

Both Ends Meet in the *MiddleWare*

Presented by:

Bill Galloway – Canadian Forest Products

**PIMA – Spring 2003
Vancouver**



Agenda

- Background
- Our Middleware Decision
- Implementation Challenges & Solutions
- Middleware Architecture
- The Interface Process
- Configuration Decisions
- Speed & Reliability
- Audit, Errors and Controls
- Benefits Realized
- Future Development
- Q&A

Canfor Background

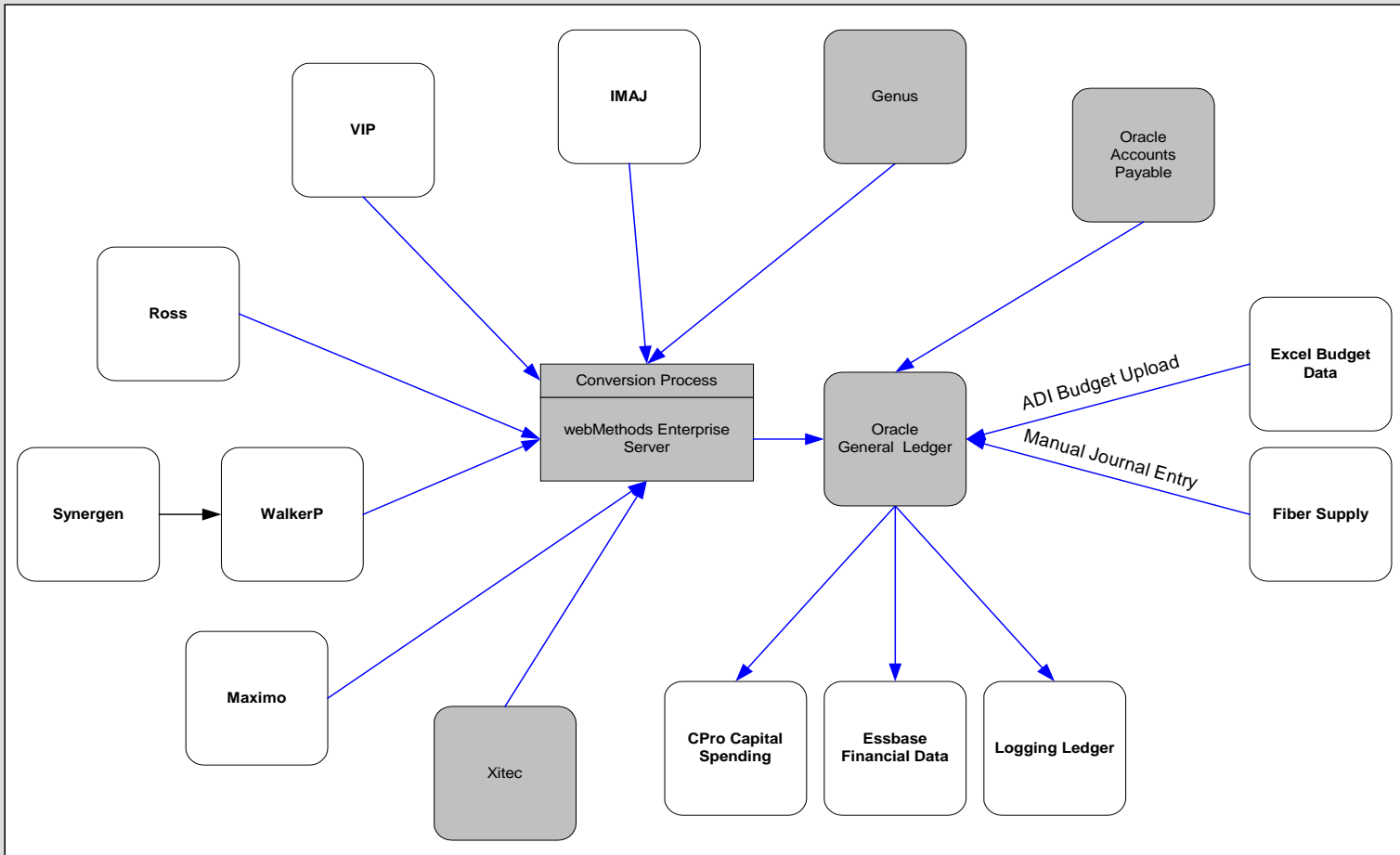
- Largest finished lumber producer in Canada
- Mainframe Walker GL/AP for 18 years
- Need to consolidate systems on core platform
- Oracle Financials GL, AP and cashed cheques



Why Middleware ?

- Feeder systems on different platforms
- Multiple variations of Old Chart of Accounts
- Vendor files in separate feeders
- Feeder system priorities and timelines
- Fixed go-live date

Why Middleware ?



Our Middleware Decision

- Strategic Direction for I.T.
- Minimize number of tools purchased, maintained, and supported (B2B and for E.A.I)
- Greater focus on business rather than technical issues of interfacing.
- Integration required for processes as well as systems.
- Aim to leverage common skills across interfaces and projects.

Less “reinventing the wheel”!



Implementation Challenges & Solutions

Challenge

- 80 byte card images
- Waiting for feeder systems' change could delay entire Oracle project

Solution

- Middleware decided which files required mapping routines
- Feeder system owners had responsibility (and timeline) for making changes in feeders.
- Isolate Oracle project from feeders

Implementation Challenges & Solutions

Challenge

- Multiple Charts of Accounts to be replaced by one in Oracle

Solution

- Oracle Financials used as lever to standardize Chart of Accounts
- Lookup table for old to new codes – 100,000 entries
- Middleware does same translation – for both GL & AP
- Middleware moves feeder account code to interface tables

Implementation Challenges & Solutions

Challenge

- Feeder vendor numbers required translation to Oracle vendor #.
- Feeder purchasing systems could not convert to Oracle vendor numbers

Solution

- Oracle Financials used as lever to cleanup & standardize vendors
- Middleware logic derives Oracle vendor id
- Middleware logic moves feeder vendor number to AP interface table

Implementation Challenges & Solutions

Challenge

- Feeder systems used IBM EBCDIC with right-hand overpunch, but new systems use ASCII
- (As if: Feeders spoke Mandarin; Oracle expected French!)

Solution

- One routine in Java to convert any numeric
- ASCII accepted also
- Logic in middleware decided which, and took appropriate action
- All GL & AP feeds convert \$ amount, statistics
- Headers & Details

Implementation Challenges & Solutions

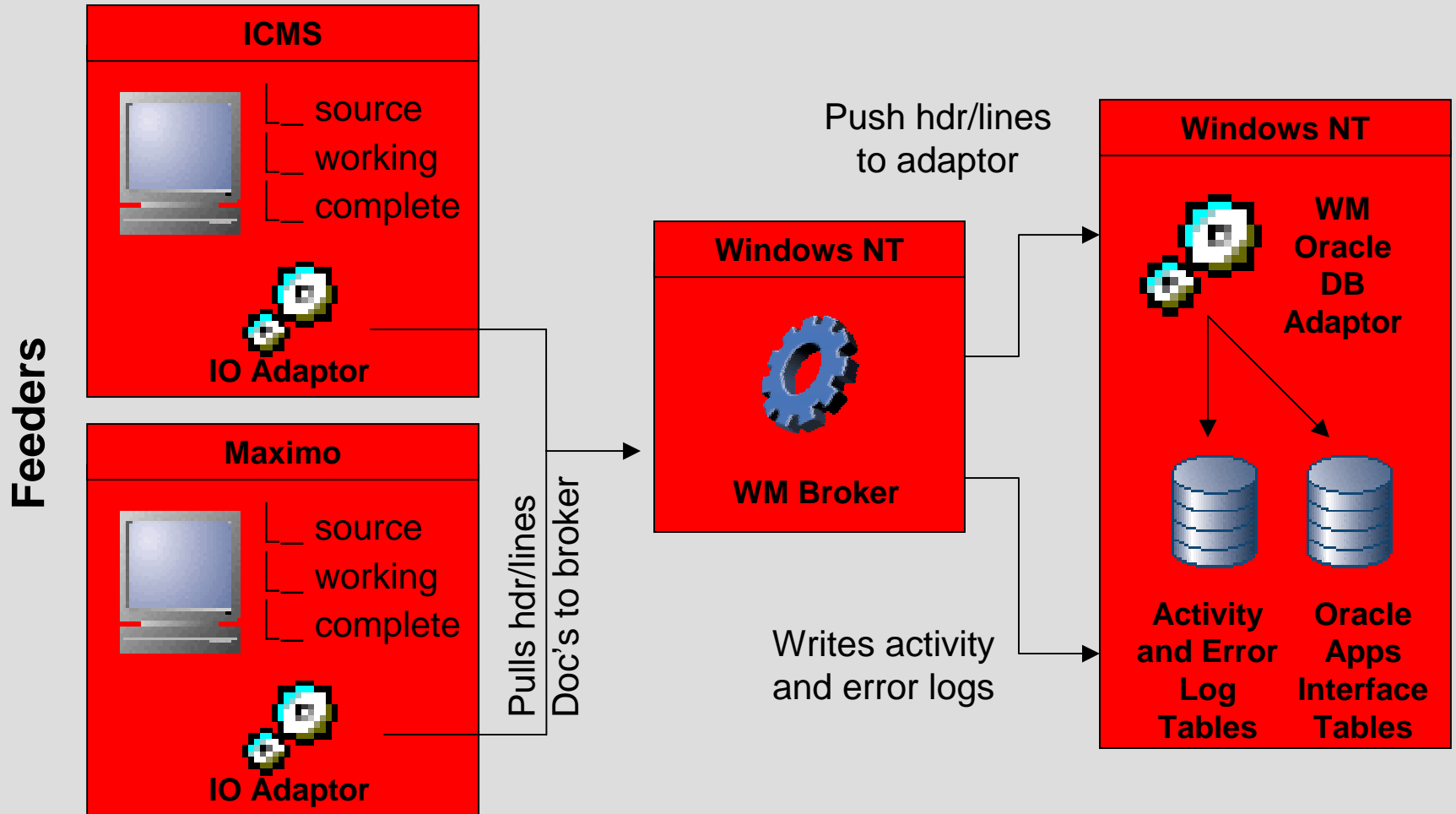
Challenge

- Terms code translation required
- Missing information (Source & Category)

Solution

- Custom table created for terms code lookup
- Config file for each feeder system used to derive missing information

Middleware Architecture



Middleware Instances

- 3 Instances of Middleware - Dev, QA, Prod
- 2 Separate servers
- Failover from Prod to QA
- Changing Middleware instance by broker pointer
- Easy to stop Middleware & re-start by shutting adapter

Key Decisions

- Adapter Type – XML, IO, Java, custom
- Where to place adapters
- What stage to trap errors
- Multi-Publish document, or single transaction
- Single hidden User id
- Separate integrations versus common process

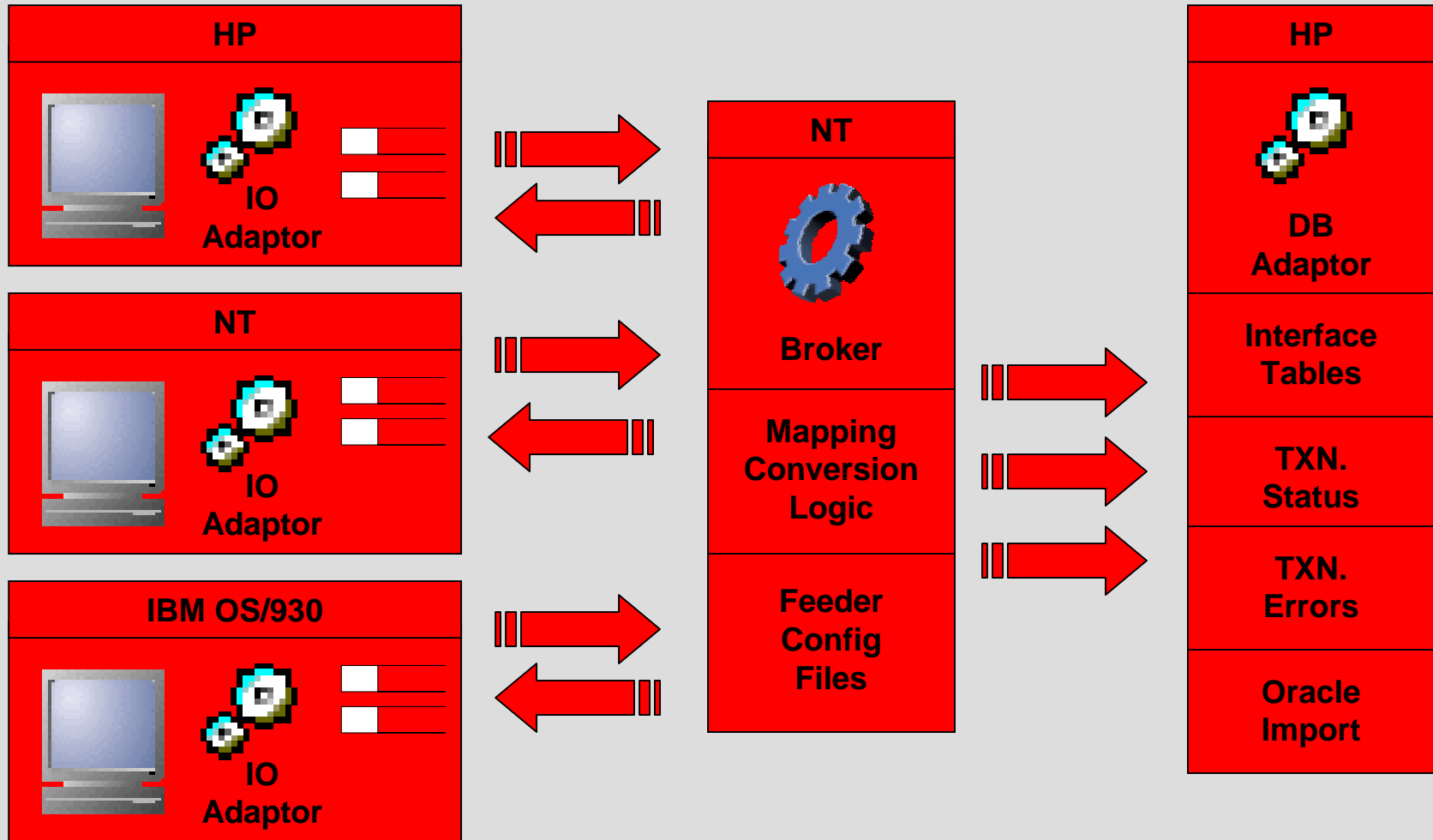
The Interface Process

1. Feeders place transaction file in designated directory
2. Middleware Adapter continuously scans directory for existence of transaction file
3. Adapter feeds transactions into integration broker
4. Broker performs mapping & conversion logic
5. Broker moves file to a 'Completion' sub-directory

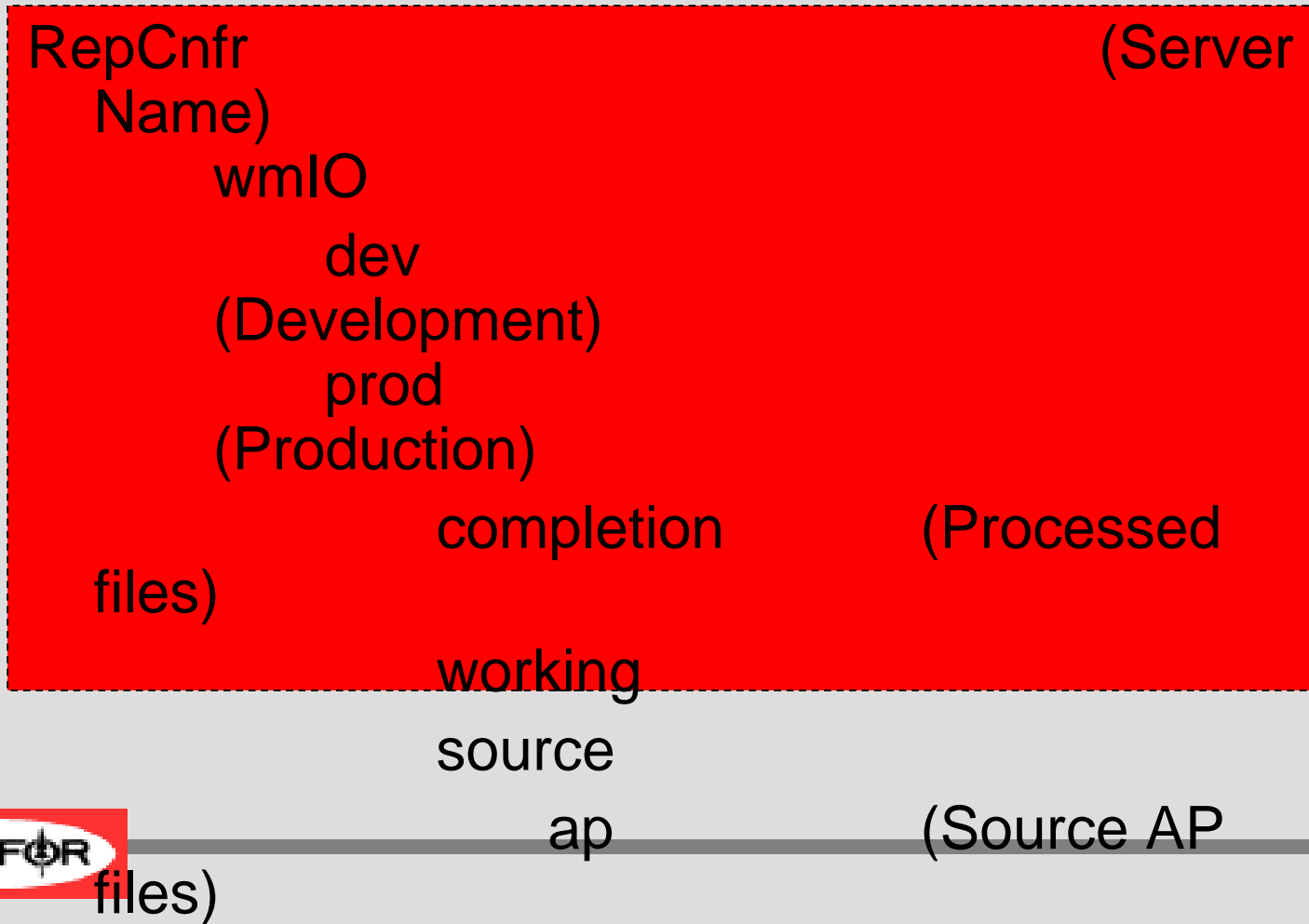
The Interface Process

6. Publishes transaction to Oracle Financials interface tables
7. Custom Table for Transaction Status
8. Custom Table for Errors
9. Concurrent request to import/approve txn's into Oracle
10. Config file supplies missing information

The Interface Process



Directory Structure for any Feeder



Transaction Logging

- Each batch is logged in and out
- Record counts must match
- File Name is stored going into Oracle Apps
- Log Table e-mailed to Central services daily

Transaction Record

Application: GL

Group ID: 20027297593877

Source: MAXIMO 10 0 0 0

File Name : d:\wmIO\prod\working\maximo_gl_621

7/29/2002 7:59:52AM DBSUCCESS

7/29/2002 7:59:38AM IOSUCCESS

Error Logging

- Control of access to Oracle Financials
- Daily reports to Central Services by e-mail.
- Option to e-mail status to feeder system keepers
- Review the actual batch in the completion folder

Error Record

Application: GL

Responsibility: WMSA,DBA

Source: IMAJ

Group ID: 200272118617405

Time Stamp: 7/21/2002 6:06:14PM

Error Code: 305

Data File Name: D:\wmlO\prod\working\GL_634

Error Message: Not invoke ComTgtDBOraGetBookID

Reliability

- Transactions are picked up within seconds
- Biggest batch is 17,000 rows – 7 minutes
- No Collisions
- No scripts to ensure transaction updates
- Guaranteed Delivery: One evening ...

Test Methods

- Very simple to place a text file in a directory
- Same transactions to heritage & new system
- Very simple to add new feeds
- Spreadsheet application does Summary Billing
- Can mix and match new and old Chart of Accounts in same batch

Benefits Realized

- Implementation on time and budget
- Technical issues and updates for all feeds resolved once
- Solid base for enhancements and upgrades
- Easy to test, both unit and functional
- Easy to add new feeds
- Good transaction logging



Future Development

- Synchronous processes: e.g. Account code validation.
- Employee/user id maintenance
- Vendor set up
- Electronic vendor invoices

Questions?

Bill Galloway

Canfor Corporation

bgalloway@mail.canfor.ca

Tel: 604-661-5327

