updates
- Support for multiple Naming Standard Definitions or “Sets” (Regular Expressions)
- Utilisation of Profiles
- Workstation (.DAT) Models (Reporting & Update) and CSE (Reporting only)
- CSE “Batch” Command-Line Interface
- Control over which objects are updated
- Detailed Logging of Updates

QA Analyser: QA Compliance Benefits

- Allows for the rapid identification and correction of incorrectly named Gen objects
- Provides a consistent starting point for subsequent transformation
- Uses client standards - flexible
- Full change control
- Extensive reporting is in standard MS Excel and Word formats
  - Management reporting on the status of the model
  - Detailed reporting of other standards if necessary
  - Detailed logging of the changes made
- Wide-ranging Update functionality
- Supports one off initiative and tidy up, and pro-active, on-going automated QA (for example for code reviews, package delivery, third party governance etc)
- Benefits specifically in reducing ongoing quality checking time and also in applying necessary changes

QA Compliance



Jumar Project Phoenix

QA Compliance
for AllFusion ® Gen

Version 1.0
for Windows

This product is protected by US and International copyright laws as described in Help...About.

QA Definition

Select compliance rules to check/apply:

- ☑ 001 Check Business System Names
- ☑ 002 Check Entity Names
- ☑ 003 Check Entity Identifier Names
- ☑ 004 Check Entity View Names
- ☑ 005 Check Import View Names
- ☑ 006 Check Export View Names
- ☑ 007 Check Local View Names
- ☑ 008 Check Group Import View Names

Profile: Default

- Default
- Business Systen
- Pstep
- Window
- All Views
- Entities
- Cascade
- Packaging

nce of Business System tandards.

Naming Exceptions

Object Type	Current Name	Proposed Name
☐ Business System	COURSE_REGISTRATION	BS_COURSE_REGISTRATION
☐ Entity Type	COURSE	ENT_COURSE
☐ Entity Type	PERSON	ENT_PERSON
☐ Entity Type	CLASS	ENT_CLASS
☐ Entity Type	CLASS_PARTICIPATION	ENT_CLASS_PARTICIPATION
☐ Identifier	ICOURSE	ID_ICOURSE
☐ Identifier	iperson	ID_iperson
☐ Identifier	iclass	ID_iclass
☐ Identifier	iclspart	ID_iclspart
☐ Entity View	NEW (PERSON)	VE_NEW
☐ Entity View	NEW (CLASS)	VE_NEW
☐ Entity View	OLD (PERSON)	VE_OLD
☐ Entity View	OLD (CLASS)	VE_OLD
☐ Import View	IMPORT_PERSISTENT (PERSON)	inORT_PERSISTENT
☐ Import View	IMPORT_PERSISTENT (PERSON)	inORT_PERSISTENT
☐ Import View	IMPORT (COURSE)	inORT
☐ Import View	IMPORT (COURSE)	inORT

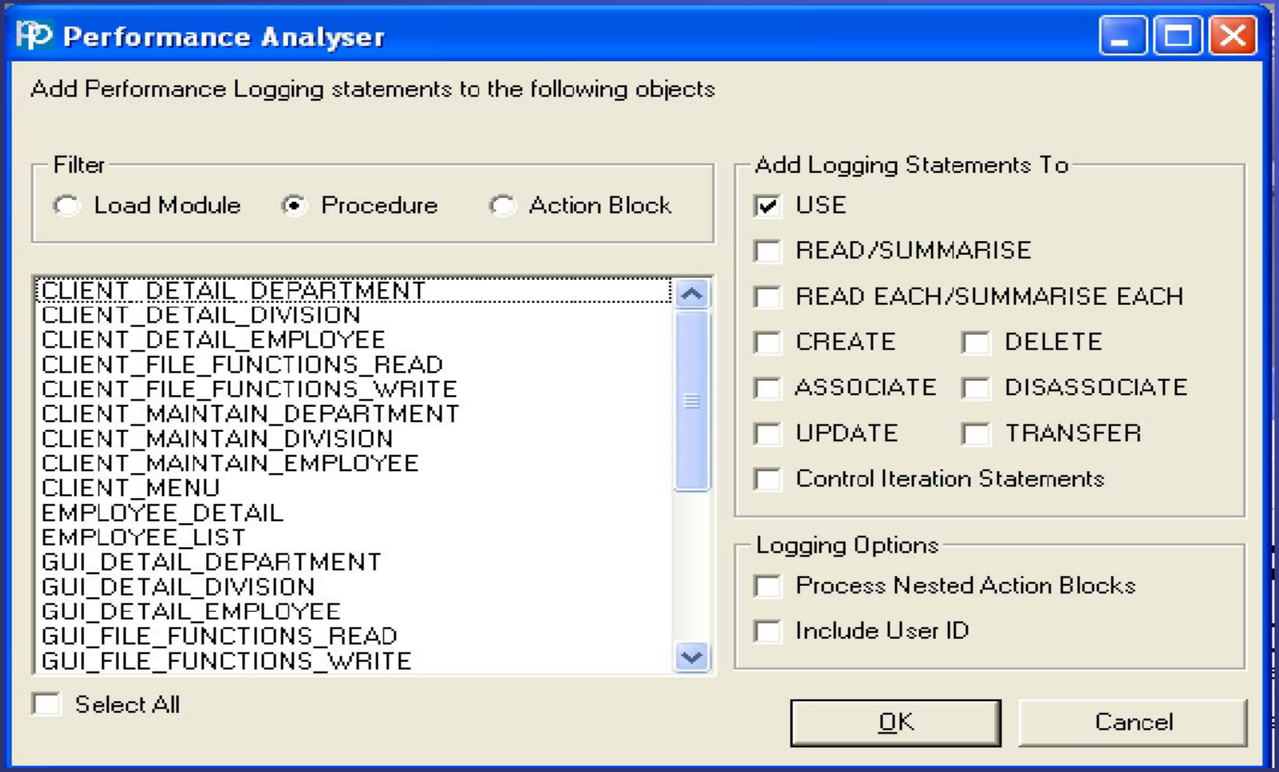
☐ Select All

● Text ○ Excel

Project Phoenix: Performance Analyser for AllFusion® Gen

- Tooling specifically aimed at supporting the performance tuning of Gen applications
- Automates the introduction and subsequent removal of performance measurement points into implementation logic
- Assists with the analysis of Gen models for many factors that influence performance and scalability
- Leaves no trace after clean up
- Can be customised to meet specific customer demands

Project Phoenix:
Performance Analyser for AllFusion® Gen



Performance Analyser

Add Performance Logging statements to the following objects

Filter

○ Load Module ● Procedure ○ Action Block

CLIENT_DETAIL_DEPARTMENT
CLIENT_DETAIL_DIVISION
CLIENT_DETAIL_EMPLOYEE
CLIENT_FILE_FUNCTIONS_READ
CLIENT_FILE_FUNCTIONS_WRITE
CLIENT_MAINTAIN_DEPARTMENT
CLIENT_MAINTAIN_DIVISION
CLIENT_MAINTAIN_EMPLOYEE
CLIENT_MENU
EMPLOYEE_DETAIL
EMPLOYEE_LIST
GUI_DETAIL_DEPARTMENT
GUI_DETAIL_DIVISION
GUI_DETAIL_EMPLOYEE
GUI_FILE_FUNCTIONS_READ
GUI_FILE_FUNCTIONS_WRITE

□ Select All

Add Logging Statements To

☑ USE
□ READ/SUMMARISE
□ READ EACH/SUMMARISE EACH
□ CREATE □ DELETE
□ ASSOCIATE □ DISASSOCIATE
□ UPDATE □ TRANSFER
□ Control Iteration Statements

Logging Options

□ Process Nested Action Blocks
□ Include User ID

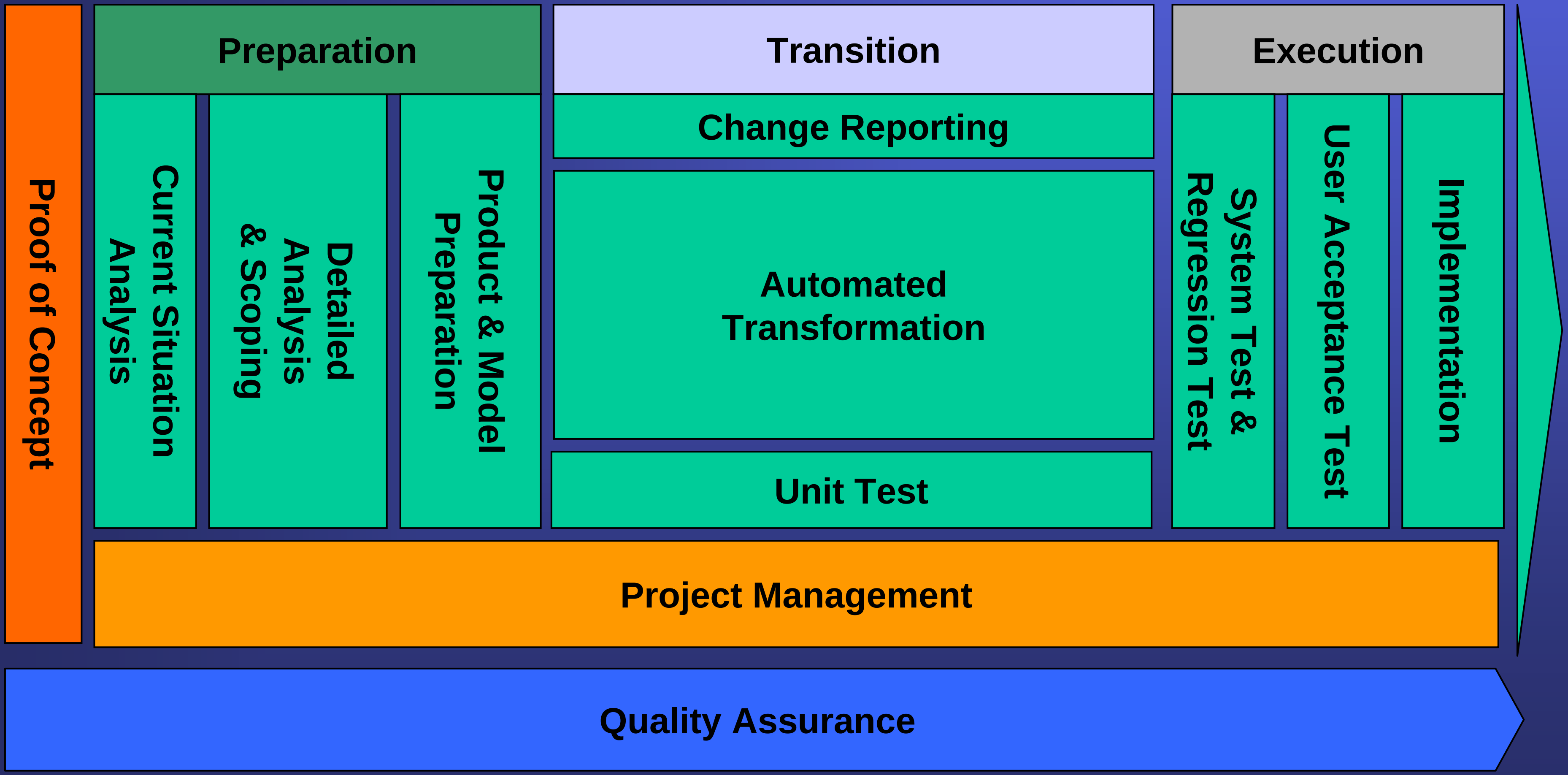
Project Phoenix:
Preparation Phase Summary

- Analysis tools provide useful information even if modernisation is not executed immediately
- Analysis tools Identify specific problem areas that require short and quick bulk update fixes
- Application architecture, DB performance and adherence to standards are all reported in depth
- Provides the deep insight that allows you to make effective cost benefit decisions over what should now be modernised
- Delivers a clear plan and cost prediction so that the business case case can be made for whichever modernisation is required
- Analysis Preparation Phase is executed with you by experienced Jumar consultants who have strong knowledge of analysis processes required and supporting tools

Phoenix Transition Phase
Automation Tooling

- Phoenix Workbench supports each ‘flavour’ of transitioning or combinations
- Batch mode execution of large numbers of objects
- Maximum automation, minimum intervention
- Componentised and configurable to specific needs
- Creates new model(s)
- Full audit and logging facilities
- Actions can be based on quality analysis of current system starting point

Transition Phase
Phoenix Tool Support



Proof of Concept

Preparation

Current Situation Analysis

Detailed Analysis & Scoping

Product & Model Preparation

Transition

Change Reporting

Automated Transformation

Unit Test

Execution

System Test & Regression Test

User Acceptance Test

Implementation

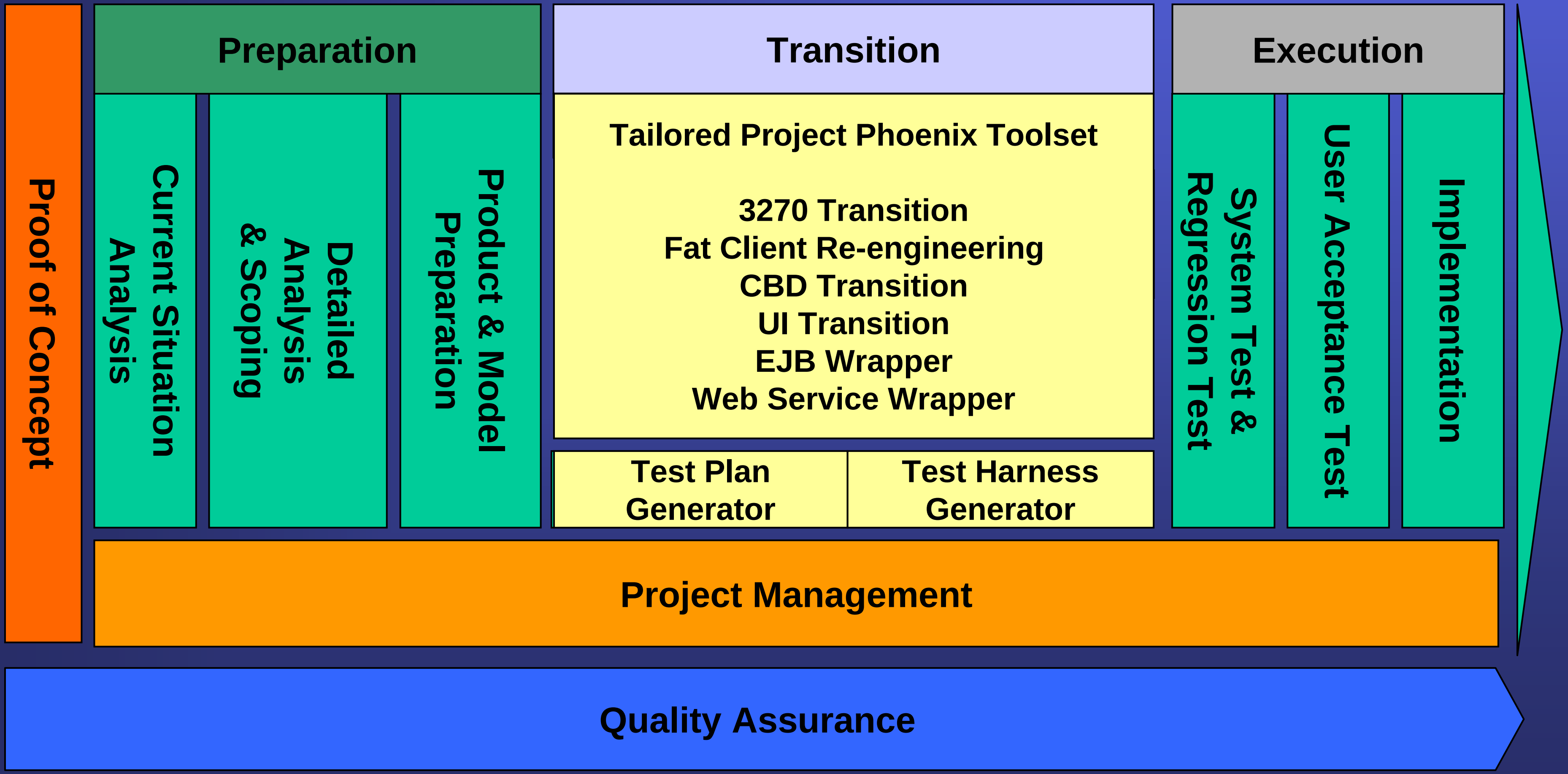
Project Management

Quality Assurance

Transition Tooling Support

- Application Architecture Transformation
  - Block-mode 3270 or Client-Server procedure automated re-engineering to expose legacy business functionality as EJB’s or Web Services under J2EE or .NET
  - Automated restructuring of ‘Fat-Client’ to ‘Server-Centric’ architectures
- User Interface Transformation
  - Automated conversion of 3270 Screens & GUI’s to Web User Interfaces
  - Automated application of new/changed UI standards
- Componentisation
  - Automated conversion of traditional AllFusion Gen models to componentised CS/3.1 compliant or service based (SOA) solutions
- Technology Transitioning
  - Re-platforming AllFusion Gen systems from Mainframe to ‘Downsized’ Platforms. Focus on Non Gen areas

Transition Phase
Phoenix Tool Support



Proof of Concept

Preparation

Current Situation Analysis

Detailed Analysis & Scoping

Product & Model Preparation

Transition

Tailored Project Phoenix Toolset

3270 Transition
Fat Client Re-engineering
CBD Transition
UI Transition
EJB Wrapper
Web Service Wrapper

Test Plan Generator

Test Harness Generator

Execution

System Test & Regression Test

User Acceptance Test

Implementation

Project Management

Quality Assurance

Project Phoenix: Application Architecture Transformation

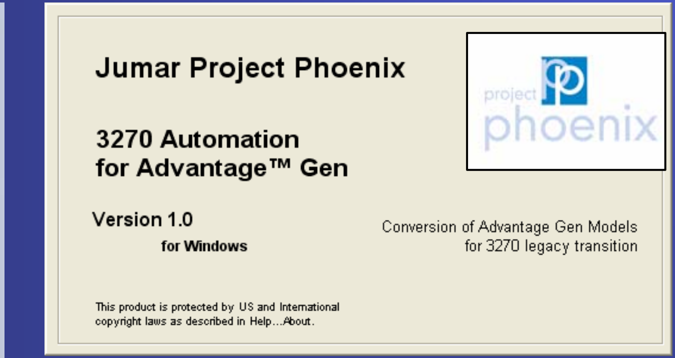
- Rule-based AllFusion Gen model manipulation
- Separation of client and server logic
- Creation of new server procedures from monolithic 3270 procedures
- Creation of corresponding Web user interface from 3270 screen design
- Wrapping of existing business logic Action Blocks into procedure steps for immediate exposure of functionality in new environment
- Migrating thick to thin client as necessary, and according to a pre-defined set of rules based on previous analysis

Project Phoenix

Green Screen Applications

Jumar Solutions' Automated UI & Action Diagram Logic Re-engineering

Web Exposure



Jumar Project Phoenix

3270 Automation
for Advantage™ Gen

Version 1.0
for Windows

project phoenix

Conversion of Advantage Gen Models
for 3270 legacy transition

This product is protected by US and International copyright laws as described in Help...About.

Legacy 3270 IEF, Composer, COOL:Gen, Advantage Gen or AllFusion Gen Applications

Automated Web User Interface Creation & Business Logic Extraction

Web User Interface

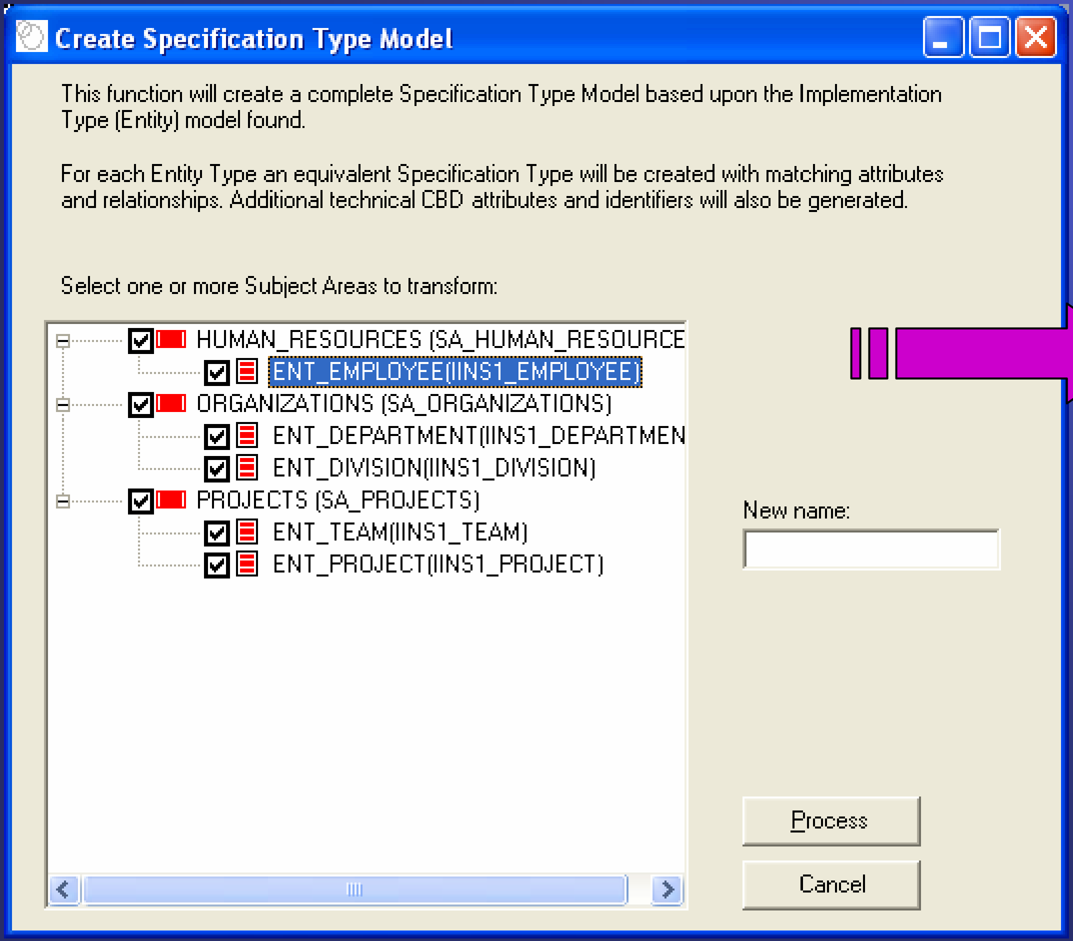
Web Services (implemented in Java/J2EE, or .NET), EJB's etc.

Service Oriented Architectures

Project Phoenix: CBD Transition

- CBD rule-based AllFusion Gen model manipulation
- Performs the activities required to componentise Gen models:
  - Model Analysis for Component Architecture
  - Derivation of Specification Type Model from Implementation (entity) model
  - Generation of translator and mapper action blocks (Specification <=> Implementation)
  - Wrapping of Action Diagrams as CBD Public Operation's
  - Replacement of calls (USEs) in consuming models
  - Data Transition
- Automation applied across multiple models (offered and consumed)
- Customised according to clients view of service based architectures

Project Phoenix:
CBD Transition



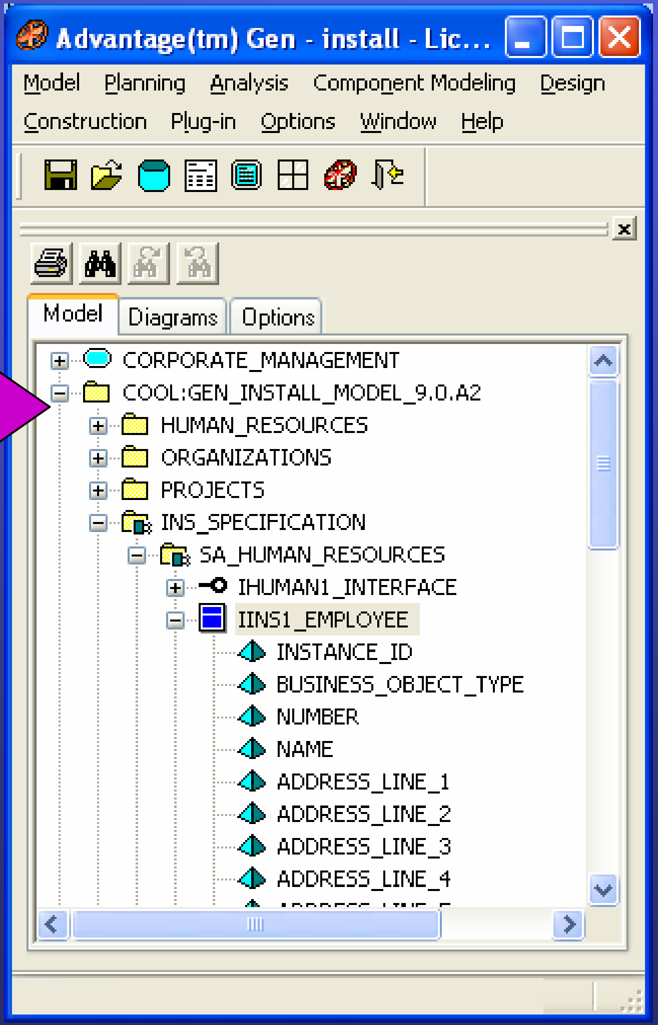
This function will create a complete Specification Type Model based upon the Implementation Type (Entity) model found.

For each Entity Type an equivalent Specification Type will be created with matching attributes and relationships. Additional technical CBD attributes and identifiers will also be generated.

Select one or more Subject Areas to transform:

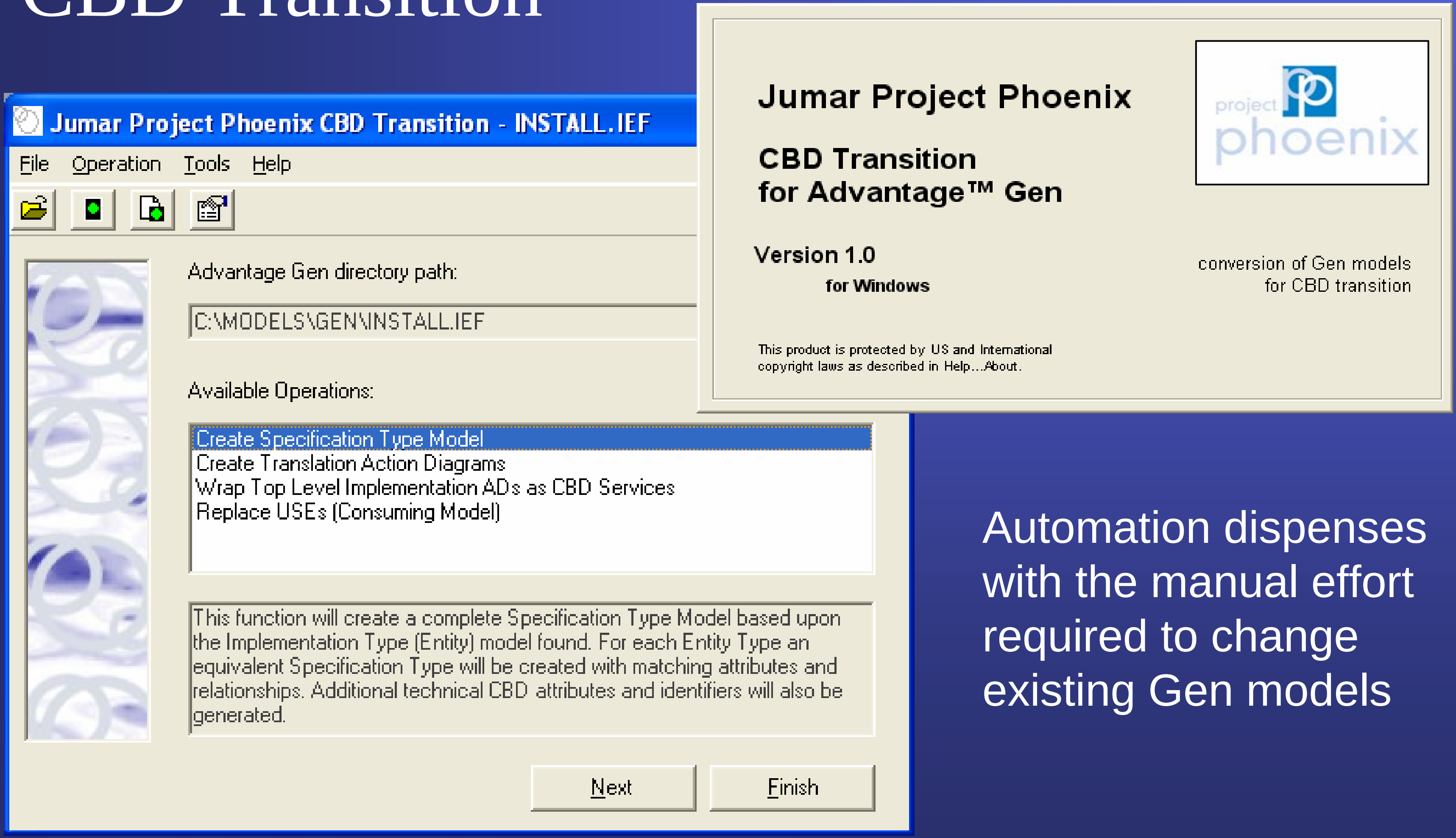
- ☑ HUMAN_RESOURCES (SA_HUMAN_RESOURCE
  - ☑ ENT_EMPLOYEE(IINS1_EMPLOYEE)
- ☑ ORGANIZATIONS (SA_ORGANIZATIONS)
  - ☑ ENT_DEPARTMENT(IINS1_DEPARTMEN
  - ☑ ENT_DIVISION(IINS1_DIVISION)
- ☑ PROJECTS (SA_PROJECTS)
  - ☑ ENT_TEAM(IINS1_TEAM)
  - ☑ ENT_PROJECT(IINS1_PROJECT)

New name:



- CORPORATE_MANAGEMENT
- COOL:GEN_INSTALL_MODEL_9.0.A2
  - HUMAN_RESOURCES
  - ORGANIZATIONS
  - PROJECTS
  - INS_SPECIFICATION
    - SA_HUMAN_RESOURCES
      - IHUMAN1_INTERFACE
      - IINS1_EMPLOYEE
        - INSTANCE_ID
        - BUSINESS_OBJECT_TYPE
        - NUMBER
        - NAME
        - ADDRESS_LINE_1
        - ADDRESS_LINE_2
        - ADDRESS_LINE_3
        - ADDRESS_LINE_4

Project Phoenix: CBD Transition



Jumar Project Phoenix CBD Transition - INSTALL.IEF

Advantage Gen directory path:

C:\MODELS\GEN\INSTALL.IEF

Available Operations:

- Create Specification Type Model
- Create Translation Action Diagrams
- Wrap Top Level Implementation ADs as CBD Services
- Replace USEs (Consuming Model)

This function will create a complete Specification Type Model based upon the Implementation Type (Entity) model found. For each Entity Type an equivalent Specification Type will be created with matching attributes and relationships. Additional technical CBD attributes and identifiers will also be generated.

Jumar Project Phoenix

CBD Transition
for Advantage™ Gen

Version 1.0
for Windows

conversion of Gen models
for CBD transition

This product is protected by US and International copyright laws as described in Help...About.

Automation dispenses with the manual effort required to change existing Gen models

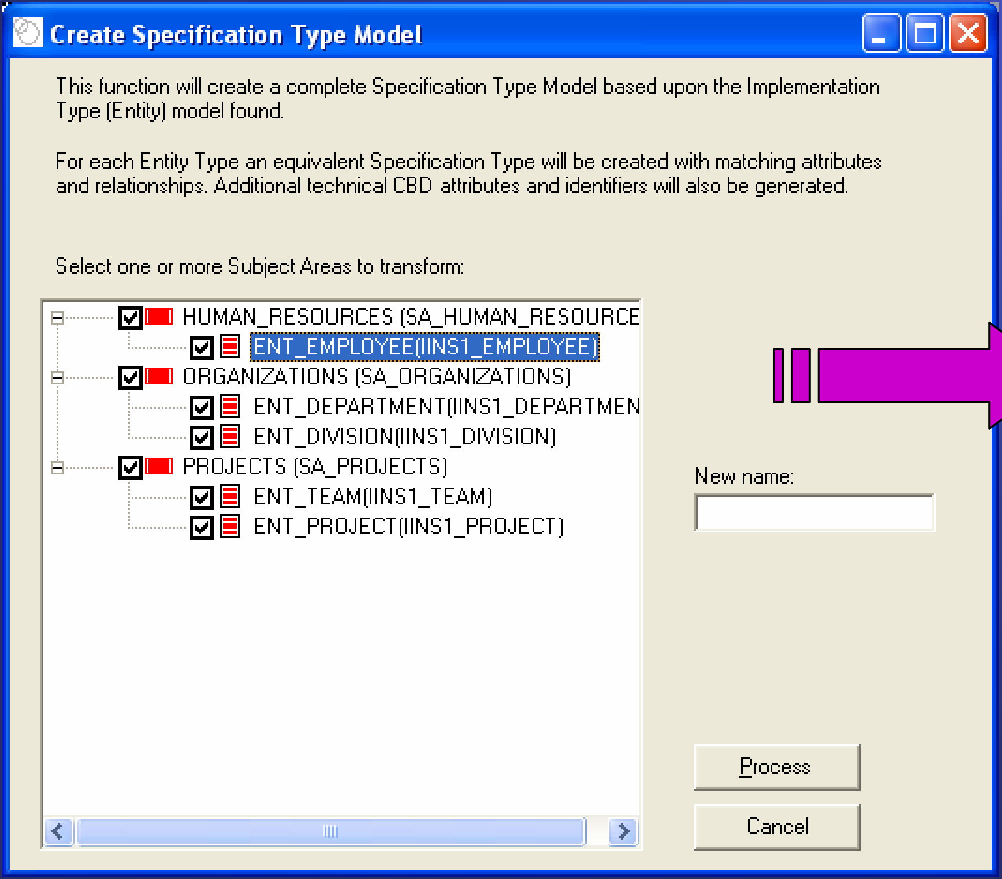
EDGE

project phoenix

CBD Transition

Specification Type Model

Automated Type Model Creation
from existing Entity Model…



Create Specification Type Model

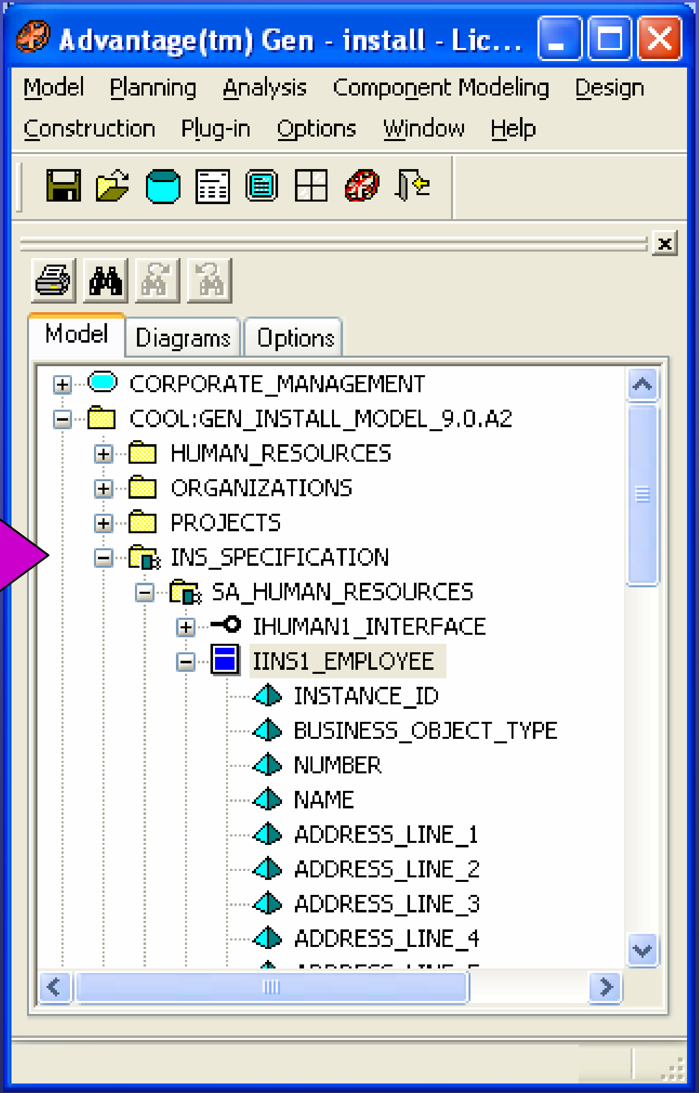
This function will create a complete Specification Type Model based upon the Implementation Type (Entity) model found.

For each Entity Type an equivalent Specification Type will be created with matching attributes and relationships. Additional technical CBD attributes and identifiers will also be generated.

Select one or more Subject Areas to transform:

- ☑ HUMAN_RESOURCES (SA_HUMAN_RESOURCE
  - ☑ ENT_EMPLOYEE(IINS1_EMPLOYEE)
- ☑ ORGANIZATIONS (SA_ORGANIZATIONS)
  - ☑ ENT_DEPARTMENT(IINS1_DEPARTMEN
  - ☑ ENT_DIVISION(IINS1_DIVISION)
- ☑ PROJECTS (SA_PROJECTS)
  - ☑ ENT_TEAM(IINS1_TEAM)
  - ☑ ENT_PROJECT(IINS1_PROJECT)

New name:

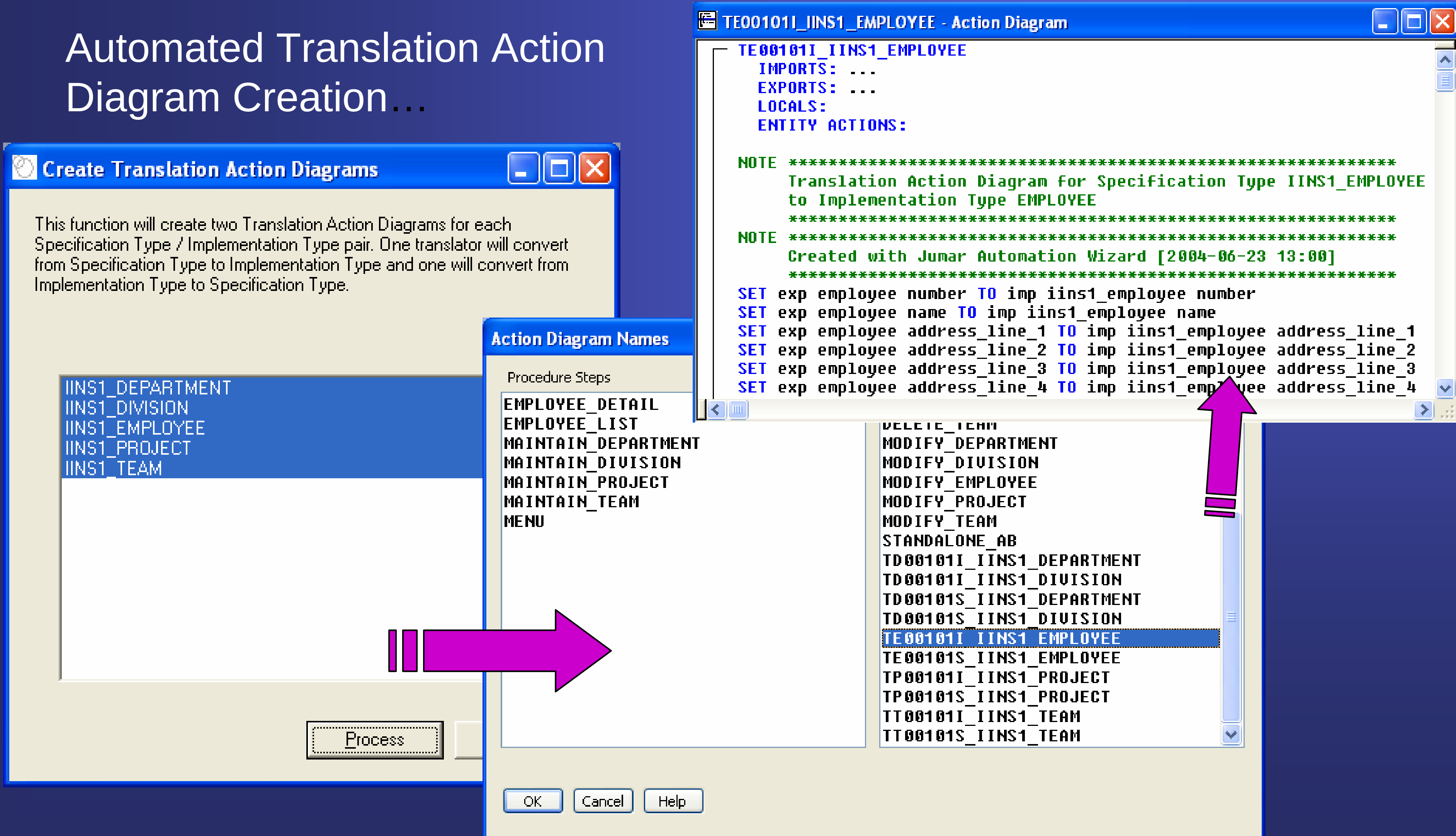


Advantage(tm) Gen - install - Lic…

- CORPORATE_MANAGEMENT
- COOL:GEN_INSTALL_MODEL_9.0.A2
  - HUMAN_RESOURCES
  - ORGANIZATIONS
  - PROJECTS
  - INS_SPECIFICATION
    - SA_HUMAN_RESOURCES
      - IHUMAN1_INTERFACE
      - IINS1_EMPLOYEE
        - INSTANCE_ID
        - BUSINESS_OBJECT_TYPE
        - NUMBER
        - NAME
        - ADDRESS_LINE_1
        - ADDRESS_LINE_2
        - ADDRESS_LINE_3
        - ADDRESS_LINE_4

CBD Transition
Translation Action Diagrams

Automated Translation Action Diagram Creation…



Create Translation Action Diagrams

This function will create two Translation Action Diagrams for each Specification Type / Implementation Type pair. One translator will convert from Specification Type to Implementation Type and one will convert from Implementation Type to Specification Type.

```
IINS1_DEPARTMENT
IINS1_DIVISION
IINS1_EMPLOYEE
IINS1_PROJECT
IINS1_TEAM
```

Action Diagram Names

Procedure Steps

```
EMPLOYEE_DETAIL
EMPLOYEE_LIST
MAINTAIN_DEPARTMENT
MAINTAIN_DIVISION
MAINTAIN_PROJECT
MAINTAIN_TEAM
MENU
```

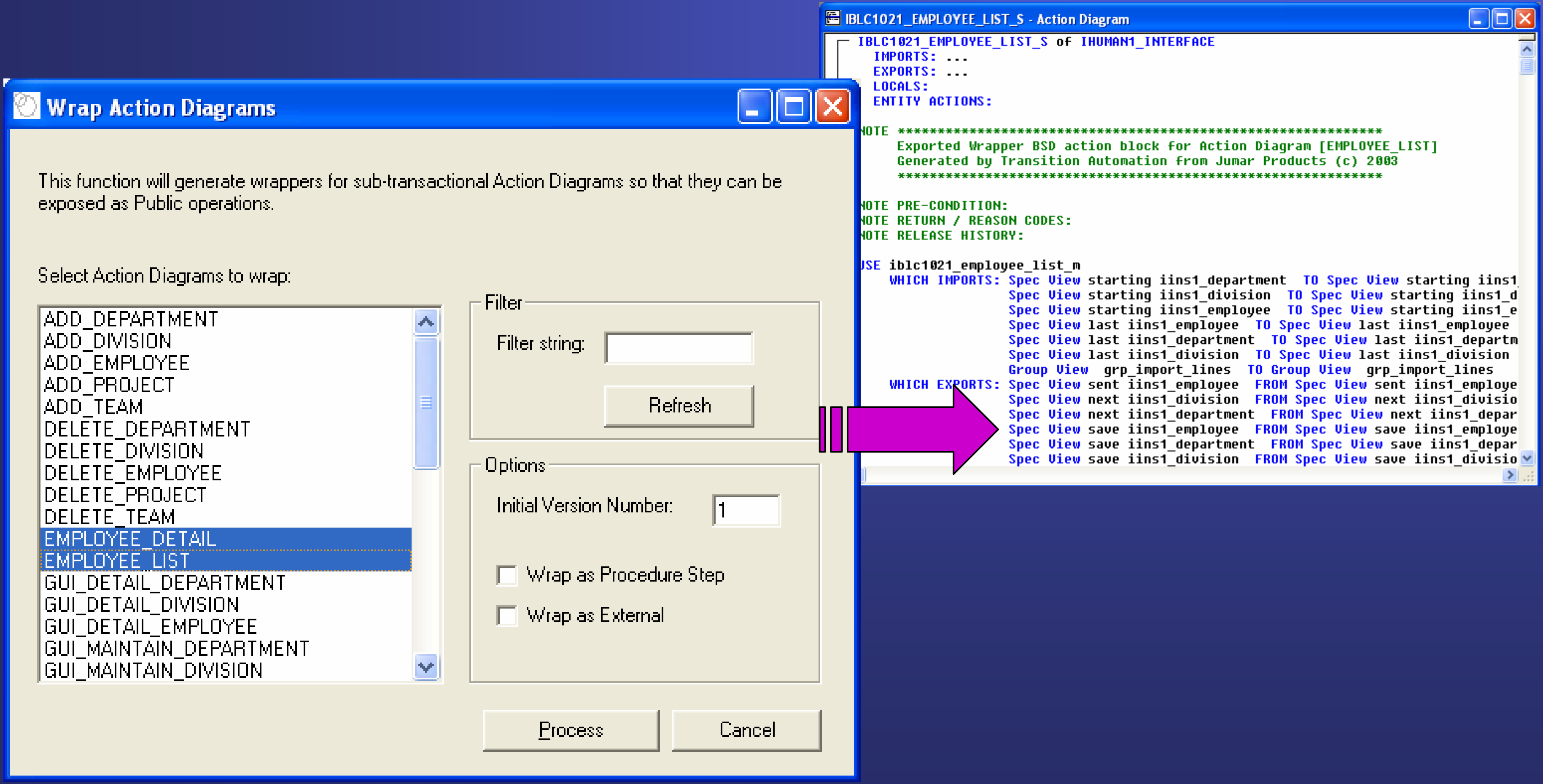
```
DELETE_TEAM
MODIFY_DEPARTMENT
MODIFY_DIVISION
MODIFY_EMPLOYEE
MODIFY_PROJECT
MODIFY_TEAM
STANDALONE_AB
TD00101I_IINS1_DEPARTMENT
TD00101I_IINS1_DIVISION
TD00101S_IINS1_DEPARTMENT
TD00101S_IINS1_DIVISION
TE00101I_IINS1_EMPLOYEE
TE00101S_IINS1_EMPLOYEE
TP00101I_IINS1_PROJECT
TP00101S_IINS1_PROJECT
TT00101I_IINS1_TEAM
TT00101S_IINS1_TEAM
```

TE00101I_IINS1_EMPLOYEE - Action Diagram

```
TE00101I_IINS1_EMPLOYEE
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS:

NOTE ************************************************************
     Translation Action Diagram for Specification Type IINS1_EMPLOYEE
     to Implementation Type EMPLOYEE
     ************************************************************
NOTE ************************************************************
     Created with Jumar Automation Wizard [2004-06-23 13:00]
     ************************************************************
SET exp employee number TO imp iins1_employee number
SET exp employee name TO imp iins1_employee name
SET exp employee address_line_1 TO imp iins1_employee address_line_1
SET exp employee address_line_2 TO imp iins1_employee address_line_2
SET exp employee address_line_3 TO imp iins1_employee address_line_3
SET exp employee address_line_4 TO imp iins1_employee address_line_4
```

CBD Transition
Public Operations

Automated Public Operation Creation…



Wrap Action Diagrams

This function will generate wrappers for sub-transactional Action Diagrams so that they can be exposed as Public operations.

Select Action Diagrams to wrap:

- ADD_DEPARTMENT
- ADD_DIVISION
- ADD_EMPLOYEE
- ADD_PROJECT
- ADD_TEAM
- DELETE_DEPARTMENT
- DELETE_DIVISION
- DELETE_EMPLOYEE
- DELETE_PROJECT
- DELETE_TEAM
- EMPLOYEE_DETAIL
- EMPLOYEE_LIST
- GUI_DETAIL_DEPARTMENT
- GUI_DETAIL_DIVISION
- GUI_DETAIL_EMPLOYEE
- GUI_MAINTAIN_DEPARTMENT
- GUI_MAINTAIN_DIVISION

Filter

Filter string:

Refresh

Options

Initial Version Number: 1

□ Wrap as Procedure Step

□ Wrap as External

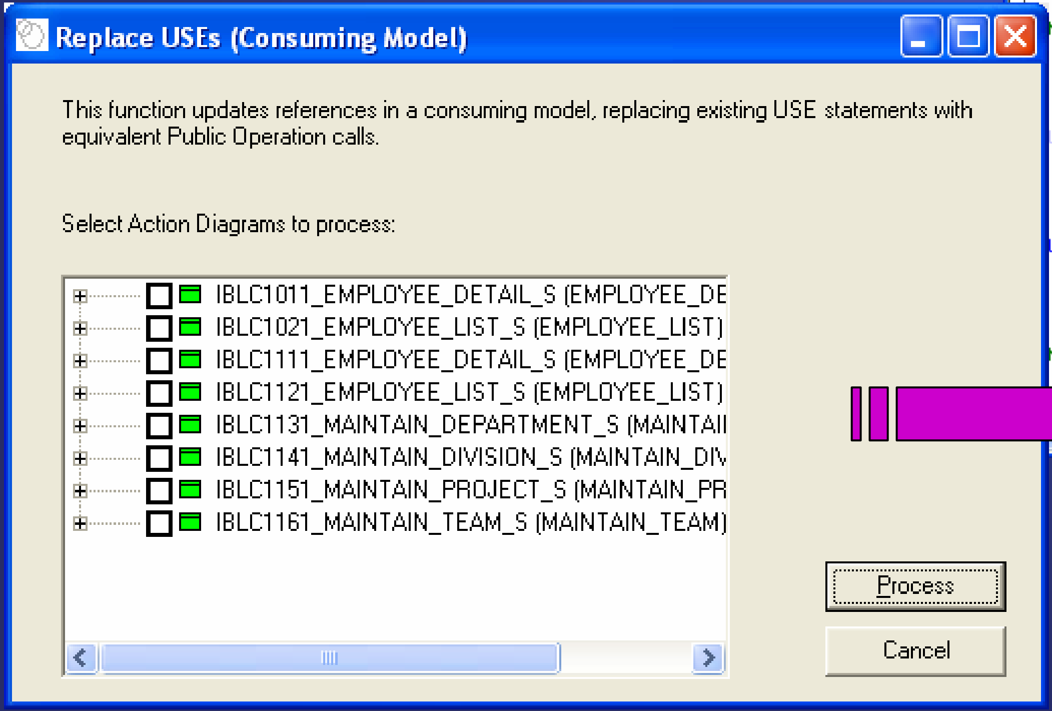
Process   Cancel

IBLC1021_EMPLOYEE_LIST_S - Action Diagram

```
┌ IBLC1021_EMPLOYEE_LIST_S of IHUMAN1_INTERFACE
│   IMPORTS: ...
│   EXPORTS: ...
│   LOCALS:
│   ENTITY ACTIONS:

NOTE ************************************************************
     Exported Wrapper BSD action block for Action Diagram [EMPLOYEE_LIST]
     Generated by Transition Automation from Jumar Products (c) 2003
     ************************************************************

NOTE PRE-CONDITION:
NOTE RETURN / REASON CODES:
NOTE RELEASE HISTORY:

USE iblc1021_employee_list_m
    WHICH IMPORTS: Spec View starting iins1_department  TO Spec View starting iins1
                   Spec View starting iins1_division  TO Spec View starting iins1_d
                   Spec View starting iins1_employee  TO Spec View starting iins1_e
                   Spec View last iins1_employee  TO Spec View last iins1_employee
                   Spec View last iins1_department  TO Spec View last iins1_departm
                   Spec View last iins1_division  TO Spec View last iins1_division
                   Group View  grp_import_lines  TO Group View  grp_import_lines
    WHICH EXPORTS: Spec View sent iins1_employee  FROM Spec View sent iins1_employe
                   Spec View next iins1_division  FROM Spec View next iins1_divisio
                   Spec View next iins1_department  FROM Spec View next iins1_depar
                   Spec View save iins1_employee  FROM Spec View save iins1_employe
                   Spec View save iins1_department  FROM Spec View save iins1_depar
                   Spec View save iins1_division  FROM Spec View save iins1_divisio
```

CBD Transition
Consumer Substitution

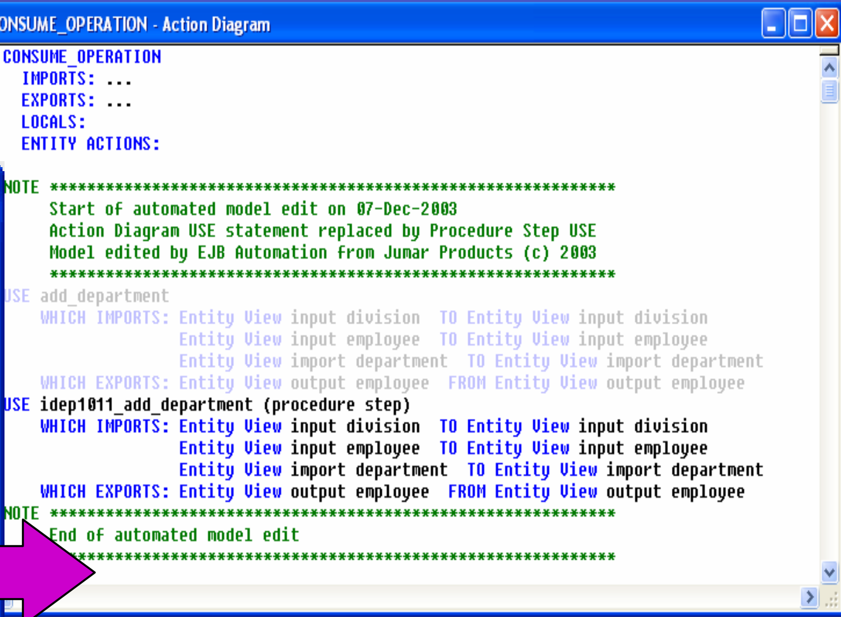
Automated Public Operation USE Substitution…



This function updates references in a consuming model, replacing existing USE statements with equivalent Public Operation calls.

Select Action Diagrams to process:

- □ 🟩 IBLC1011_EMPLOYEE_DETAIL_S (EMPLOYEE_DE
- □ 🟩 IBLC1021_EMPLOYEE_LIST_S (EMPLOYEE_LIST)
- □ 🟩 IBLC1111_EMPLOYEE_DETAIL_S (EMPLOYEE_DE
- □ 🟩 IBLC1121_EMPLOYEE_LIST_S (EMPLOYEE_LIST)
- □ 🟩 IBLC1131_MAINTAIN_DEPARTMENT_S (MAINTAI
- □ 🟩 IBLC1141_MAINTAIN_DIVISION_S (MAINTAIN_DIV
- □ 🟩 IBLC1151_MAINTAIN_PROJECT_S (MAINTAIN_PR
- □ 🟩 IBLC1161_MAINTAIN_TEAM_S (MAINTAIN_TEAM)



```
CONSUME_OPERATION
  IMPORTS: ...
  EXPORTS: ...
  LOCALS:
  ENTITY ACTIONS:

NOTE ************************************************************
     Start of automated model edit on 07-Dec-2003
     Action Diagram USE statement replaced by Procedure Step USE
     Model edited by EJB Automation from Jumar Products (c) 2003
     ************************************************************
USE add_department
    WHICH IMPORTS: Entity View input division  TO Entity View input division
                   Entity View input employee  TO Entity View input employee
                   Entity View import department  TO Entity View import department
    WHICH EXPORTS: Entity View output employee  FROM Entity View output employee
USE idep1011_add_department (procedure step)
    WHICH IMPORTS: Entity View input division  TO Entity View input division
                   Entity View input employee  TO Entity View input employee
                   Entity View import department  TO Entity View import department
    WHICH EXPORTS: Entity View output employee  FROM Entity View output employee
NOTE ************************************************************
     End of automated model edit
     ************************************************************
```

Customer Example 1

Retail Organisation - Application Architecture Transformation and CBD

- Creation of a componentised application and ‘thick to thin client’ procedure conversion
- Key requirements
  - Rules in thick client moved to server - more available, reusable
  - More efficient maintenance following componentisation
  - Cross enterprise reuse of system assets
- Results
  - Analysis of current system to identifying transitioning possible
  - New component architecture
  - Tailored Phoenix and Jumar:Xtras tools
  - Proof of concept and pilot before the full rollout
  - Transition over next 12 months
- Jumar’s role
  - Tailored Phoenix automation, architecture, consultancy services, education

Customer Case Study 2

Financial Systems Integrator –CBD SOA Transition.

- Creation of a componentised, interface and service based architecture
- Key requirements
  - Coexist with business as usual changes
  - Re-factoring with speed – disruption minimised, benefits faster
- Results
  - Staged transition with Phoenix automation at its core
  - Phoenix tailored to exact clients needs
  - Previously closed business rule assets exposed for Gen and non Gen reuse
  - Duplication minimised, standards enforced such that application is more modular and maintainable
- Jumar’s role
  - Five year Plan -Phoenix automation, consultancy, services, education

Project Phoenix:
CBD/SOA Transition Benefits Summary

- Experience the full benefits of CBD and SOA:
  - Re-use
  - Flexibility
  - Maintainability
- Project Phoenix CBD Automation automates the activities necessary to transition AllFusion Gen models to CBD and SOA architectures.
- Makes the transition cost effective and feasible with very high level of automation
- Automated process produces consistent and maintainable components that meet your business requirements

Project Phoenix:
UI Transition

- 3270 green screen and GUI into WUI
- Mapping rules are critical – bespoke
- Tailored UI standards definition
- Conversion in bulk takes seconds per UI
- Saving a man day effort average per transaction

Project Phoenix: Technology Transition

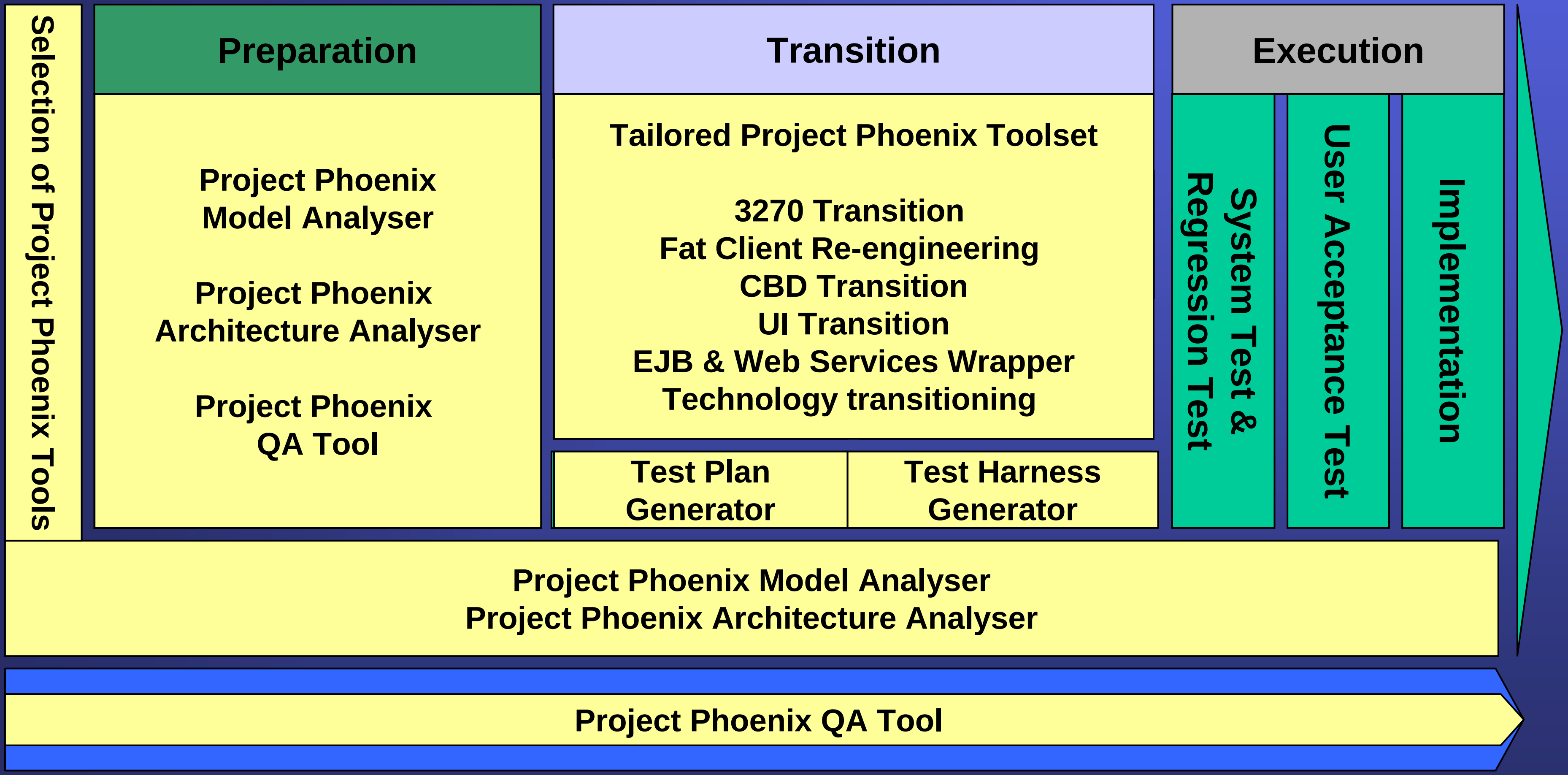
- Technology downsizing (e.g. Mainframe to UNIX transition)
- Gen deals well with most of Gen code transition
- Notable exceptions such as ‘reserved words’ and DBMS features
- Non Gen objects are largest part of job
  - External action blocks
  - JCL to Unix scripts etc…

Customer Example 3

Telecommunications company

- Support for re-platforming and technology downsizing
- conversion of COBOL externals generated into C
- Key requirements
  - Speed
  - quality
- Results
  - Converted externals in C developed faster and in less cost than rebuilding manually
  - Risk of errors reduced using automation
- Jumar’s role
  - Outsource execution of EAB conversion. Web Pilot, transitioning consultancy

Project Phoenix:
Transition, Testing



Selection of Project Phoenix Tools

Preparation

Project Phoenix
Model Analyser

Project Phoenix
Architecture Analyser

Project Phoenix
QA Tool

Transition

Tailored Project Phoenix Toolset

3270 Transition
Fat Client Re-engineering
CBD Transition
UI Transition
EJB & Web Services Wrapper
Technology transitioning

Test Plan
Generator

Test Harness
Generator

Execution

System Test &
Regression Test

User Acceptance Test

Implementation

Project Phoenix Model Analyser
Project Phoenix Architecture Analyser

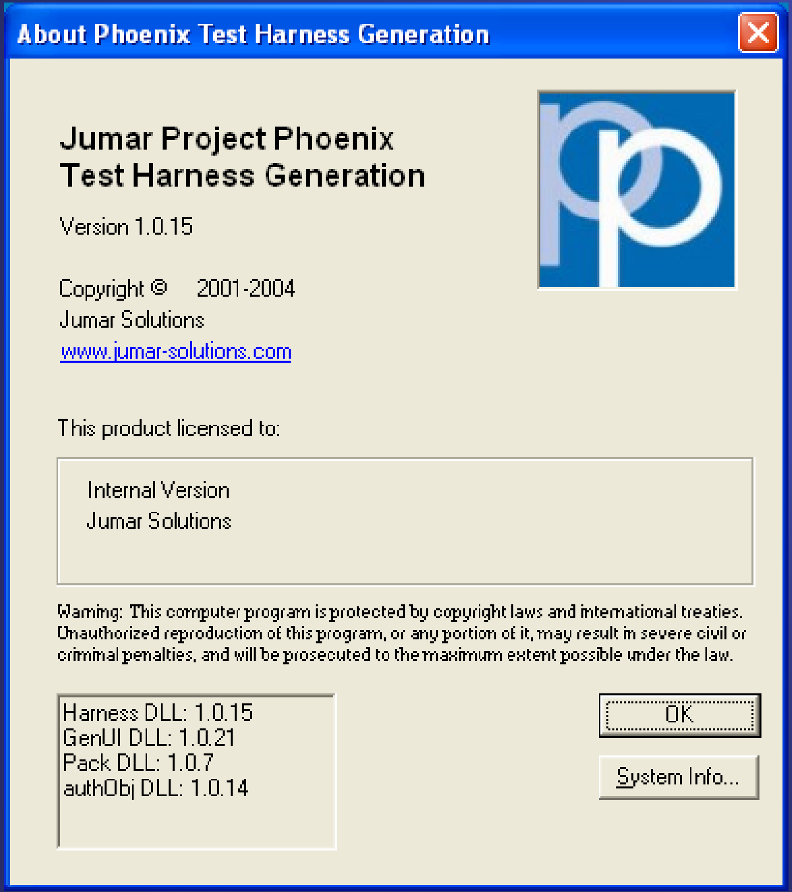
Project Phoenix QA Tool

Plus Jumar:Xtras Products including BulkUpdate and ModelReporter

Project Phoenix:
Test Harness Generation

- Creates test harnesses quickly and accurately
- Can be used for Action Blocks and Procedure Steps
- Bulk creation possible
- Option to pre-populate data needed for testing
- Window or Co-operative packaging
- Objects created conform to naming standards

Project Phoenix:
Test Harness Generation



About Phoenix Test Harness Generation

Jumar Project Phoenix
Test Harness Generation

Version 1.0.15

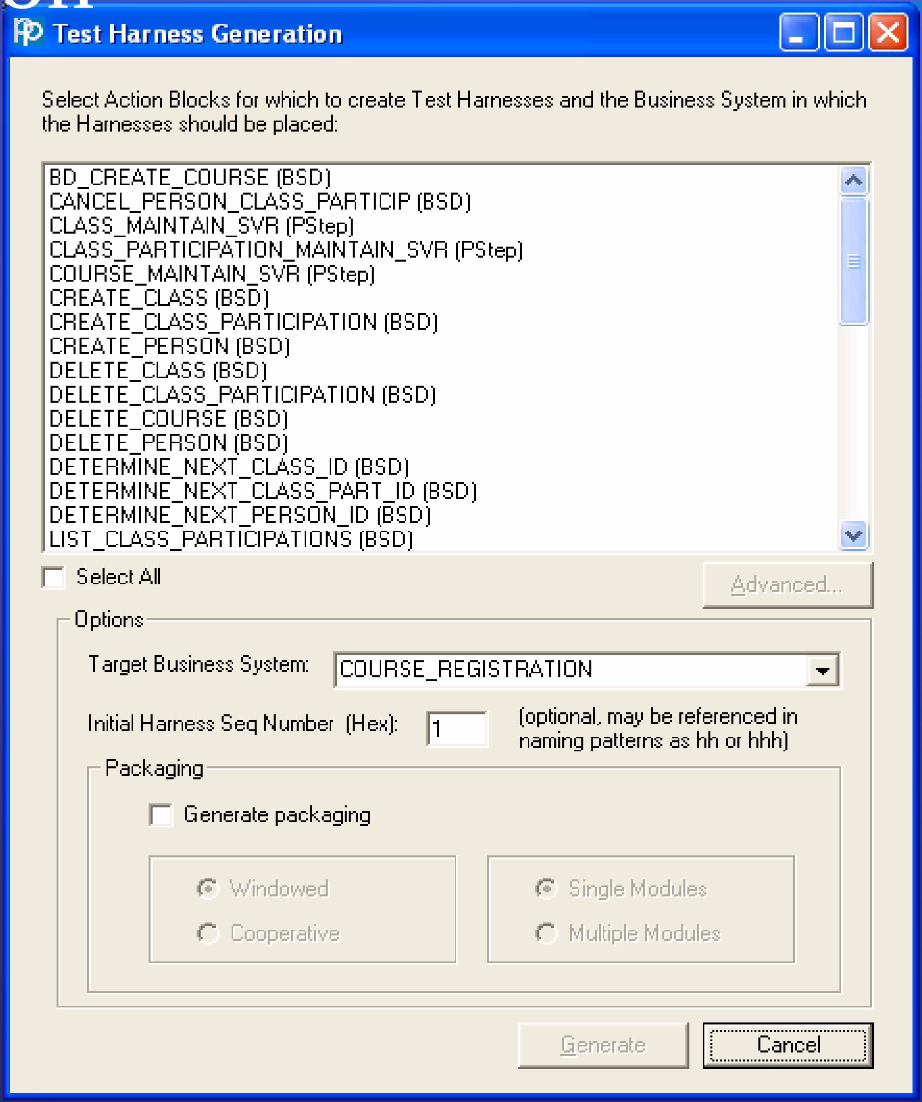
Copyright © 2001-2004
Jumar Solutions
www.jumar-solutions.com

This product licensed to:

Internal Version
Jumar Solutions

Warning: This computer program is protected by copyright laws and international treaties. Unauthorized reproduction of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

```
Harness DLL: 1.0.15
GenUI DLL: 1.0.21
Pack DLL: 1.0.7
authObj DLL: 1.0.14
```



Test Harness Generation

Select Action Blocks for which to create Test Harnesses and the Business System in which the Harnesses should be placed:

```
BD_CREATE_COURSE (BSD)
CANCEL_PERSON_CLASS_PARTICIP (BSD)
CLASS_MAINTAIN_SVR (PStep)
CLASS_PARTICIPATION_MAINTAIN_SVR (PStep)
COURSE_MAINTAIN_SVR (PStep)
CREATE_CLASS (BSD)
CREATE_CLASS_PARTICIPATION (BSD)
CREATE_PERSON (BSD)
DELETE_CLASS (BSD)
DELETE_CLASS_PARTICIPATION (BSD)
DELETE_COURSE (BSD)
DELETE_PERSON (BSD)
DETERMINE_NEXT_CLASS_ID (BSD)
DETERMINE_NEXT_CLASS_PART_ID (BSD)
DETERMINE_NEXT_PERSON_ID (BSD)
LIST_CLASS_PARTICIPATIONS (BSD)
```

□ Select All

Options

Target Business System: COURSE_REGISTRATION

Initial Harness Seq Number (Hex): 1 (optional, may be referenced in naming patterns as hh or hhh)

Packaging

□ Generate packaging

● Windowed
○ Cooperative

● Single Modules
○ Multiple Modules

Project Phoenix:
Test Harness Generation
What is created?

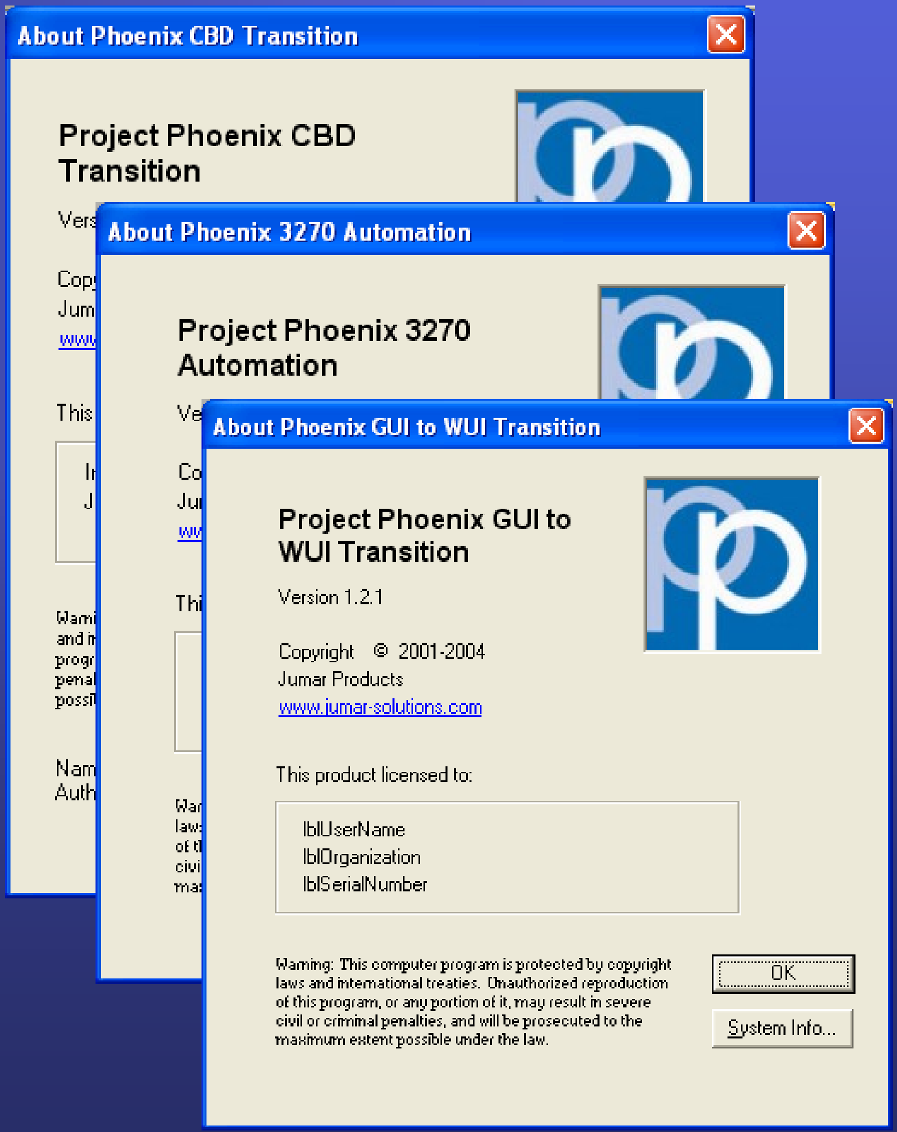
- For an Action Block the tool creates:
  - A server procedure step which calls that Action Block
  - A client procedure step which presents a window for data entry
- For a Server Procedure step the tool creates:
  - A client procedure step which presents a window for data entry

Project Phoenix: Test Harness Benefits

- Provides a consistent interface for unit testing of all Action Blocks and Procedure Steps – supports quality in maintenance and regression testing
- Quick
  - Takes minutes rather than hours (1-2 hour effort saving each).
  - Bulk creation
- Automation ensures it is standard
  - No time spent making it look better than needed
  - Complies to naming standards
- Error Free
  - Takes the views from the tested AB or PStep

Phoenix Automation Capabilities

- Model Analysis
- Technology downsizing
- 3270 Transition
- CBD Transition
- GUI to WUI Transition
- Fat Client Re-engineering
- QA Compliance
- EJB Wrapper
- Action Block Promotion
- Test Harness Generator



About Phoenix CBD Transition

Project Phoenix CBD Transition

About Phoenix 3270 Automation

Project Phoenix 3270 Automation

About Phoenix GUI to WUI Transition

Project Phoenix GUI to WUI Transition

Version 1.2.1

Copyright © 2001-2004
Jumar Products
www.jumar-solutions.com

This product licensed to:

lblUserName
lblOrganization
lblSerialNumber

Warning: This computer program is protected by copyright laws and international treaties. Unauthorized reproduction of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Project Phoenix:
In Summary

Project Phoenix is a unique combination of consultancy and flexible automation tools that enable:

- Effective re-use of existing business rules and code
- Exposure of previously hidden code for re-use and integration
- Cost and time effective re-architecting and modernisation achieved via automation
- Quicker delivery for business advantage
- Re-architecting to become a realistic, viable alternative to
  - Offshore option
  - Purchasing replacement packages
  - Rebuilding from scratch
- Supportive of the ‘Asset’ based approach to IT development and maintenance

Project Phoenix: Further Information

Project Phoenix encompasses a huge range of automation solutions:

- More detailed and focused demonstrations are available on request
- One day workshop to prove the effectiveness and applicability of the Project Phoenix approach in your own environment
- Model Quality Review: For a detailed QA & Analysis report of your current Gen Model portfolio

www.jumar-solutions.com

dave.tomkins@jumar-solutions.com

Discussion/Questions

jumar
360° business
and IT solutions

www.jumar-solutions.com
info@jumar-solutions.com
+44 121 788 4550

ca smart™

ca
Computer Associates®
SOLUTION PARTNER

Deloitte.
Technology Fast50
Winner 2005

CNETNETWORKS™ UK
technology
awards
FINALIST2005