Progress Corticon BRMS Data Integration

Christopher S. Hogan
Senior Systems Engineer
October 5, 2013

PROGRESS EXCHANGE 2013
DISCOVER. DEVELOP. DELIVER.

Agenda

- Business Rules and Decisions
- Data Integration
  - Extended Operators
  - Service Call-Outs
  - Enterprise Data Connector
  - High Performance Batch Processor
- Wrap Up

Business Rules and Decisions

Corticon enables organizations to make better, faster decisions by automating business rules

DECISIONS

SHOULD CREDIT BE EXTENDED?

Do not provide credit to delinquent accounts

Senior officer approval required
for amounts greater than $100,00

DECISIONS IN SECONDS, PREVENT LOSSES, INCREASE CUSTOMER SATISFACTION

RULES

HOW TO PLAN A SHIPMENT?

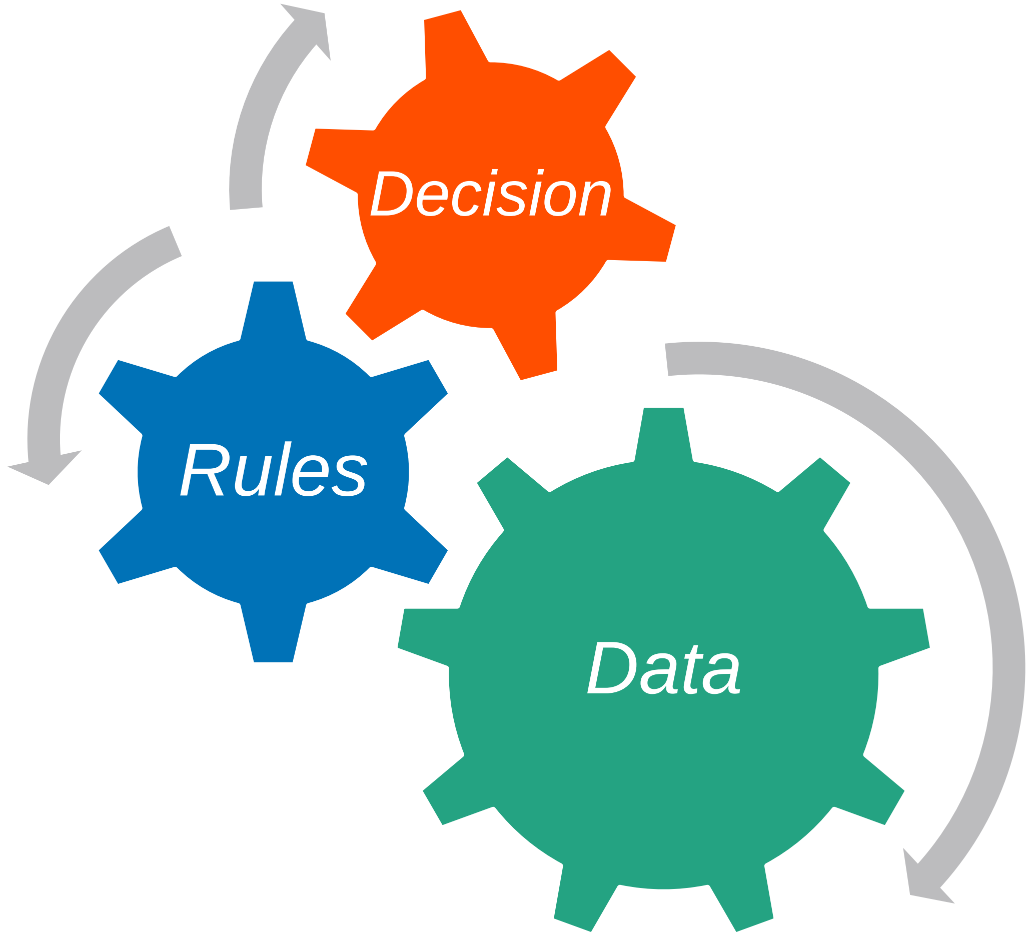
Hazardous materials must be shipped in double hull tankers

Class 7 super tankers require a minimum berthing distance of 300 feet

AVOID DISASTROUS OIL SPILL, REDUCE COSTS

Decisions, Rules & Data

- Decisions encapsulate a high level business function
- Business policies support the function as rules
- Rules evaluate and update business data
- Data is crucial to rules & decisions, but…
- Rules and decisions are generally agnostic to the source of the data



Amazon pricing scenario

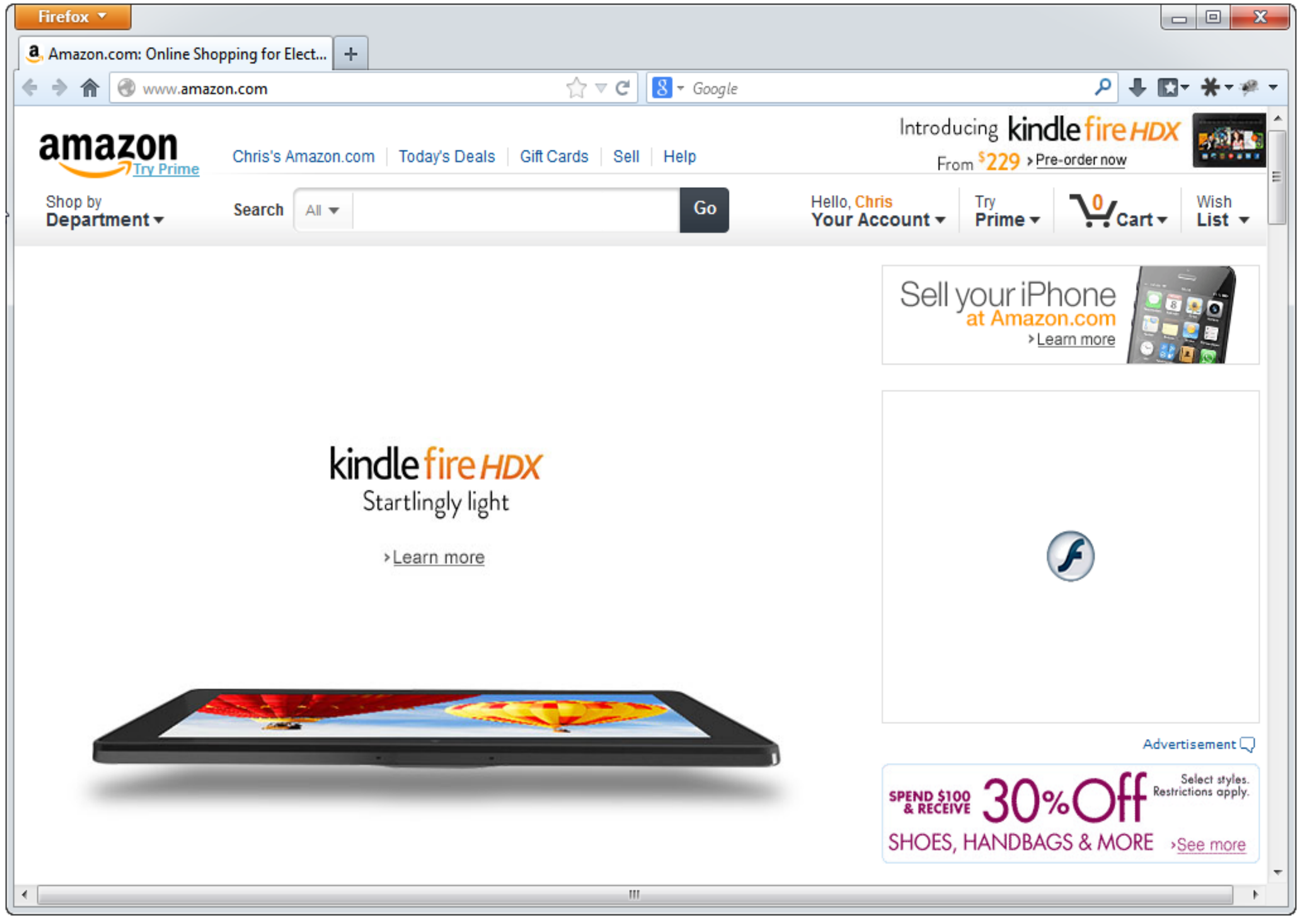
10% discount for:

- Customer longer than 2 years
- Sum of all orders is over $1000
- Applied for Amazon Prime
- Haven’t ordered in the last 6 months
- One order from a premium partner
- Live within 50 miles of a super hub

Very data intensive

Options

- Customer provides data
- Application provides data
- Decision gathers data


Firefox
Amazon.com: Online Shopping for Elect...
www.amazon.com
Google
amazon
Try Prime
Chris's Amazon.com
Today's Deals
Gift Cards
Sell
Help
Introducing kindle fire HDX
From $229 › Pre-order now
Shop by Department
Search
All
Go
Hello, Chris Your Account
Try Prime
0 Cart
Wish List
Sell your iPhone at Amazon.com › Learn more
kindle fire HDX
Startlingly light
› Learn more
Advertisement
SPEND $100 & RECEIVE 30% Off
Select styles. Restrictions apply.
SHOES, HANDBAGS & MORE › See more

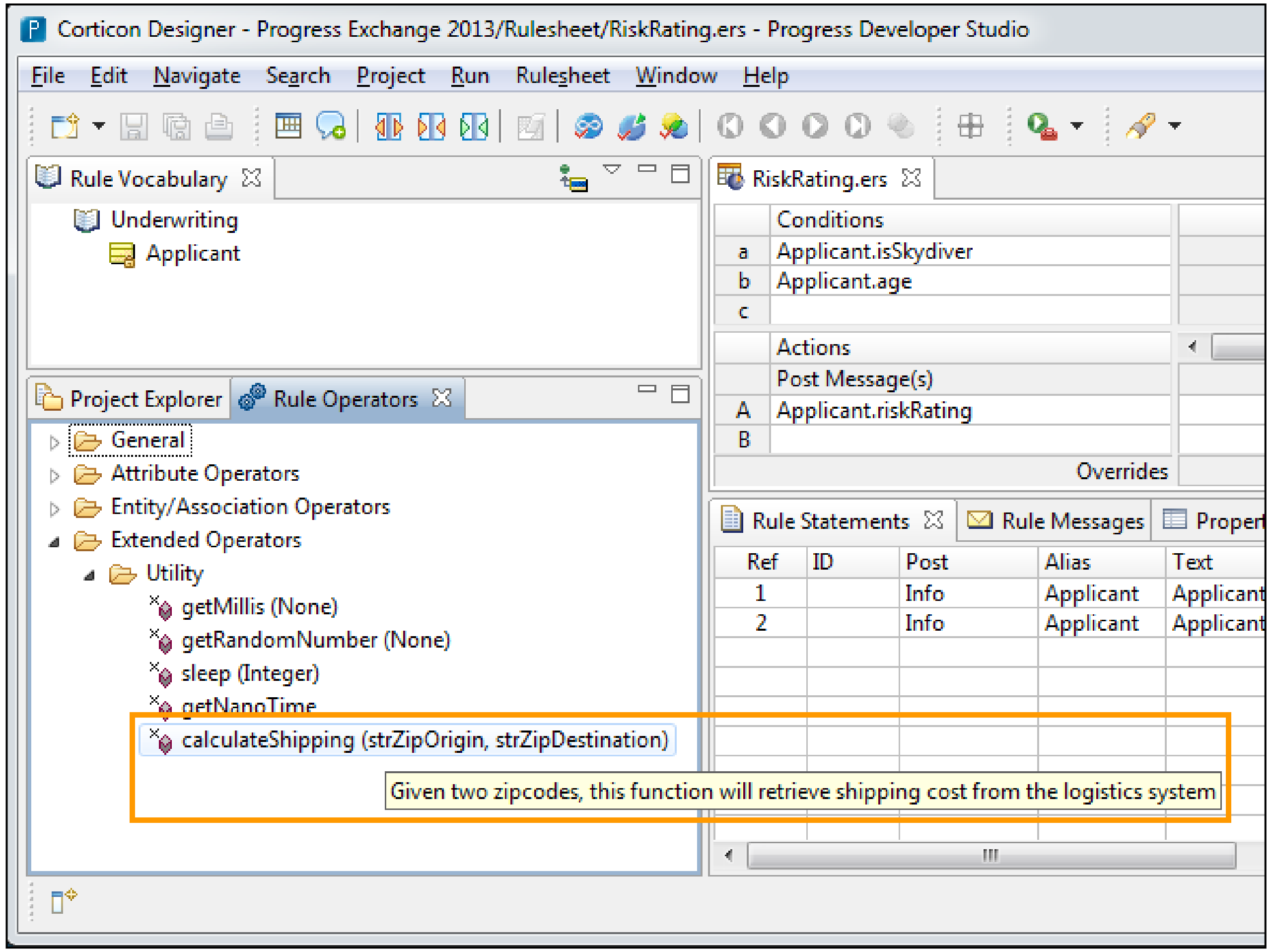
Agenda

- Business Rules and Decisions
- Data Integration
  - Extended Operators
  - Service Call-Outs
  - Enterprise Data Connector
  - High Performance Batch Processor
- Wrap Up

Extended Operators

Extended Operators

- Add your own function to the existing Corticon Rule Operators
- Can extend base data type
  - Boolean
  - Date
  - Time
  - Decimal
  - Integer
  - String
- Or, implement as a standalone
- Built as an Eclipse Plugin


Corticon Designer - Progress Exchange 2013/Rulesheet/RiskRating.ers - Progress Developer Studio
File Edit Navigate Search Project Run Rulesheet Window Help
Rule Vocabulary
Underwriting
Applicant
Project Explorer
Rule Operators
General
Attribute Operators
Entity/Association Operators
Extended Operators
Utility
getMillis (None)
getRandomNumber (None)
sleep (Integer)
getNanoTime
calculateShipping (strZipOrigin, strZipDestination)
RiskRating.ers
Rule Statements
Rule Messages
Proper

Given two zipcodes, this function will retrieve shipping cost from the logistics system

	Conditions
a	Applicant.isSkydiver
b	Applicant.age
c	
	Actions
	Post Message(s)
A	Applicant.riskRating
B	
	Overrides

Ref	ID	Post	Alias	Text
1		Info	Applicant	Applicant
2		Info	Applicant	Applicant

Demonstration

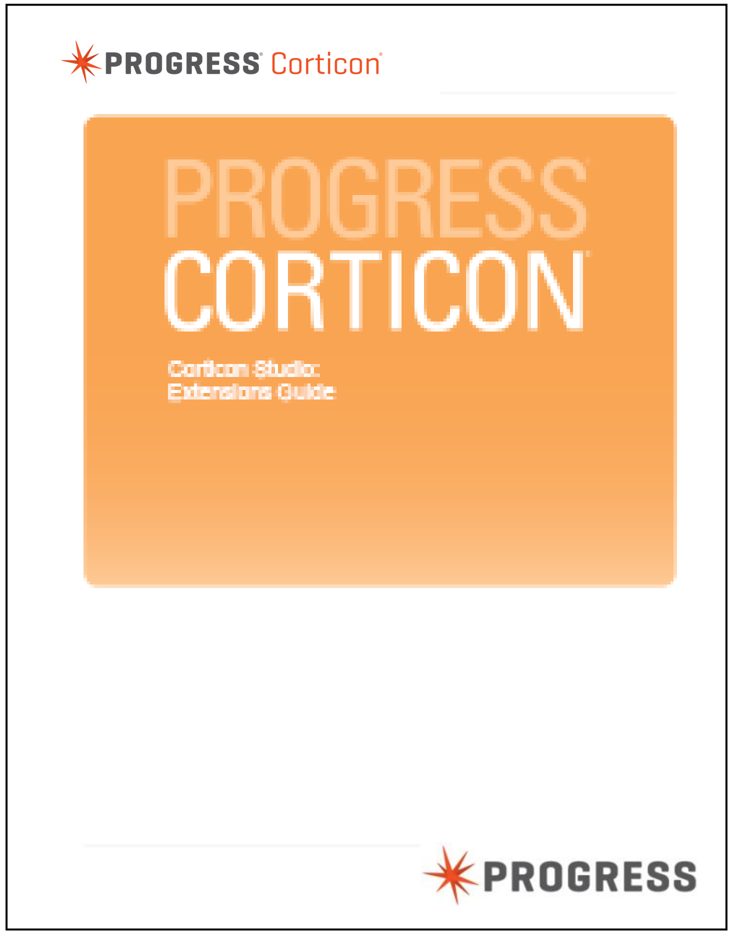
Extended Operators

Benefits:

- Simple and intuitive for rule authors
- Great for “simple” lookups
- Re-useable

Limitations:

- Can’t take an object (collection) as a parameter
- Can’t return data collections, only single values
- Must be coded in java



PROGRESS Corticon

PROGRESS CORTICON

Corticon Studio: Extensions Guide

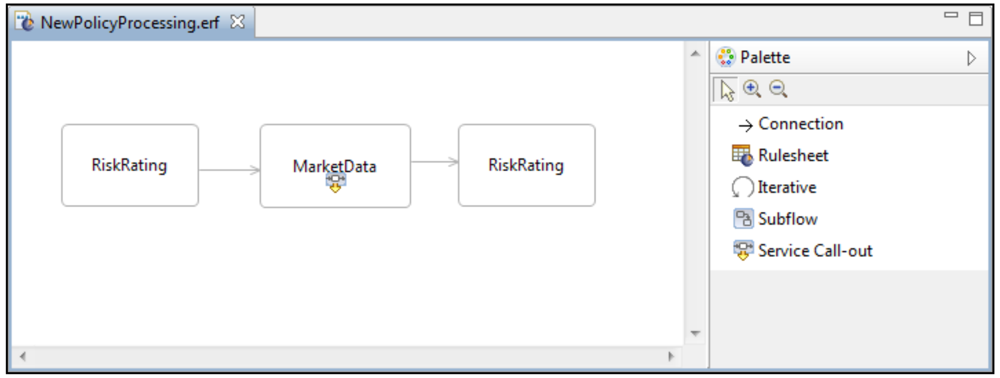
PROGRESS

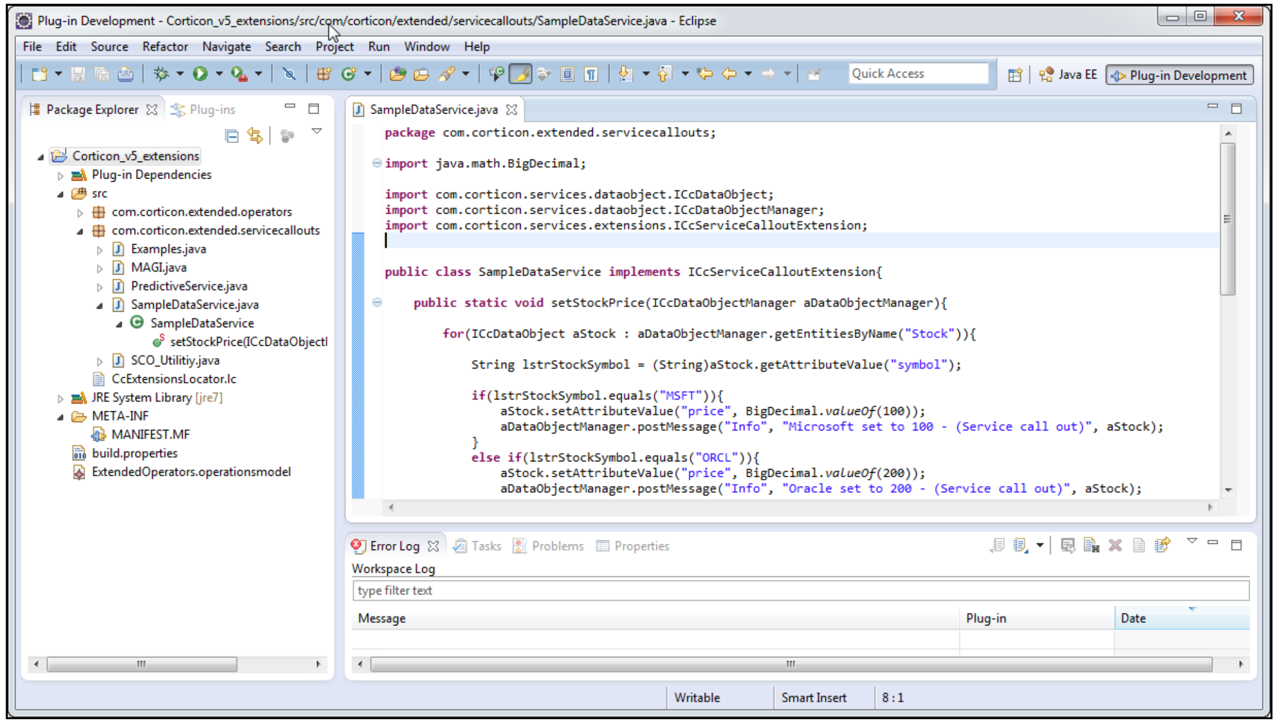
Service Call-Out

PROGRESS

Service Call-Outs

- Add your own complex integration code to the Ruleflow
- Modeled as a discreet step, just like a Rulesheet
- Built as an Eclipse plugin

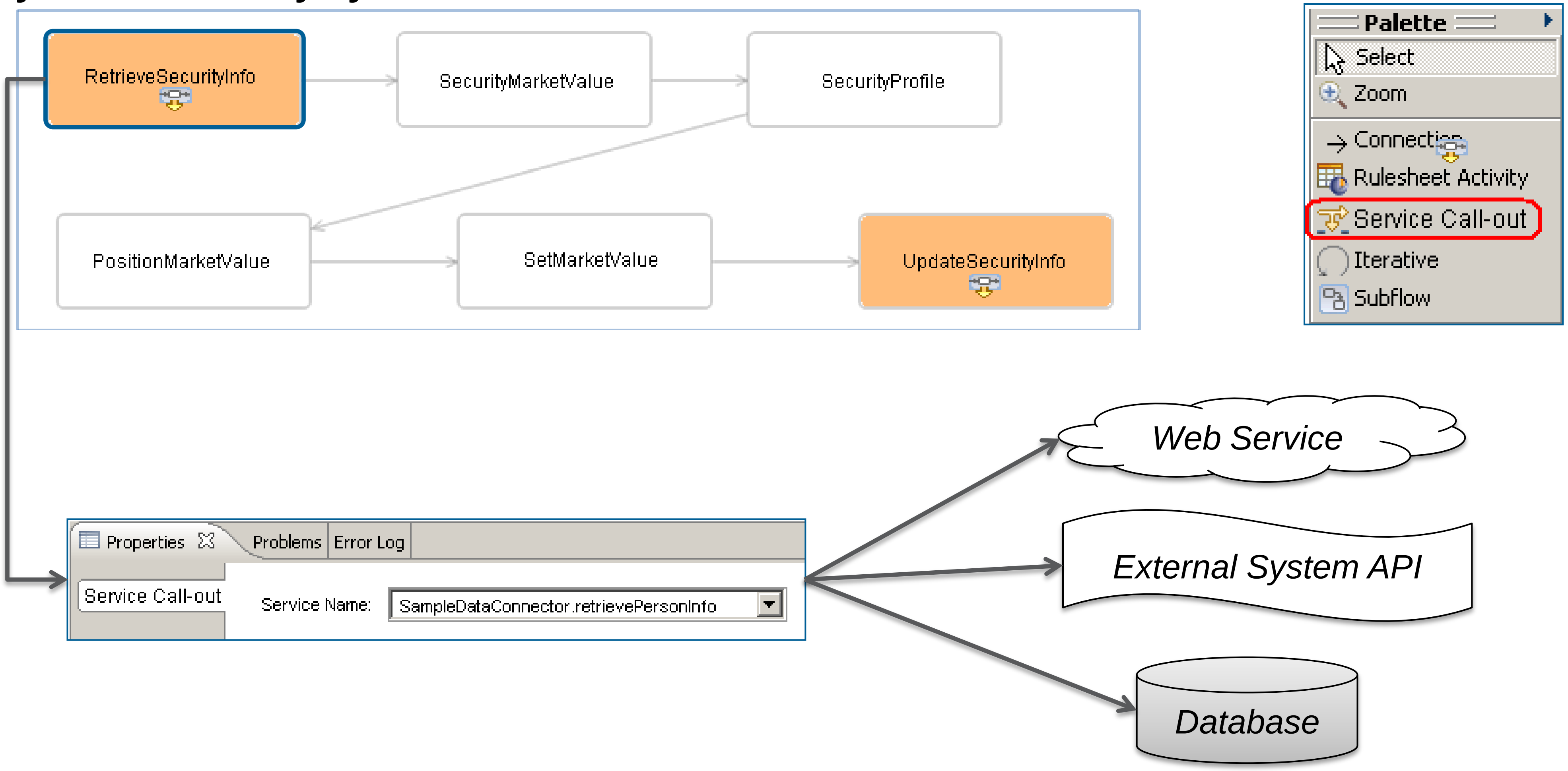




```java
package com.corticon.extended.servicecallouts;

import java.math.BigDecimal;

import com.corticon.services.dataobject.ICcDataObject;
import com.corticon.services.dataobject.ICcDataObjectManager;
import com.corticon.services.extensions.ICcServiceCalloutExtension;


public class SampleDataService implements ICcServiceCalloutExtension{

    public static void setStockPrice(ICcDataObjectManager aDataObjectManager){

        for(ICcDataObject aStock : aDataObjectManager.getEntitiesByName("Stock")){

            String lstrStockSymbol = (String)aStock.getAttributeValue("symbol");

            if(lstrStockSymbol.equals("MSFT")){
                aStock.setAttributeValue("price", BigDecimal.valueOf(100));
                aDataObjectManager.postMessage("Info", "Microsoft set to 100 - (Service call out)", aStock);
            }
            else if(lstrStockSymbol.equals("ORCL")){
                aStock.setAttributeValue("price", BigDecimal.valueOf(200));
                aDataObjectManager.postMessage("Info", "Oracle set to 200 - (Service call out)", aStock);
```

Service Call-Outs – Leverage Existing IT Services

Flexibly connect to any system or data source



Demonstration

PROGRESS

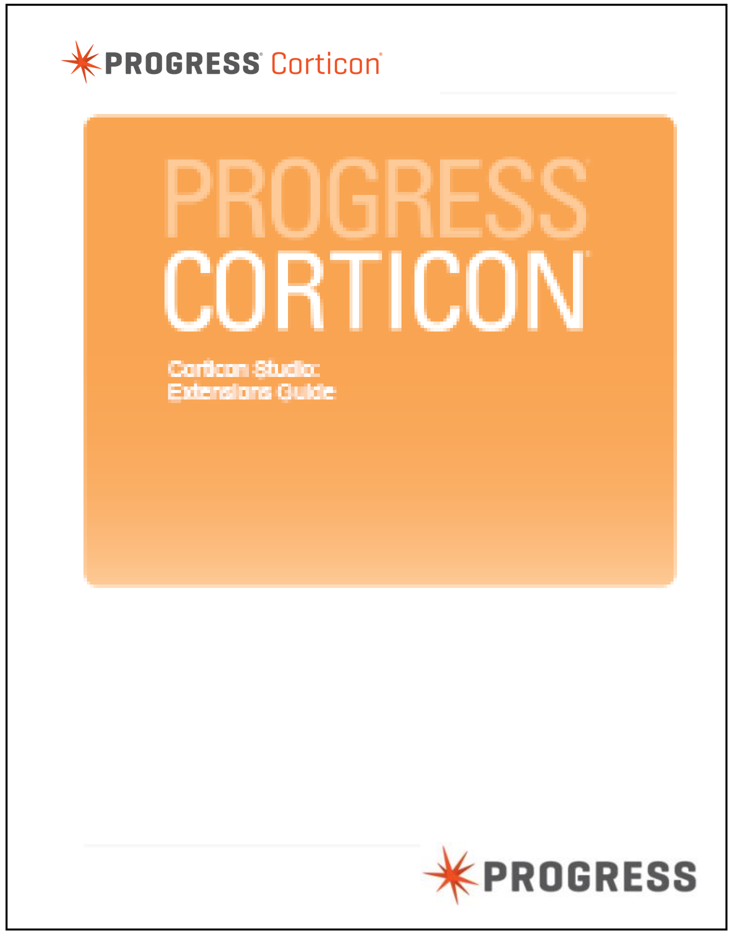
Service Call-Outs

Benefits:

- Provide run-time access to all working memory
- Have complete control of execution between Rulesheets
- Rule author is hidden from complexity

Disadvantages:

- Functions as a “black-box”
- Tightly coupled to a single Vocabulary
- Must be coded in java


PROGRESS Corticon

PROGRESS CORTICON

Corticon Studio: Extensions Guide

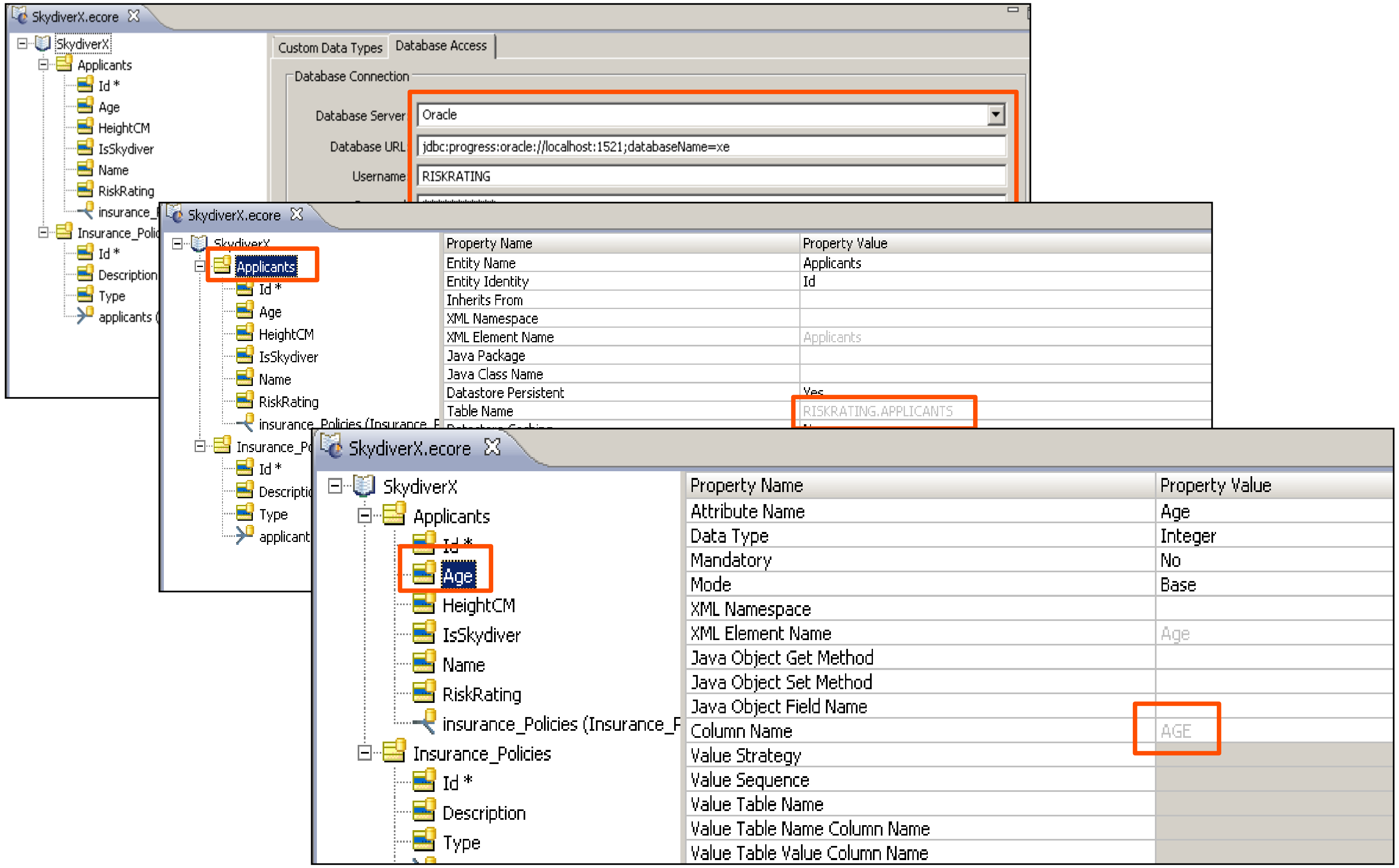
PROGRESS

Enterprise Data Connector

Enterprise Data Connector

- Provides read/write access to commercial RDBMS
- No coding at all!
- Vocabulary is mapped to database metadata via point and click

Enterprise Data Connector – Vocabulary Mapping



SkydiverX.ecore

- SkydiverX
  - Applicants
    - Id *
    - Age
    - HeightCM
    - IsSkydiver
    - Name
    - RiskRating
    - insurance_
  - Insurance_Polic
    - Id *
    - Description
    - Type
    - applicants (

Custom Data Types | Database Access

Database Connection

Database Server	Oracle
Database URL	jdbc:progress:oracle://localhost:1521;databaseName=xe
Username	RISKRATING

SkydiverX.ecore

- SkydiverX
  - Applicants
    - Id *
    - Age
    - HeightCM
    - IsSkydiver
    - Name
    - RiskRating
    - insurance_Policies (Insurance_F
  - Insurance_Po
    - Id *
    - Descriptio
    - Type
    - applicant

Property Name	Property Value
Entity Name	Applicants
Entity Identity	Id
Inherits From	
XML Namespace	
XML Element Name	Applicants
Java Package	
Java Class Name	
Datastore Persistent	Yes
Table Name	RISKRATING.APPLICANTS

SkydiverX.ecore

- SkydiverX
  - Applicants
    - Id *
    - Age
    - HeightCM
    - IsSkydiver
    - Name
    - RiskRating
    - insurance_Policies (Insurance_F
  - Insurance_Policies
    - Id *
    - Description
    - Type

Property Name	Property Value
Attribute Name	Age
Data Type	Integer
Mandatory	No
Mode	Base
XML Namespace	
XML Element Name	Age
Java Object Get Method	
Java Object Set Method	
Java Object Field Name	
Column Name	AGE
Value Strategy	
Value Sequence	
Value Table Name	
Value Table Name Column Name	
Value Table Value Column Name	

Enterprise Data Connector

1. In Request Payload, using Seed data = supplying the Primary Key of a table.
   - Applied when:
     - Consuming system supplies PKs in Request Payload (*minimal* size of the Request Payload)
     - All associated data is automatically retrieved (lazy loading)
     - Ideal when you have a complex vocabulary whereby passing in an entire payload requires complex data manipulation on the consumer side to get the payload populated correctly.

2. In Rulesheet using **Extend to DB** Scope section and *optionally* using **Database Filters**
   - Applied when:
     - During rule processing additional data must be conditionally looked up (e.g. rates)
     - Batch processing

Demonstration

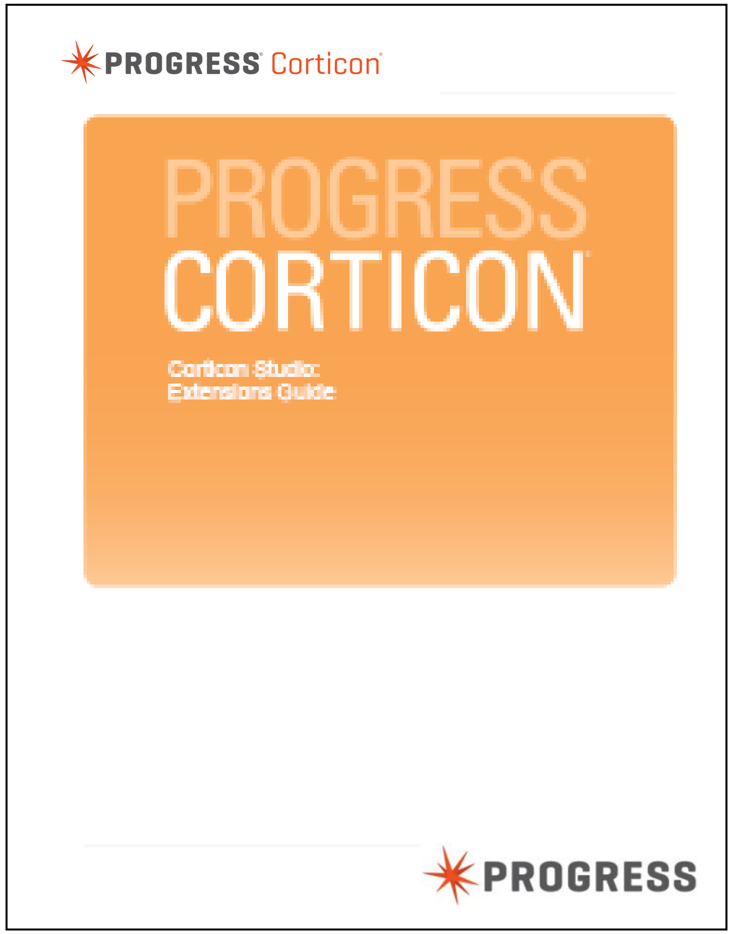
Enterprise Data Connector

Benefits:

- Model driven approach – No SQL/JDBC code
- Database independence – Rules don't change if data does
- Minimize client application re-factoring when data needs change
- Leverages industry leading Progress DataDirect technology

Disadvantages:

- Limited to RDBMS
- Require good DB schema (PKs, FKs, etc)
- Require JDBC connectivity to database


PROGRESS Corticon

PROGRESS CORTICON

Corticon Studio: Extensions Guide

PROGRESS

High Performance Batch Processor

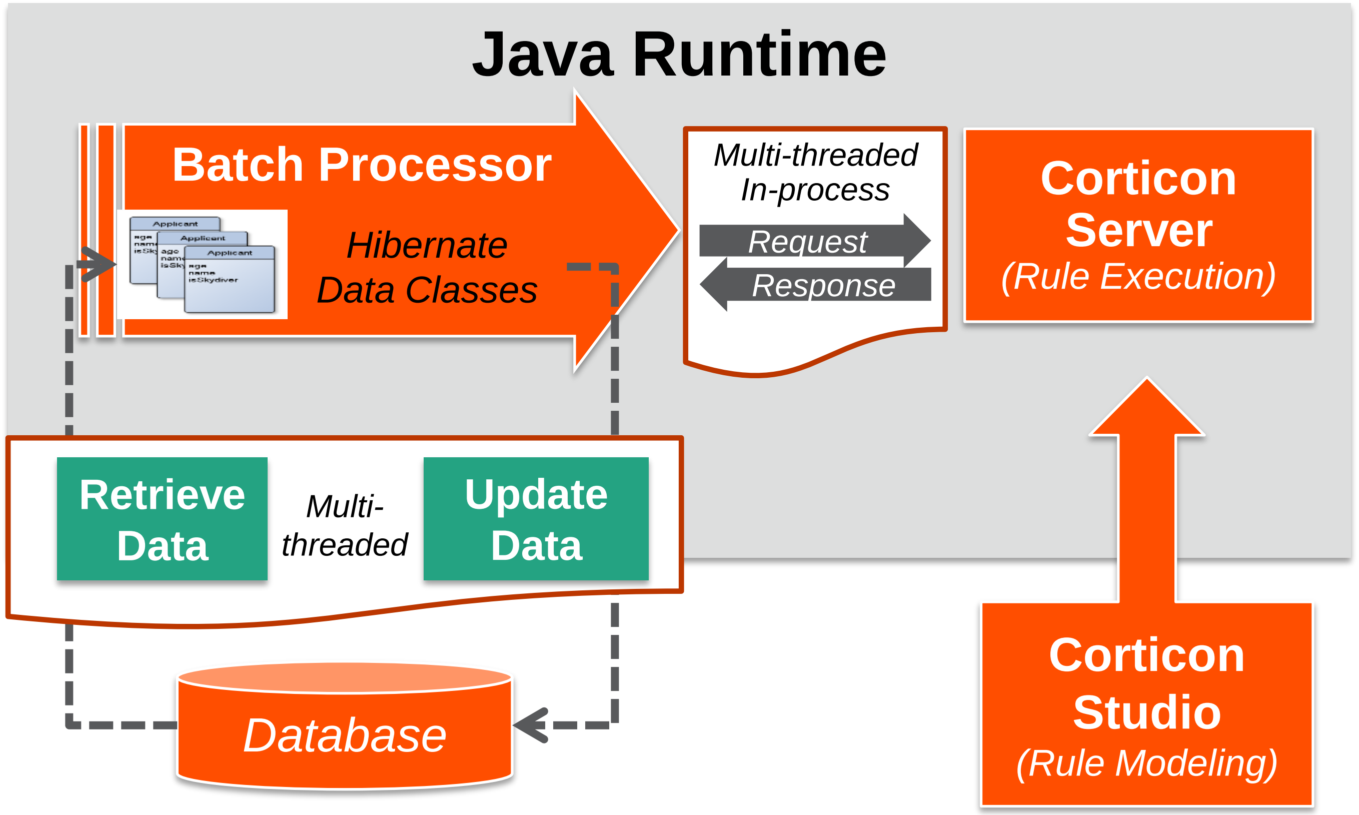
PROGRESS

High Performance Batch Processor

- Executing large volumes of persisted records in a database
- Based on Hibernate
  - No JDBC coding
  - Annotations provide all database connectivity details
- Uses Java Object Messaging interface to Corticon
- Multi-threaded to best use Corticon concurrent execution architecture

High Performance Batch Processor – Architecture

- Decision service modeled in Corticon
- Java data classes mapped to database
- Data retrieved from database and payload created
- Request submitted to Corticon Server
- Response received and results updated in database



Demonstration

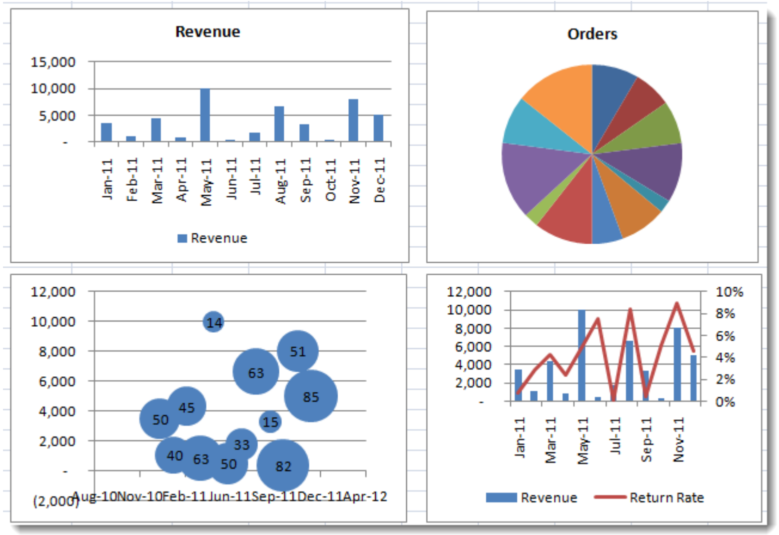
High Performance Batch Processor

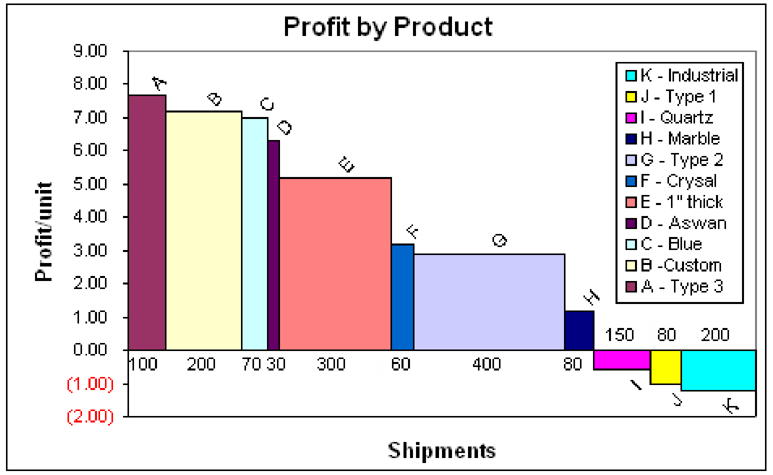
Benefits:

- Multi-threaded architecture ensures maximum throughput
- Can be architected to run across multiple machines
- Querying ability allows for only subsets of records to process
- Re-run against new rule models to allow “what-if” scenarios
- Decision outcome persisted to database for analytics

Disadvantages:

- Limited to RDBMS with Hibernate support
- Required Java Object Messaging and Hibernate skills





PROGRESS

Agenda

- Business Rules and Decisions
- Data Integration
  - Extended Operators
  - Service Call-Outs
  - Enterprise Data Connector
  - High Performance Batch Processor
- Wrap Up

PROGRESS