

**BUILDING FLEXIBLE
ENTERPRISE PROCESSES
USING BUSINESS RULES
AND BPEL PROCESS
MANAGER**

*An Oracle White Paper
Jan 2005*

BUILDING FLEXIBLE ENTERPRISE PROCESSES USING BUSINESS RULES AND BPEL PROCESS MANAGER

I. INTRODUCTION

Efficient business processes are one of the main competitive differentiators for any successful company. These processes orchestrate interactions between systems, services, people and partners to achieve key strategic and operational objectives. The definition and flawless execution of business processes enable an organization to provide quality products and services, reduce costs, improve customer service and react quickly to changing market conditions.

A typical business process often includes a number of decision points. These decision points generally have an effect on the process flow or the resources involved and in some cases also validate incoming data. These are evaluated based on certain conditions and facts internal or external to the process and predefined company policies or rules. In many organizations, business analysts or a central policy committee defines these rules, but then the actual rule evaluation is embedded in the process logic itself. This typically results in inconsistent application of these rules across processes and also makes it difficult to propagate changes to all processes when company policies are revised.

The desire to have adaptive business processes that can quickly respond to changing business environments, government regulations and competitive pressures has resulted in the need for a Business Rules Engine as a core component of Business Process Management (BPM) systems. Before going into the details of business rules, let's define what business rules mean and when they should be considered:

According to the Business rules group¹, business rules can be defined in one of two ways – (a) from a business analyst's perspective, a business rule is "guidance that there is an obligation concerning conduct, action, practice, or procedure within a particular activity or sphere" (b) from an information system perspective a business rule is "a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business". A business rules product includes a declarative rule language, an efficient inference engine and tools that support definition and management of rules. Business rules should be able to bridge the gap between the business analyst's view and the information systems view of rules.

Business rules should be considered if your organization is facing one of the following challenges:

1. The company requires greater agility in its business processes. Ideally, you would like to quickly respond to competition and changing regulations and change the behavior of processes without modifying or redeploying the business process.

¹ Business Rules Group <http://www.businessrulesgroup.org>

2. Business processes need to be changed dynamically or optimized based on inferences from business activity monitoring tools. However, the business processes do not have appropriate interfaces to incorporate these inferences.
3. Business rules and decision points are embedded inside multiple processes in the organization. Hence it is not clear which processes depend on which organizational policy or guideline.
4. A central group or committee creates policies and guidelines - however during implementation, policies are often interpreted differently by each process owner and often applied inconsistently.
5. Business analysts have limited visibility and are unable to modify rules after they are included in the business processes. IT personnel generally implement rules differently from how they are expressed by business users
6. The company has no central repository of policies. Hence there is no easy way to change a policy and have it immediately applied across the organization in all the departments.

An increasing number of companies are looking at web services and the Service Oriented Architecture (SOA) as an architectural blueprint for addressing the integration requirements involved in building connected applications. BPEL and other web services standards address common application requirements in an open, portable and standard way. If your organization is facing one of the challenges listed above, you should also consider using business rules as part of your process architecture. The Oracle BPEL Process Manager and Oracle Business Rules enable organizations to build flexible SOA based solutions by maximizing leverage of existing resources while minimizing the cost of deploying new applications.

The rest of the paper discusses how rule engines can be effectively used as a Decision Service within the SOA framework. Section II and III provide an overview of the Oracle Business Rules and the Oracle BPEL Process Manager. Section IV describes the Decision Service and how it may be used with business processes. In Section V, we take a typical use case to describe usage of business rules in the context of the LoanFlow BPEL process.

II. ORACLE BUSINESS RULES

Oracle Business Rules is a new component of Oracle Application Server 10g. It allows business rules to be specified separately from application code. Separating the business rules from code allows business analysts to change business policy quickly with graphical tools.

Oracle Business Rules includes a high performance inference engine, a rules language and a rules authoring GUI with an SDK. Fig 1 shows the architecture of Oracle Business Rules.

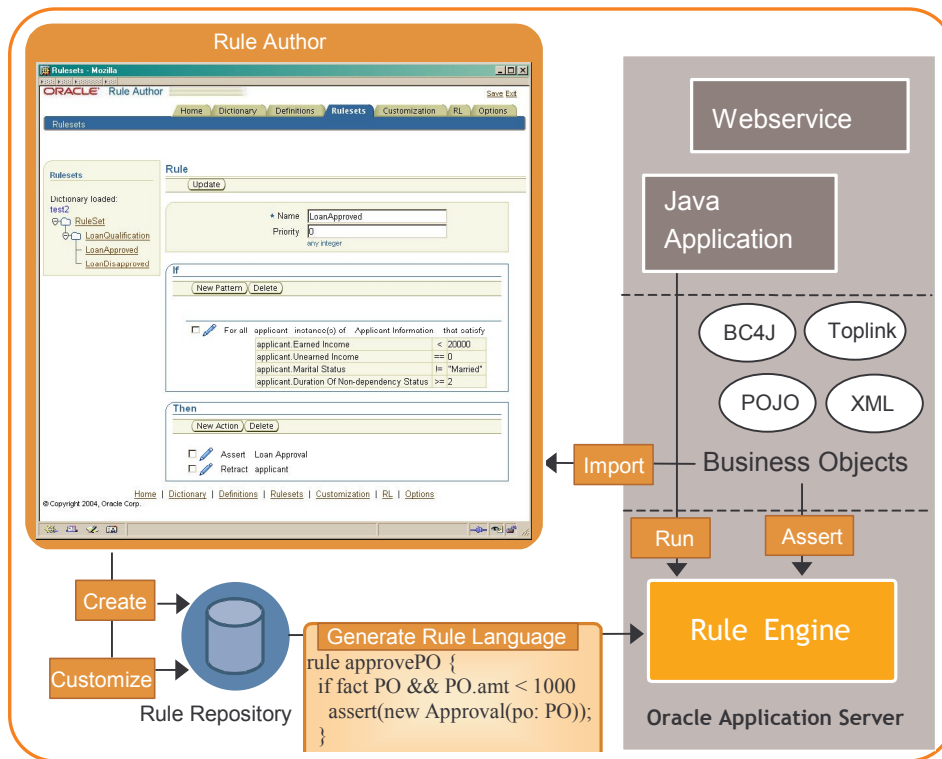


Fig 1: Oracle Business Rules Architecture

- The **Rule Author** allows business users to create and edit rules applications through an intuitive form interface. It uses a business dialect as opposed to a developer language. The Rule Author is a general-purpose application, which is layered upon the Rule SDK. It is supplied both as a thin client application and as a JDeveloper plug-in.
- The **Rules Language (RL)** is a fully featured rules programming language designed for integration with Java. The RL interpreter is written in Java and can be embedded in Java programs. From RL, one can create, modify, and write rules upon Java objects and can invoke Java methods. Like other rules languages, RL is largely declarative rather than procedural. This makes it much easier to develop certain kinds of applications, and allows the rules engine to globally optimize evaluation of multiple rules. RL programs are simply text strings which can be stored anywhere.
- The **Rule Engine** is an efficient implementation of the industry standard Rete algorithm. The main functionality of this engine includes the ability to load rules, assert/retract facts into working memory, perform inferences and interfaces to expose working memory status. The engine can be invoked as a java callable library or a standalone service – in both cases it consumes a modest amount of resources (working overhead is less than 2M bytes) and relies on a rules repository for rule sets and other metadata. The repository can be a database or files.

- The **Rule SDK** enables development of Java editing applications, typically GUI or web browser applications, through which business users can edit and view rules. The SDK provides (a) a dictionary API which allows the developer to define business objects and editing templates available to the business user, (b) an API for editing, validating, and debugging business rules and for generating correct runtime structures and (c) a runtime API for loading, executing, monitoring, and auditing business rules.

III. ORACLE BPEL PROCESS MANAGER

The most critical step in the business process lifecycle is the deployment of the process to a platform that can orchestrate the flow and execute the various tasks in the process. Orchestrating a set of services into an end-to-end process flow entails multiple technical requirements - which include binding to heterogeneous systems, synchronous, and asynchronous message exchange patterns, data manipulation, flow coordination, user workflow, exception management, non-deterministic events, compensating transactions, version management etc. The goal of the BPEL standard is to provide a richer and yet simpler abstraction/standard for addressing those requirements. BPEL is rapidly becoming the defacto industry standard for process orchestration and execution.

The **Oracle BPEL Process Manager** provides a comprehensive integration platform and is the most mature, scalable and robust implementation of the BPEL execution engine available today. The BPEL process manager leverages Oracle OC4J as the underlying J2EE server. Some of the key features of the process manager include:

- Standards support – The engine includes native support for BPEL based process orchestration and web services standards such as WSIF, XSLT etc. It uses WSDL as the component model and XML as the data model.
- Performance and scalability - The high performance BPEL engine executes multiple BPEL processes concurrently and provides a “dehydration” capability so that the state of long-running flows is automatically maintained in a database, enabling clustering for both fail-over and scalability. Other major features include side-by-side versioning, process partitioning and advanced exception management.
- Connectivity services - The extensible WSDL binding framework enables connectivity to protocols and message formats other than SOAP. Bindings are available for JMS, email, JCA, HTTP and many other protocols for connecting to hundreds of back-end systems. Adapters are available to connect to a variety of packaged applications (SAP, PeopleSoft etc.) and legacy systems.
- Data management services – these include data transformation using XSLT/XQuery and translation from non-XML formats to XML and vice versa.
- User workflow services – these services enable the integration of people and manual tasks into BPEL flows. The workflow features include task assignment and routing, multiple workflow patterns, identity services, notification services and a comprehensive worklist.

- The BPEL Console provides a web-based interface for management, administration and debugging of processes deployed to the BPEL server. Audit trails and process history/reporting information is automatically maintained and available both through the BPEL Console.
- Sensor based Business Activity Monitoring framework - User defined sensors can be used to track specific activities or variables within the process and take specific actions.

Figure 2 shows the architecture of the BPEL Process Manager and the various components.

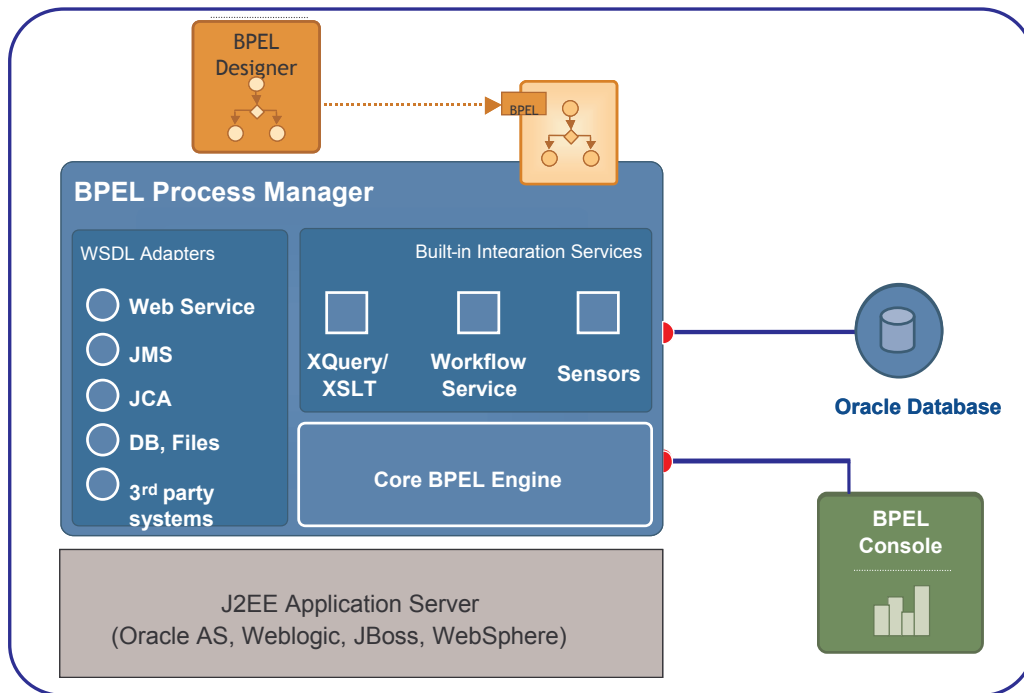


Fig 2: BPEL Process Manager architecture

IV. BUSINESS RULES AS A DECISION SERVICE

As described earlier, the SOA framework enables creation of composite applications by integrating a set of synchronous and asynchronous services and systems into a process flow. Enabling these applications is a two-step process. The first step is to publish various services that will be used in the application and the next step is to compose, or orchestrate them into business flows. Publishing a service means taking a function within an existing application or system and making it available in a standard way, while orchestration is composing multiple services into an end-to-end business process. Business rules can easily plug into this framework as a Decision Service.

A Decision Service is a mechanism for publishing rules and rule sets as a reusable service that can be invoked from multiple business processes. This service is a standalone deployment unit that comprises the following components:

1. A web service that wraps the rule session to the underlying rules engine. This service would let business processes assert and retract facts as part of the process. In some cases all facts may be asserted from the business process as

one unit, while in some cases the business process may incrementally assert facts and eventually consult the rules engine for inferences. Hence the service has to support both stateless as well as stateful interactions.

2. Rules that are evaluated by the Decision service using the rules engine. These are defined using the Rule Author and loaded in the rules repository.
3. Metadata that describes facts that are needed for specific rules to be evaluated. Each rule that is exposed as a service will use different types of facts. These facts have to be exposed via XSD definitions and appropriate WSDL operations need to be exposed for rule evaluation. For example: a CreditRating rule set may expect a customer's SSN, previous loan history etc as facts, but a PensionPayment rule may expect an employee's years of service, salary, age etc as facts.
4. Adapters that load facts periodically from backend systems such as databases or files. Facts may be passed in directly from the business process or may be periodically loaded from a backend repository such as a database, file system or an external application. The adapter services can be used internally as part of the Decision service to load facts. Example: For the CreditRating decision service, facts such as the customer's SSN and loan amount may be passed in from the business process, but other facts such as the customer credit history and outstanding loans may be retrieved using adapters from a backend repository

Figure 3 shows the Decision service as part of the overall BPM infrastructure.

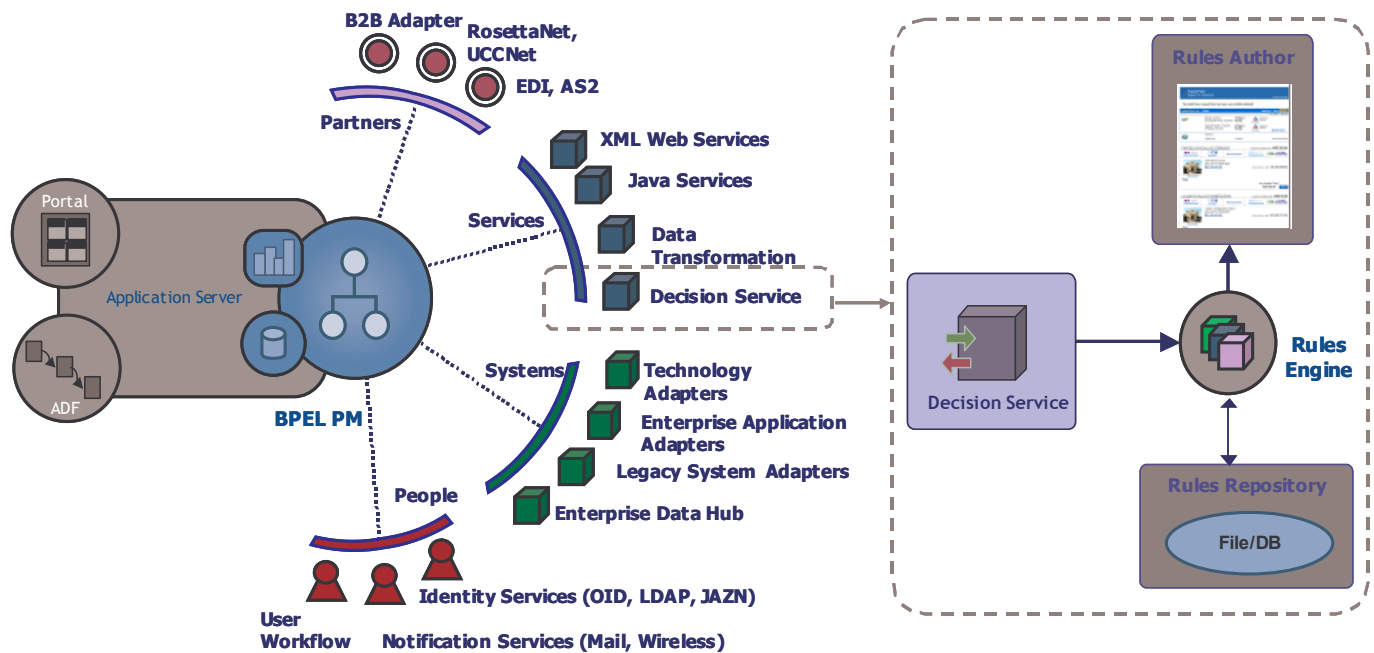


Fig 3: Decision Service with BPEL Process Manager

Some examples of decision service usage are:

- Dynamic processing – rules can be used to perform intelligent routing within the business process based on service level agreements or other guidelines. For example: “if the customer requires response within one day send the loan application to StarLoan only. If the customer can wait longer, then route the request to 3 different loan agencies”
- Externalize decision points in the process – Typically there are a number of conditions that need to be evaluated as part of a business process. However, the parameters to these may be changed independent of the process. For example: give loans only to customers with credit score of at least 650. This value may be changed dynamically based on new guidelines by business analysts
- Data validation and constraint checks – rules may be used to validate input documents or apply additional constraints on requests. For example: a new customer request must always be accompanied with an employment verification letter and bank account details.
- User workflow – rules are frequently used in the context of user tasks in the business process
 - Policy based task assignment is used to dispatch tasks to specific roles or users. For example: a process that is used to handle incoming requests from a portal may route Loan requests and insurance quotes to a different set of roles.
 - Escalation and Delegation policies - these policies may be used to route requests to multiple users for approval or to escalate tasks to other users if a specific user does not act in a certain amount of time. For example: if a user does not act on a high priority task in 2 days it may be reassigned to his (or her) manager or to an alternate user. Another example: a new customer insurance policies must always be approved by the agent and a supervisor.
 - Load balancing of tasks among users – when a task is assigned to a set of users or a role, each user in that role acquires a set of tasks and acts on them in a specified time. For new incoming tasks, policies may be applied to set priorities on the task and put them in specific user queues. For example: A specific loan agent is assigned at most 10 loans at any time assuming 80% resource utilization
- Business activity monitoring – rules may be used in conjunction with BAM tools to raise alerts based on certain policies. Also inferences from KPI measurements may be used to affect the flow of the business process. For example: If the number of loans this month from high risk customers has exceeded 30, then send stop accepting loan requests from customers with credit score less than 700.

V. LOAN FLOW SCENARIO

Consider the following example of a LoanFlow process that is deployed at a typical loan broker. The loan broker accepts a request from a client, performs a credit check with an external service and then routes the application to two different loan agencies. After receiving the offers, the best offer is selected and the customer is notified. The scenario consists of the following participants - LoanBroker, CreditRating service, StarLoanService, UnitedLoanService and the customer. The LoanFlow process (Fig 4) orchestrates interactions between these services. Lets use this scenario to illustrate usage of the Decision Service. Lets assume that the rules engine is used in both the CreditRating service and the StarLoan service.

The Credit rating service may use the customers SSN, prior credit history, age and outstanding loans to determine the customer credit rating, the risk and maximum amount to lend a specific customer. In this case some of the rules could be

```
IF customer credit score is less than 500 AND outstanding loans are greater than $50,000 AND  
loan requested is greater than $40,000 THEN  
set the Credit Risk to "High"  
IF customer credit score is greater than 500 AND outstanding loans are less than $50,000 AND  
set the Credit Risk to "Low"
```

However, different loan agencies may interpret the results differently and further apply rules to determine if the customer should be granted the loan, what interest rate should be given and the appropriate approval policies. For example, StarLoan loan may use the following rules to approve or reject the loan:

```
// Rules for "High" risk customers  
IF customer credit risk is High AND loan requested is greater than $40,000 AND  
[outstanding loans this month to high risk customers] is greater than $5,000,000 THEN  
"Reject" the Loan application  
  
IF customer credit risk is High AND loan requested is less than $50,000 THEN  
"Approve" the Loan application with an APR of 5%  
  
IF customer credit risk is High AND loan requested is between $50,000 and $75,000 THEN  
"Route" the Loan Application to Manager for review  
  
IF customer credit risk is High AND loan requested is greater than $75,000 THEN  
"Reject" the Loan application  
  
// Rules for "Low" risk customers  
IF customer credit risk is "Low" OR the customer credit risk is "Medium" THEN  
"Approve" the loan with an APR of 4.5%  
  
IF the customer credit risk is not "High" AND the customer is a returning customer THEN  
"Approve" the loan with an APR of 4%
```

In this case the Decision Service is used to evaluate rules based on both static data i.e the loan application as well as data such as "outstanding loans to high risk customers" from the Business Activity monitoring system. Similarly, rules are also used for dispatching the loan application to multiple levels for high-risk customers. Note that these conditions

could have been done as “switch” statements in the business process itself. However, using business rules gives the loan agency the flexibility to change these dynamically based on market conditions without redeploying the business process.

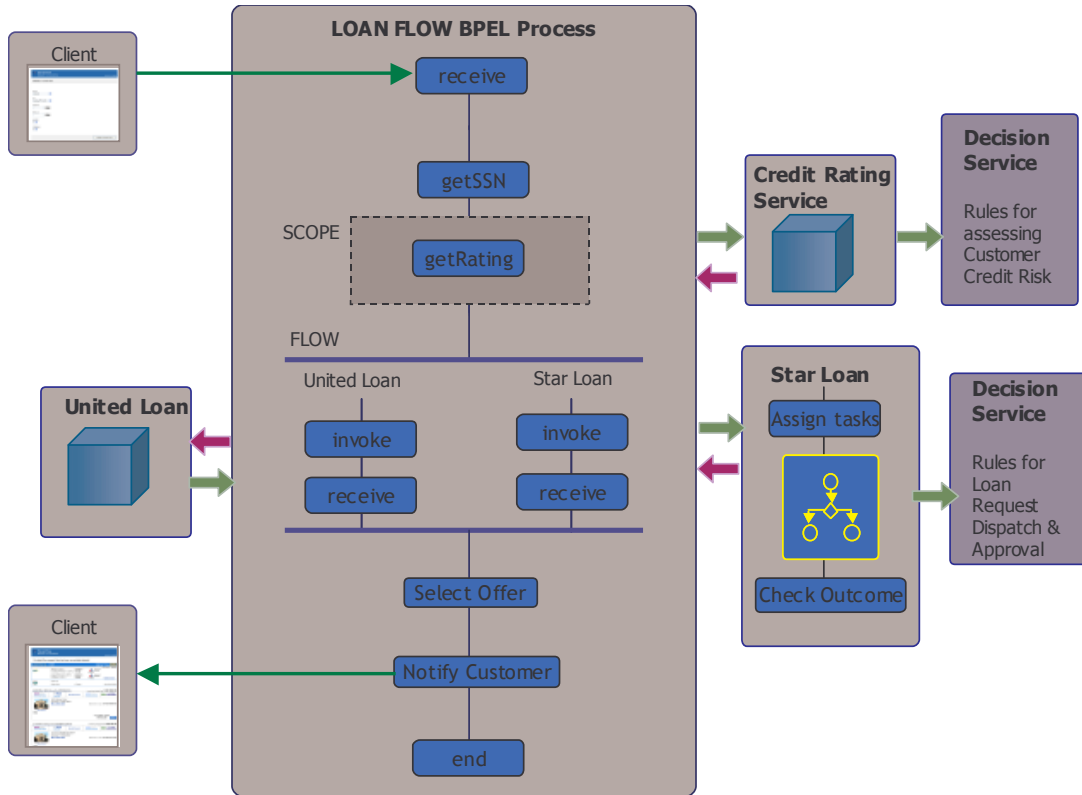


Fig 4: Decision Service as part of the LoanFlow BPEL process

VI. PARTNERS

Oracle also partners with a number of vendors to provide business rules capabilities in conjunction with the Oracle BPEL Process Manager and Business Activity Monitoring. Some of these vendors include

- ILOG, Inc. (<http://www.ilog.com>) – ILOG JRules can be used to create a web service that can be used for making decisions as part of BPEL Processes

VII. CONCLUSION

Business Rules Engines are critical for the deployment of effective business processes that can quickly respond to changing business conditions. Oracle Business Rules and Oracle BPEL Process Manager provide a comprehensive platform for building rule based business processes that enable maximum agility and transparency. For more details on these technologies go to <http://www.oracle.com/technology/index.html>

Building Flexible Enterprise Processes using Business Rules and BPEL Process Manager

Jan 2005

Authors: Bhagat Nainani, Kathryn Gruenefeldt, Ralf Mueller

**Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.**

**Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com**

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2001 Oracle Corporation
All rights reserved.