Talend Rocks

# Scene 1 Introduction Cam [Done]

Hello and welcome to Talend Rocks a podcast dedicated to data integration using products from Talend.

My name is John Tucker and I am a data integration consultant. In this edition I will be looking at the ESB or Enterprise Service Bus product.

To demonstrate the capabilities of the software I have developed a model scenario based on a hypothetical chain of supermarkets however the concepts apply to many other scenarios such as market research, accumulating sensor data, in fact anything that collects data from discrete locations prior to distribution throughout organisation.

For educational purposes a variety of techniques will be used which may overlap but in practice methods would be standardised.

# Scene 2 Retail POC cam [Done]

Let’s consider the proof of concept scenario.

A supermarket with stores across the British Isles wishes to return sale details to a central database via a regular polling mechanism. Each sale will comprise a varying number of products in differing quantities. The data will be used for management reporting, stock control and the population of financial and analytical systems. Minimal human interaction should be required to maintain the information flow.

In our scenario the local sales will be generated within the central database but in the real world they would be imported from local data stores.

So, what are we looking to achieve with the model?

# Scene 3 -PPT Slide 1 [Done]

(Narrate slide no intro)

# Scene 4 Sales Generation [Done]

I previously mentioned simulating communication polling and that is going to be achieved by developing a sales generation engine which can be called by each store in turn on a regular basis.

To approximate real world conditions a degree of raggedness must be introduced in the data otherwise values would be unrealistically similar and to avoid this situation, quantity, product and total parameters will be modified each time the procedure is called.

# Scene 5 PPT Slide 2 [done]

(Narrate slide no intro)

# Scene 6 Links to Other Information (Slide With Cam)

## 6a cam [done]

To keep this podcast accessible, I have tried to not delve too deeply into Talend specific technical details. Should you wish to gain further insight there are some additional resources available.

## 6b ppt slide 3 [Done]

First my Wordpress site, John Tucker Integration has a number of blogs, articles and tutorials related to the Talend platform.

There is a large and vibrant user community which can be accessed through Talend Forge.

Finally you will find numerous discussions, examples and tutorials on line, featuring both Talend and all the related technologies including Apache Camel, MySQL and other database systems, JMS and Apache Karaf, the runtime used by Talend. The list is endless.

There are a few concepts that it is worth taking a preliminary look at to help understand the Talend ecosystem.

# Scene 7 PPT Slide 4 Intro [Done]

Let’s consider Talend Routes.

They provide task instigation, conditional branching, and iteration functions to control the flow of data through transformational tasks.

They are not directly involved in transformations, handing the processing off to other tasks.

[Then Slide]

# Scene 8 PPT Slide 5 Intro [Done]

Routes of more precisely Talend ESB is built on the apache camel framework, providing a graphical development interface creating java code that uses the libraries from the framework, but what is Camel?

[Then Slide]

# Scene 9 PPT Slide 6 Intro [Done]

I previously mentioned JMS or Java Messaging Service. This is a mechanism to transport data by encapsulating it in a wrapper and moving through various locations, based on identification labels in the wrapper, known as meta data. A message can be moved to a destination, through various paths, without direct knowledge of the data. A good analogy is placing a letter in an envelope, writing on an address and placing it in a letterbox. The letter will arrive at the recipients address but only they and the sender have access to the contents, the journey has been executed using the address on the envelope. In this case the letter is the data, the envelope is the JMS message wrapper and the address is the metadata.

A few points with JMS:

[Then Slide]

The scope of a JMS message is the executing program, to keep that message available for subsequent processing, outside the scope, a copy needs to be kept in a persistent store. This store is known as a broker which stores the information in a FIFO or first in first out queue. Other programs can then perform the reverse

consuming data from the queue back into a JMS message.

# Scene 10 PPT Slide 7 Intro [Done]

Having seen how data is moved around by routes, let’s look at the other side of Talend, Jobs.

As part of the Extract Transform Load or ETL layer, jobs transform data changing it from input format to output.

Jobs can run as stand-alone tasks or they can be called from a route.

To reiterate.

[Then Slide]

# Scene 11 PPT Slide 8 Intro [Done]

We will begin with the Route that controls the data generation process to simulate order polling.

[Then Slide]

# Scene 12 PPT Slide 9 Intro [Done]

As the route runs, for each store it calls the job to generate sales which transforms the data via a series of steps.

[Then Slide]

# Scene 13 Review To Date Cam [Done]

We have had a brief introduction to the concept of Routes and Jobs within Talend. We also had a look at their use in the process to generate sales data. Now will be a good time to present an overview of where these procedures sit, within the proof of concept.

# Scene 14 (PPT Slide 3) -Done

Each store is processed in turn and this will be implemented using a talend route, called on a scheduled basis.

Traversing the stores, a sales generation engine, created using a talend job, will be called, with parameters set by the route, simulating the data polling.

Frequency of the operation is defined in a context parameter of the route.

Data is used to populate other systems throughout the organisation.

Stock control will be intrinsic to the process, a further series of jobs checking each sale against availability and interacting with supplier systems.

Sales data is fed into a financial ledger built on a SQL Server database .

A data warehouse also housed in SQL Server for BI Reporting.

Self Service analytics using power bi and Excel

And sales analysis using a no sql database

# Scene 15 Stock Maintenance Overview Cam [Done]

Having seen how the sales are generated now let’s consider another part of the process, stock control. When an order is made each sale line will consume several products and we must assume that they have been procured from a finite stockpile. As more sales are made the stock diminishes and at some point, will need to be replenished. This is termed the minimum re-order level. To check if this state has been reached, each order line is checked at individual product level. When minimum level is reached a purchase order for a defined number of units is e-mailed to the supplier. On receipt products will be dispatched together with a file containing their invoice, loaded into the system for invoice matching at a later point. In the real world this would probably be processed via EDI or B2B channels.

Unit cost price may vary by order. To be able to perform accurate margin calculations each batch of products is considered to be contained in a logical silo. Order stock is consumed from the active silo and each time the silo quantity is decremented. Once exhausted the next product silo becomes active. Silos are a logical concept only and individual products don’t have a real connection to the cost price but by saying a group of x products were purchased at cost price y, a realistic analysis by cost price is possible. This is often referred to as stock integration.

# Scene 16 Stock Maintenance PPT Slide 11 [Done]

Let’s consider the constraints of Stock Maintenance for this demo.

[Slide]

# Scene 17 Stock Control Diagram PPT Slide 12 [Done]

The diagram shows the stock control mechanism which runs asynchronously from order creation.

Consume messages placed in the JMS queue by the sales generation and parse into sales lines.

Find the product and active product silo records, then for each item

Decrement the silo quantity and if minimum order level has been reached, create a purchase order and e-mail to supplier.

Otherwise check if the silo is empty and if so, make the next silo for the product active.

# Scene 18 Stock Maintenance Design Principle PPT Slide 13 [Done]

I am going to reiterate the design principles of the stock control mechanism. Note that in the proof of concept, to assist clarity by limiting the scope, order fulfilment is instant. rather like a Star Trek replicator. In reality of course there would be lead-times to consider.

[Slide]

# Scene 19 Routes and Jobs Intro Cam [Done]

There are several Talend processes that define the proof of concept system and I have divided them into 4 groups:

Routes, the camel based mediation procedures that move data around.

Called, Standalone or JMS jobs which can be called by a route, run as a standalone package or wait for JMS messages to arrive in a queue. A lot of components used to create the last kind of job are from the message orientated middleware of MOM for short.

Rest or Soap Jobs which implement web services

And finally sub-jobs that can be called by other jobs.

# Scene 20 PPT Slide 14 [Done]

Lets model the interaction between jobs and routes in the 4 categories.

[slide]

First, we show routes that call jobs and the connection between them

We then have a route, in this case initiated by the existence of a file in a folder, which doesn’t call a job but writes to a queue instead allowing a job to be ran asynchronously.

Next there are the JMS or MOM jobs and links to any web services consumed.

Lastly the soap service jobs and a rest wrapper.

# Scene 21 Talend to AMQ PPT Slide 15 [Done]

JMS messages written to queues controlled by the ActiveMQ broker are used to communicate between several programs.

[Slide]

The generate sales job writes to the saleslog queue

The queue is consumed by the job to check product levels for stock control which writes data to the NoSQLSales Log queue

This message is then consumed by the job to write to MongoDB. A good example is created here of cross application connectivity. Although the operation to write to Mongo is created in a different version of Talend studio and run stand alone, there is seamless interaction with the data created by product level job running in karaf, the osgi runtime container.

In the final example the route to process order confirmation files from suppliers writes messages to the Supplier Order Confirmation queue. These are consumed by the job to book in supplier confirmations ready for invoice matching later.

# Scene 22 Database connections Cam [Done]

One final area I want to consider, before moving on to the runtime demonstration, is database connectivity. In this system connections are being made to 3 different systems:

MySQL used to create the central sales database

SQLServer used as an accounting ledger

And MongoDB for sales analysis

Let’s have a look at how the Talend processes are connecting to these databases.

# Scene 23 Database 1 PPT Side Slide 16 [Done]

[Slide]

The sales generation job has direct connection to the product, sales and sales line tables

Checking product level for stock control uses the Process Sales stored procedure to access the product and stock silo tables.

The web service Rest Get Stores has direct access to the store table.

# Scene 24a Database 2 PPT Side Slide 17 [Done]

A lot of the functions required for sales and stock control, centre around product information, and these have been encapsulated in a REST api

[Slide]

The various api calls can be seen here as they access various database tables either directly as in product data or through stored procedures such as the assigned supppliers call

# Scene 24b Database 3 PPT Side Slide 18 [Done]

[Slide]

The REST service shown here inserts data into a SQLServer product table via a SQLServer stored procedure

Then we have the SOAP service jobs which access sales, product and store tables via stored procedures

# Scene 25 MongoDB PPT Side Slide 19 [Done]

In MongoDB which is a NoSQL database the concept of records and tables is replaced by documents consisting of a key and a JSON value, stored in collections

[Slide]

Looking at the data jobs they read and write to the saleslog collection in the MongoDB database.

# Scene 25a Runtime Intro Cam [Done]

Having defined the goals for the proof on concept let’s have a look at it in action. Remember documentation and all source code can be accessed through my Wordpress site listed earlier in this video.

To reiterate this can be access via http 3.8.7.77.

# Scene 26 Set up Run Time data [Done]

Reset Data mysql, sqlserver and mongo

Display mysql

Display sqlserver

Display Mongo

Display ActiveMQ

# Scene 27 Standalone tasks [Done]

Task Scheduler

MOM bundle to write Mongo

# Scene 28 Runtime Setup [Done]

Karaf setup

Show deploy folder

Start runtime

Start Client

Show installed bundles

Show data sources

# Scene 29 Demonstrate changing db data [Done]

Display mysql

Display sqlserver

Display Mongo

Display ActiveMQ

# Scene 30 Gmail and Supplier Invoice [Done]

Show GMails

Show purchase orders and confirmations

# Scene 31 [Done]

Excel Soap demo

Power bi demo

# Scene 32 [Done]

Run mongo read in big data

Power bi and mongo a few hours later

# Scene 33 [Done]

Stop the runtime

# Scene 34 Final Sum up Cam [Done]

We have seen the proof of concept system in action, noting that it produced the sales data and combined it centrally before redistributing out to various business systems. We saw that it ran without user interaction, populating targets as required, achieving the design goals set out earlier in this presentation.

In the example mySQL and SQLServer were used for relational databases but these could have been substituted with other systems including Oracle, Postgres or cloud-based options such as Aurora, Azure maybe even Redshift for large organisations. The flexibility of Talend and its multitude of connectors make this possible. Equally MongoDB could be replaced by DocumentDB or a Mongo Atlas Cluster and there are SAAS products for BI such as snowflake that could easily be connected to.

Talend itself comes in a variety of options including a cloud based paas offering and there are trials of a managed version of the on-premise system hosted in AWS currently underway.

That brings us to the end of this podcast. My name is John Tucker. thank you for watching and see you next time on Talend Rocks.

Tutorial Intro [Done]

Hello, my name is John Tucker and I am a data integration consultant. In this video I will be looking presenting a tutorial demonstrating how to write data to MongoDB using Talend.

For this example I will be using the community edition of Talend open studio big data edition, together with a local edition of MongoDB

Further Talend related examples, tutorials and blogs together with sample source code, is available via my wordpress site at <http://3.8.7.77>