

## ***Gargen, the G&R component generator***

### **What is Gargen?**

Gargen is a tool that allows you to step through a screen driven function of a legacy application and then generate a component that replicates that function whenever called. The function can consist of one or many interactions with the legacy system. The component can have one or many input parameters for screen input fields, and can collect output from one or more areas in one or more output screens.

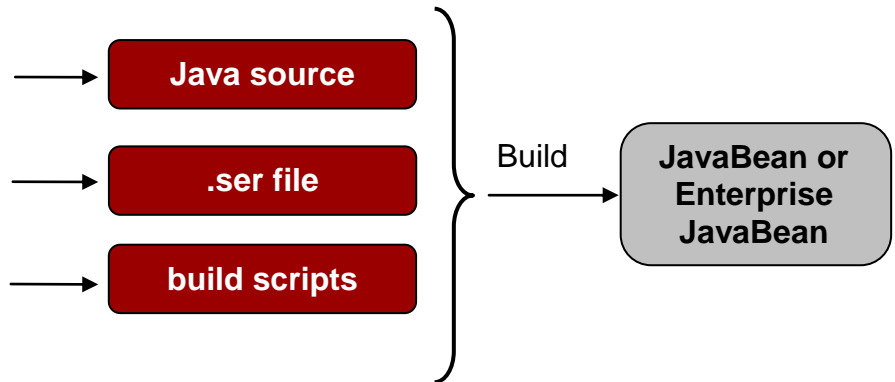
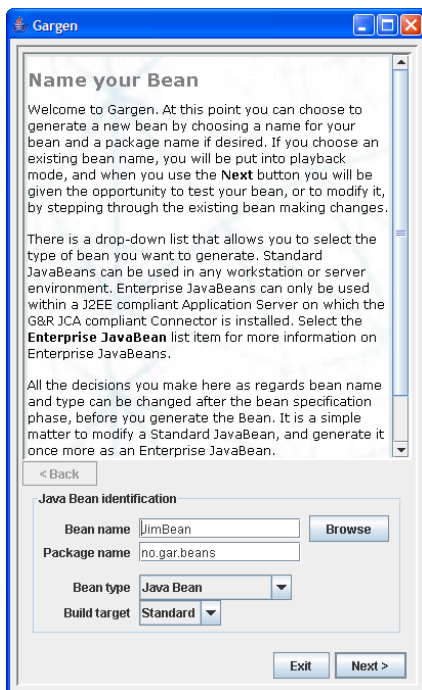
- No changes at all in the mainframe production system
- Submit real time transactions to CICS, IMS, TP8 or TDS
- Receive responses speedily and reliably in real time
- IBM and Bull mainframes
- IBM3270, IBM5250, Bull VIP, Bull DKU terminal emulations
- Components for Java applications, applets, Servlets, JSP, J2EE, Web services

### **Why do you need Gargen?**

- A large part of your Enterprise data is managed by legacy systems that you need to keep running smoothly for business continuity
- You need to streamline development of new services in a service-oriented architecture on standardized platforms
- You need to make your Enterprise data available to these new services, without touching a line of code in the legacy systems

Programming of JavaBeans for communication with legacy systems can be time-consuming because of the trivial programmatical details. You must check that your bean has received the expected screen, select the data on the screen that you would like to be an output from your bean and identify the input that you want to enter from the input parameters to your bean.

You can automate programming of these communication JavaBeans using Gargen.



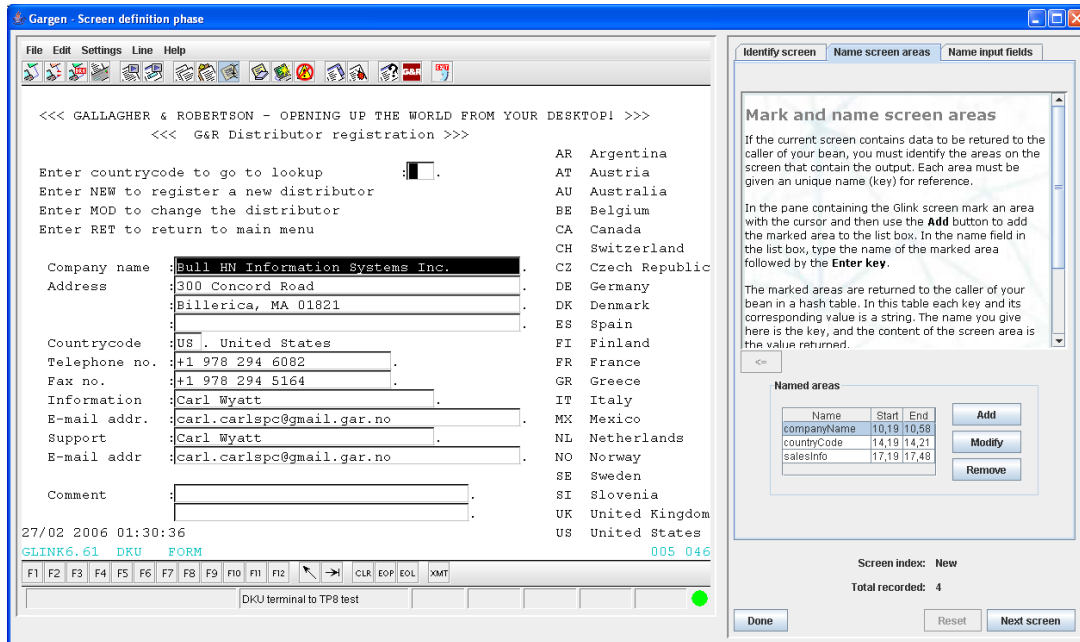
### **Build script for a standard JavaBean for use with**

- Java application
- Servlet

### **or for deploying your Enterprise JavaBean on:**

- Sun reference model
- ObjectWeb JOnAS 4.1.4
- BEA WebLogic Server 7.0, 8.1
- IBM WebSphere 5.1
- Oracle AS 10g

## Component generation made easy



### Define your JavaBean

- From Gargen logon to the target application
- Gargen starts Glink for Java, and you see the target application in an emulation window
- Step through the screens that provide the function you want to encapsulate
- Mark areas that uniquely identify each screen
- Mark input fields that your JavaBean will need to set when duplicating the function
- Mark output areas that your JavaBean will return as output
- Save your work

### Test your JavaBean from within Gargen

- See the application screens as your bean supplies input and collects output
- At the end of the test Gargen displays the output from your bean

### Generate your JavaBean

- Generate a standard JavaBean for use with Java applications or Java applets
- Generate a standard Java bean for use in web applications using Glink Enterprise Edition as a framework for Servlets or Java Server Pages
- Use standard JavaBeans as above together with GCPM, the G&R Connection Pool manager, included in Glink for Java Enterprise Edition
- Generate an Enterprise JavaBean for use with Bull HooX in the J2EE environment of your choice

### Deploy your JavaBean

- Follow the instructions we provide for deploying standard JavaBeans in non-J2EE environments
- Use the build scripts we provide for deploying Enterprise JavaBeans in J2EE environments

You use **Glink for Java Professional Edition** as a framework for developing workstation applications using Gargen JavaBeans

You use **Glink for Java Enterprise Edition** as a framework for developing web applications implemented as Servlets, using Gargen JavaBeans for host communication

You use **GCPM**, the G&R Connection Pool manager included in Glink Enterprise Edition to provide your Gargen JavaBeans with facilities for session pool management

You use **Bull HooX** in the J2EE environment to provide your Gargen Enterprise JavaBeans with a JCA compliant adapter for standardized communication with the mainframe applications