# Table of Contents

**BUSINESS RULES CONCEPTS** .................................................................................................................. 2

**BUSINESS RULES** ..................................................................................................................................... 2

**RULE INFEERENCE CONCEPT** .................................................................................................................. 2

**BASIC BUSINESS RULES CONCEPT** ........................................................................................................... 3

**BUSINESS RULE ENGINE ARCHITECTURE** ............................................................................................... 4

**BUSINESS RULE ENGINE ARCHITECTURE** ............................................................................................... 4

**DECLARATIVE RULE** ...................................................................................................................................... 5

** RETE ALGORITHM** ......................................................................................................................................... 5

By using the RETE Algorithm .......................................................................................................................... 5

Benefits of RETE Algorithm ............................................................................................................................ 5

**FIRING OF RULES IN RULE SESSION** ........................................................................................................ 6

**OVERVIEW OF THE BUSINESS RULE COMPONENT IN SOA COMPOSITE APPLICATION** ..................... 7

**INTRODUCTION TO THE BUSINESS RULE COMPONENT** ......................................................................... 7

**DEVELOPING BUSINESS RULE FOR AN APPLICATION** ............................................................................ 7

**DECISION SERVICE ARCHITECTURE** .......................................................................................................... 8

**WORKING WITH DECISION SERVICE** ......................................................................................................... 9

**EXAMINING BUSINESS RULES COMPONENT FILES** .............................................................................. 10

**RULES DICTIONARY** .................................................................................................................................... 10

**WORKING WITH DATA MODEL ELEMENTS** ............................................................................................ 11

**INTRODUCTION TO DATA MODEL ELEMENTS** .......................................................................................... 11

**CREATING XML FACTS, RL FUNCTIONS, GLOBALS, BUCKETSETS AND RULESETS** ............................... 11

Creating links to Other Dictionaries ................................................................................................................ 12

**IDENTIFYING THE RULE STRUCTURE** ....................................................................................................... 12

**WORKING WITH DECISION TABLES** ......................................................................................................... 12

**UNDERSTANDING BUSINESS RULES AND DECISION FUNCTIONS** ....................................................... 13

Invoking Rules in Decision Functions ........................................................................................................... 13

**TESTING BUSINESS RULES** ...................................................................................................................... 14
Business Rules Concepts

Business rules

Business rules are:

- Declarative statements that describe business policies or describe key business decisions, such as:
  - Business policies such as spending policies and approval matrices.
  - Constraints such as valid configurations or regulatory requirements.
  - Computations such as discounts or premiums.
  - Reasoning capabilities such as offers based on customer value.
- Separated from code to enable their definition and modification to be made by nontechnical users.

Rule Inference Concept

Online Dictionary defines Inference as "the act or process of inferring."

Inferring means to derive a conclusion from facts. To make an inference means that one set of true conditions can imply "something else," and then that "something else" can be used to further satisfy other conditions, which in turn imply other things.

Inferences in a Rules Engine means that these "chained" implications are done automatically as part of the normal operation of the engine.

Rule inference:

- Makes a statement of truth based on a former truth
- Enables powerful and modular declarative assertions

![Rule Inference Diagram](image-url)
Basic Business Rules Concept

Oracle Business Rules are stored in a rule dictionary, which is a file in the file system. A rule dictionary contains one or more definitions of:

- Facts
- Constraints
- Functions and
- Rulesets

Facts:
- Are data or business objects on which the Rules Engine evaluates rule conditions.

Rules:
- Are declared as: “If Condition Then Action”.
- Have an action: Assign, Assert, Call function (or Java method).
  OR
- Each rule is declared as a conditional expression with an action. The condition evaluates and compares facts. If the evaluation of facts is true, then one of the following actions is performed:
  o Assigning values to other facts or results
  o Asserting values
  o Calling functions (or Java methods) to execute procedural code

Ruleset:
- Has a collection of rules.
- Is a unit of execution.
- May be chained.
  OR
- A ruleset is a collection of one or more related rules that are seen as a unit of execution and can be chained together to yield an outcome.

Dictionary:
- Has a collection of fact types, global variables/constants, functions, and rulesets

Bucketsets:
- Bucketsets are constraints defined in the Rule Dictionary as lists of values, list of range values, or enumerations

Decision Function:
- Is a declarative function that handles the assertion of inputs as rule facts.
- Collects the consequent actions to be executed.
- Runs rulesets and returns results.

Globals:
-Globals are shared constants or variables defined in the Rule Dictionary that are accessible to all rules.
**RL Functions:**

- RL Functions are used in rules and defined in the Rule Dictionary using the Oracle Business Rule Language.

**Business Rule Engine Architecture**

Oracle Business Rules is a rule-based system based on the foundations of the RETE algorithm.

The RETE algorithm is used by Oracle Business Rules to optimize the pattern matching process for rules and facts added to working memory.

The RETE algorithm stores partially matched results in a single network of nodes in working memory, enabling the Rules Engine to avoid unnecessary rechecking when facts are added, modified, or deleted.

To process facts and rules, the RETE algorithm creates and uses an input node for each fact definition and an output node for each rule.

Oracle Business Rules is a data-driven forward chaining system, such that the facts determine which rules are evaluated. When a matching rule fires, the rule may add more facts. New facts are again compared against the rules. The process repeats until a conclusion is reached or the cycle is stopped or reset. This process is called an inference cycle.

A rule-based system using the RETE algorithm is the foundation of Oracle Business Rules. A Rule based system consists of:

- The rule-base: business policies expressed as IF-THEN rules and Decision Tables
- Working memory: fact information added to the system by applications using assert calls.
- Inference Engine (Rule Engine): processes the rules, performs pattern-matching to determine which rules match the facts, for a given run through a set of facts.
Declarative Rule

Oracle Business Rule use declarative rules, where you create rules that make declarations based on facts rather than coding.

In declarative rules:
- Statements are declared without any control flow
- Control flow is determined by the Rules Engine
- Rules are easier to maintain than procedural code
- Rules relate well to business user work methods.

RETE Algorithm

- Oracle Business Rules uses the RETE algorithm to optimize the pattern matching process for rules and facts.
- The RETE algorithm stores partially matched results in a single network of nodes in working memory.
- To process facts and rules, the RETE algorithm creates and uses an input node for each fact definition and an output node for each rule.

By using the RETE Algorithm

By using the RETE algorithm, the Rules Engine avoids unnecessary rechecking when facts are deleted, added, or modified. To process facts and rules, the Rete algorithm creates and uses an input node for each fact definition and an output node for each rule.

Fact references flow from input to output nodes.

In between input and output nodes are test nodes and join nodes.
- A test occurs when a rule condition has a Boolean expression.
- A join occurs when a rule condition ANDs two facts.

A rule is activated when its output node contains fact references.

Fact references are cached throughout the network to speed up recomputing activated rules. When a fact is added, removed, or changed, the RETE network updates the caches and the rule activations; this requires only an incremental amount of work.

Benefits of RETE Algorithm

The benefits of implementing the RETE Algorithm is: Rules can be added and removed without affecting other rules Rules with common conditions share nodes in the Rete network Each rule firing typically changes just a few facts and the cost of updating the Rete network is proportional to the number of changed facts, not to the total number of facts or rules.
The benefits of implementing the RETE Algorithm are:

- Rules can be added and removed without affecting other rules
- Rules with common conditions share nodes in the Rete network
- Each rule firing typically changes just a few facts and the cost of updating the Rete network is proportional to the number of changed facts, not to the total number of facts or rules.

Firing of Rules in Rule Session

Oracle Business Rules uses working memory to maintain fact data, asserted by client applications. A Rule Session is created for client applications and contains working memory for that session. Working memory holds rules, facts and an agenda comprised of one or more activations. When a client asserts (adds) or retracts (removes) fact instances in working memory, that is when facts are changed or rules are explicitly executed:

- Pattern matching of rules and facts is performed, such that:
  - Matching rules are added to the agenda, called Activations
  - Non-matching rules are removed from agenda
- Rules are fired for activations, causing the rules to run and execute associated actions, when conditions for rules evaluate with a true result

Execution of actions may assert, modify, or retract facts and cause new activations to be added or removed from the agenda. A loop condition is possible if a rule’s action causes it to fire again. Rules are fired sequentially, but not in any pre-defined order. A Rule Session contains a ruleset stack, such that activated rules are fired as follows:

- Rules within top-of-the-stack ruleset are fired
- Within a ruleset, firing is ordered by user-defined priority
- Within the same priority, the most recently activated rule is fired first

Only rules within the ruleset stack are fired. However, all rules in a session are pattern-matched, and if matched they are activated.
Overview of the Business Rule Component in SOA Composite Application

Introduction to the Business Rule Component

A Business Rules service component, also called a Decision component, can be used in an SOA composite application:

- Wired and executed by a BPEL process component
- Exposed as a service to directly to run business rules
- Executed to implement dynamic routing with a Mediator component
- Implemented as advanced routing rules in Human Task definitions

Developing Business Rule for an Application

When developing Oracle Business Rules you use the JDeveloper Rules Designer to create a set of rules that can be evaluated by an application at run-time. Creating a Rule Dictionary requires:

1. Creating or importing a dictionary into a project using the JDeveloper Rules Designer.
2. Populating the Rule Dictionary with the set facts, functions, globals, bucketsets, links, decision functions, and rulesets. A data model contains business data definitions for facts or data objects used in rules. The Rule Dictionary is an XML file that stores the data model for an application in a .rules file.

Note: Globals are variable or constant data that are visible to all rulesets in the dictionary.

Bucketsets can be used as a list of values, to represent a value range, or as an enumeration.
3. Creating one or more facts, which represent data or objects added and maintained in working memory of a Rule Session. Facts can represent a sales order or a customer credit history. Facts can be created as XML Facts (derived from XML schema elements), Java Facts (based on a Java class), Rule Language (RL functions) facts, ADF Business Component Facts (based on View Object definitions).

Note: At run-time, mapping between the XML Schema definition and the XML Fact types uses the Java Architecture for XML Binding (JAXB).

4. Creating rulesets as a container for a set of one or more rules or decision tables.

5. Defining rules to express the business policies based on the facts in the data model.

**Decision Service Architecture**

A Decision Component is:

- A mechanism for publishing rules and rulesets as a reusable service
- Is an SCA component within a composite and wired to a BPEL component for rule execution, used in a Mediator component for dynamic routing, and used for Advanced Routing Rules in Human Task components.
**Working with Decision Service**

**A Decision Service:**

- Is a Web Service interface for executing exposed decision functions for underlying Oracle Business Rules.
- Is invoked from multiple business processes, such as a Java and composite applications.

A **Decision Service** is a mechanism for publishing executable decision functions, such as rules in a ruleset, as a reusable service that can be invoked from multiple business processes. Decision Service metadata (in a .decs file added to the composite application) comprises the decision function name exposed through a Web Service interface.

The goal is to separate volatile policy logic from other business logic. The policies are implemented in rules executed on a Rules Engine, and other business logic is implemented in Java executing in Java Virtual Machines (JVMs).

A **Decision Service** is configured declaratively and performs the following operations:

- Collects rulesets and other decision functions under a single executable umbrella.
- Handles the assertion of inputs as rule facts into the Oracle Business Rules Engine working memory.
- Collects the consequent actions to be executed.
- Runs rulesets.
- Returns results.

In a **Decision Service** the rules executed are organized into several rulesets, which can be executed in a prescribed order. For example, the business application can send facts that flow to the first ruleset, which in turn may assert additional facts that are used by a second and subsequent rulesets until finally result facts flow back to the caller of the Decision Service as output facts.
Examining Business Rules Component Files

After creating the Business Rule component in your composite application, or BPEL process, several files are created in the project folder. The files are:

- A Decision Service XML schema that defines the structures for the input and output facts
  - e.g., OracleRules1_DecisionService_1Types.xsd
- A Decision Service WSDL document that defines the basic stateless and stateful Web service operations that can be invoked.
  - e.g., OracleRules1_DecisionService_1.wsdl
- A Business Rules Dictionary (.rules file) for containing the ruleset definitions, rule, facts, and globals added to the dictionary.
  - e.g., OracleRules1.rules
- The Business Rules metadata (.decs file) defining the mapping of Decision Service operations to the Decision Functions that can be executed.
  - OracleRules1.decs
- The Business Rules .componentType file representing the component definition for the Composite Editor.
  - e.g., OracleRules1.componentType

Rules Dictionary

Creating or editing the Rule Dictionary (.rules file) is done in the Rules Designer, in which you can define:

- Rulesets with one or more Rules, Decision Tables, or both
- Create Facts, Functions, Globals, bucketsets, Links, Decision Functions

A Dictionary holds rules that are executed by an application. The Rules Designer saves rules and their associated definitions in the Rule Dictionary (a .rules file) within a project sub-folder. The Rule Dictionary is created when you first add the Business Rule component to your composite or BPEL Process flow. By using the Rules Designer you can define:

- Facts, such as Java facts, XML facts, RL facts, and ADF Business Components facts.
- Rulesets, which are containers for one or more rules executed in sequence.
- Decision tables, which contain a set of related rules in a table format.
- Rules, which are IF-THEN constructs that evaluate conditions and execute actions.
- A constraint, which can be an enumeration, range, or regular expression.
- Globals (variables and constants), which are named types that can be based on object, string, or standard primitives, which can be used in rule definitions and actions.
- Functions, which are parameterized functions based on the Rule Language (RL) syntax.
- Bucketsets, which provide the ability to define a list of values or range of values to be used within your business rules.
- Links, which enable you to add, edit, delete, or view dictionary links.
- Decision functions, which enable you to create or edit decision functions.
Working with Data Model Elements

Introduction to Data Model Elements

- The Data Model in Oracle Business Rule consists of fact types, functions, globals, bucketsets, decision functions, and dictionary links.
- To implement the data model portion of an Oracle Business Rules application you create a dictionary and add data model elements.
- To complete the dictionary, you create one or more rulesets containing rules that use or depend upon these data model elements.

Creating XML Facts, RL Functions, Globals, Bucketsets and Rulesets

**Facts** define the structure of data objects asserted and saved in working memory.

Facts can be created from XML schema elements, Java classes, RL functions, and ADF-BC View Objects

**RL Functions**: Oracle Business Rules functions are similar to a Java method. However, unlike java methods Business Rules functions do not belong to a class. Business Rules functions can be used:

- To extend a Java application object model to perform rule operations without modifying the original Java code.
- In a condition or an action associated with a rule or a Decision Table.
- To share the same expression between several rules.
- To return results to the application.

**Globals**: Globals are similar to public static variables in the Java language, that is they are single values that are shared across rules in the Rule Dictionary. A global represents a data value or expression that can be evaluated in rule conditions and expressions. For example, if a 10% discount is used in several rules, you can create and use a "discount" global variable, so that changes to the value need only be done in one place while discount is applied to many rules in the dictionary.

**Bucketsets**: A Bucketset defines a set or range of values that can be used as a constraint to limit the acceptable set of values for a fact or a property of a fact in Oracle Business Rules. You can define a bucketset as a global construct that allows reuse, or as a local construct that only applies to one condition expression. You can associate fact type properties with bucketsets. There are three forms of bucketsets:

- **List of values (LOV)**: that defines a list of values.
- **List of Ranges**: that defines a list of value ranges, defined by the range endpoints.
- **Enumeration**: that defines a list of fixed types. Enumerated types can be imported from XML or Java facts.

**Rulesets**: Rulesets provide a way of organizing your rules. Before rules can be created, you need to create a ruleset as a container for the rules.
**Creating links to Other Dictionaries**

An Oracle Business Rules dictionary enables you links to other dictionaries in addition to the built-in dictionary, which includes the standard functions and types that all Oracle Business Rules applications require.

In addition to creating a main dictionary, you may create additional application-specific dictionaries to read and write their properties. The complete data model defined by a main dictionary and its linked dictionaries is called a **combined dictionary**. This is useful for data model sharing. A fully qualified dictionary name is called a **DictionaryFQN**, which consists of the following two components:

- The dictionary package name
- The dictionary name

A dictionary refers to a linked dictionary using its **DictionaryFQN** and an alias. Oracle Business Rules uses the **DictionaryFQN** to find a linked dictionary. The example shows how to create a linked dictionary by selecting the **Create > Resource Picker** option, and in the SOA Resource Browser either:

- Select a linked dictionary from a deployed composite through an MDS connection, or
- Select a linked dictionary from another project in the same application workspace.

**Identifying the Rule structure**

A rule has a specific structure to be meaningfully evaluated. The parts are listed and shown in the slide.

The following is a brief description of each part:

- The name identifies a specific rule in the ruleset.
- The rule test defines a condition that is evaluated for the matching pattern to limit rule evaluation for particular instances of the pattern matched type specified.
- The rule action is the functionality executed for the matching pattern and test of the rule.

**Working with Decision Tables**

A decision table:

- Executes a matrix of IF-THEN rules displayed in a single spreadsheet-style view.
- Contains a collection of related business rules with condition rows, rules, and actions.

A Decision Table enables you to create and use business rules in an easy to understand format that provides an alternative to the IF-THEN rule format. The Decision Table format is intuitive for business analysts who are familiar with spreadsheets. The formal structure of a Decision Table makes it easier to author, understand, and change multiple similar rules.

**Oracle Business Rules Decision Tables** provide the following features:
• Powerful Visualization: Compact and structured presentation. This visualization matches the way real world business policies that are expressed with many tables, declarative, and organized into simple steps.

• Error Prevention: Avoids incompleteness and inconsistency. Because a Decision Table is well structured, automated tools can check for conflicts, redundancy, and incompleteness to speed development of valid, consistent business rules.

• Modular Knowledge Organization: Group rules into a single table. A spreadsheet metaphor puts groups of rules that work together onto a single viewable pane. For example, if there are six rules that check an applicant’s eligibility, it is more convenient to see all the rules than to view the rules as individual but related rules.

• Optimization of Rules and Performance Benefits: Oracle Business Rules Decision Tables provide automated features that can reduce the number of required rules, as compared to the IFTHEN rules (this is called rule coalescing).

• Rule Validation and Verification: Provides capabilities for ensuring the logical consistency of rules before deployment. Automated tools for checking conflicts, incompleteness, or gaps, help speed development of valid, consistent business rules.

Understanding Business Rules and Decision Functions

A Business Rules Dictionary defines Decision Functions that:

• Specify the Service Name invoked to execute its rules.
• Enables other applications to execute rules associated with the Decision Function.
• Configures the collection of rulesets and decision functions as a single execution unit.
• Stores input data assertions in working memory.
• Collects rule actions to be executed.
• Executes the matching rulesets.
• Returns results to the calling application.

Invoking Rules in Decision Functions

The Business Rule Invoke activity executes rules specified by:

• The Dictionary, which defines the .rules file used, and
• The Service Name, a Decision Function’s Service Name

In a BPEL process the Business Rule activity is in fact a Scope that contains other standard BPEL activities, one of which is the Invoke activity that executes the rules defined in the rulesets selected in the Decision Function dictionary definition.

The two components that define which Decision Function is executed are:

• The Rule Dictionary in which the Decision Function is defined, where it is associated with a Decision Service by specifying the Service Name field in the Decision Function definition.
• The Service Name specified in the Decision Function is selected as the Service for the Business Rule added to the BPEL process. This links the Invoke operation to the required Decision Function.
Testing Business Rules

You can test the Business Rule either during:

- Design Time
- Runtime

Diagram:

- Testing Business Rules
  - Design time using Oracle Rules Designer
  - Runtime using Oracle Enterprise Manager