Software Group

Guide Share France
Groupe de Travail MQ
janvier 2011

Carl Farkas
SW IOT TechWorks zWebSphere Application Integration Consultant
IBM France D/2708
Paris, France
Internet : farkas@fr.ibm.com

© 2010 IBM Corporation

Agenda

- Annonces
- Alerts
- AMS experience

© 2010 IBM Corporation

FTE 7.0.4 beta

- Enhanced integration with IBM Sterling Connect:Direct to enable file transfer across MQ and Connect:Direct domains

Faites-moi signe si vous souhaitez participer à ce beta

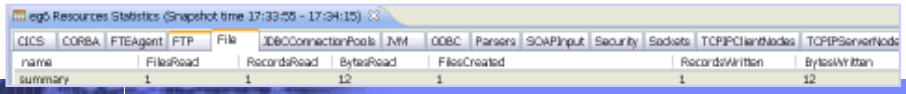
© 2010 IBM Corporation

WMB 7.0.0.3 (FixPack 2)

© 2010 IBM Corporation

New File Read Nodes and Other Enhancements

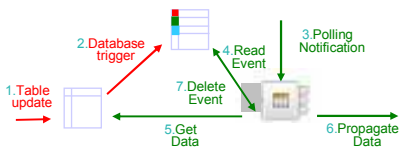
- **New File Read Node**
  - Get a single record from a file, or whole file, in the middle of a message flow
  - Typical scenarios include in-flow transformation and routing
    - ✓ e.g. Web service request identifies file to be transformed
    - ✓ e.g. Route MQ message based on file table data
  - Combine with other MB nodes including FTE for sophisticated 'in-flow' processing
  - Significant addition to existing local, network, SFTP, FTP, and MQFTE file support
- **Supports Advanced Features for in-flow file processing**
  - Dynamic file identification allows file read to change on per request basis
  - Define where record starts, ends and where to place result in message tree
  - Records parsed as per file input node, fixed, delimited and parsed
  - Includes 'Read by key' allowing user to determine exact record e.g. `\record\field4='special'`
  - 'Read by byte offset' allows user to locate exact position in file
  - Extracted data (including partial record) can be placed anywhere in propagated output message
  - Stream based processing means whole record is not kept in memory
  - Disposition of none, delete, rename, archive when file finished processing
- **File Node Enhancements, including FTP**
  - 'Skip First Record' simplifies CSV header records
  - (S)FTP Server timeout configurable service
  - Resource Statistics now available to understand file processing



© 2010 IBM Corporation

Database Input Node Enhancements

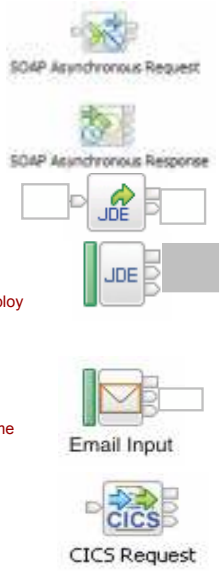
- **Database Input node**
  - Allows database tables to be treated as input source for message processing
    - ✓ Selection criteria include multiple tables, complex joins, and other database oriented semantics
  - Changes (Insert Update, Delete) occur to database tables; database trigger records in Event Table
    - ✓ Polled queries from database start a message flow: design allows for future triggered starts
- **Code-free Query Generation Enhancements**
  - Development tools complete database queries without user requiring any SQL knowledge
    - ✓ Power users can exploit custom ReadEvent, BuildMessage & EndEvent SQL routines
  - Table schemas (XSDs) automatically generated for 3rd party tooling imports
- **Extended Database support**
  - Full range of support across all supported databases
  - DB2 & Oracle augmented with Informix, Sybase, SQL server and solidDB
- **Exploit existing Event Tables Technologies**
  - Standard usage sample
    - ✓ Learn how to use database node with database trigger
    - ✓ Illustrates basic & extended usage scenarios
  - WBIA JDBC Adapter migration sample
    - ✓ Replace MQ input node with database input node
    - ✓ Reuses existing WBIA event tables
    - ✓ Preserves existing message sets



© 2010 IBM Corporation

New Connectivity Nodes and Updates

- **SOAP/JMS transport & HTTP Enhancements**
  - Now support SOAP/JMS async request and response nodes
    - ✓Message flow thread not held during external SOAP/JMS invocation
  - SOAP/JMS operations (optionally) now included in message flow transaction
    - ✓e.g. Transact SOAP/JMS request & response with database update
- **JDEdwards**
  - Connects JDE systems to wider enterprise applications
  - Built-in input and output nodes exploits JDEdwards JCA adapter
  - Complements existing SAP, SEBL, PSOFT ERP nodes
  - Typical scenarios include MQ, File, Web Services <->JDE and SAP, CICS, IMS<->JDE…
  - Contains operational sophistication of these ERP nodes, e.g. incremental discovery and deploy
- **Email Input node**
  - Supports processing email input from common email systems
  - Various candidate protocols (POP3, IMAP)
  - Complements existing email output node
  - Input email properties can be described at design time, and overridden dynamically at runtime
- **CICS Node Enhancements**
  - 3 tier topologies (MB->CTG->CICS) now supported for advanced HA and WLM options
  - Channels and containers now supported with full built-in sample
    - ✓Supports >32K COMMAREA, different model to COMMAREA
    - ✓Easy to construct using CICS node tooling and/or runtime collections
- **TCP/IP Node Enhancements**
  - Enable SSL for TCP/IP nodes – for secure socket connectivity inbound and outbound
  - SSL Security populates `LocalEnvironment` to enable client filtering scenarios



© 2010 IBM Corporation

Alert

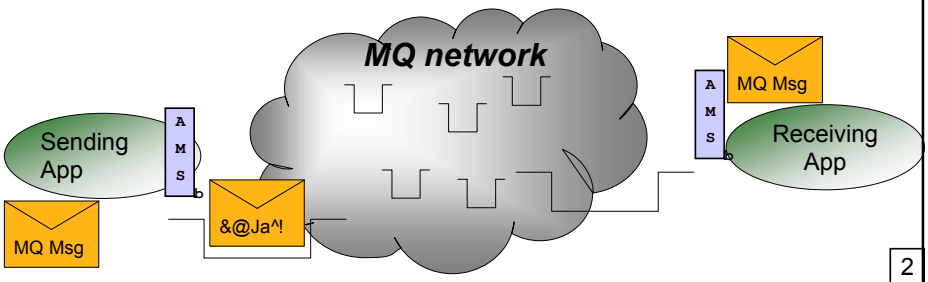
- Possible CPU increase on MQ z/OS CHIN with migration to v7.0.1
  - Customer had a trace enabled on acctg class, and after migration, additional SMF 116 records produced and CPU increased
  - Turning off trace reduced CPU by 1/3
  - Still investigating

© 2010 IBM Corporation

What is MQ AMS?

WebSphere MQ Advanced Message Security V7.0.1

- New product, announced and available Oct 5, 2010
- Provides security for MQ messages, end-to-end with no application changes
- It is a simple “add-on” product that enhances WebSphere MQ v6 or v7
- Security policies are used to define the security level required which leverage X.509 certificates



2

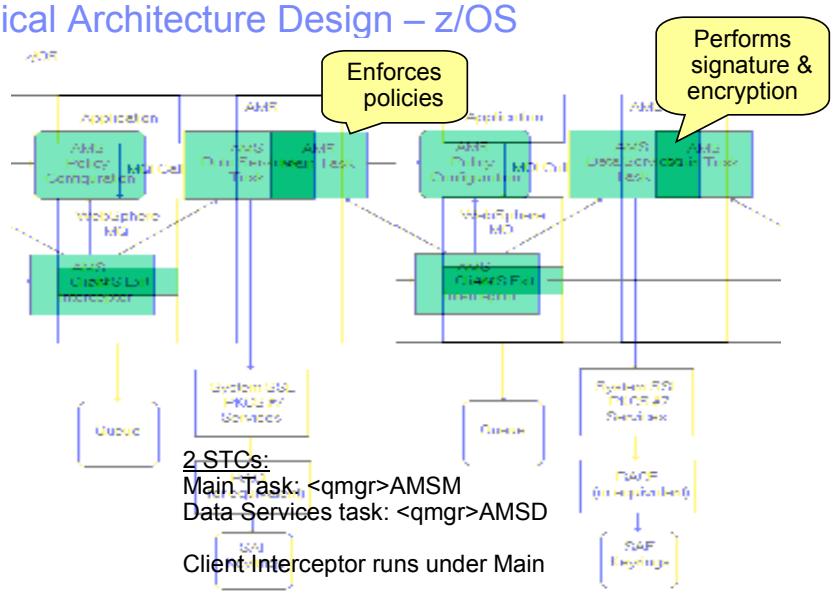
© 2010 IBM Corporation

AMS Key Features

- Secures sensitive or high-value MQ messages
  - Privacy via message content encryption
  - It leverages digital certificates (X.509) and Public Key Infrastructure (PKI) to protect MQ messages
- Detects and removes rogue or unauthorized messages before they are processed by receiving applications
  - Authentication via certificate *above and beyond* operating system
  - Authorization to queue *above and beyond* MQ OAM or SAF
- Verifies that messages are not modified in transit
  - Message Integrity via digital signature of message content
- Protects messages not only when they flow across the network but when they are at rest in queues
- Messages from existing MQ applications are transparently secured using “interceptors”
  - No application changes are necessary
- No pre-requisite products other than MQ
- Replaces (and upward compatibility) with MQ Extended Security Edition (ESE)

© 2010 IBM Corporation

Logical Architecture Design – z/OS



2 STCs:
Main Task: <qmgr>AMSM
Data Services task: <qmgr>AMSD

Client Interceptor runs under Main

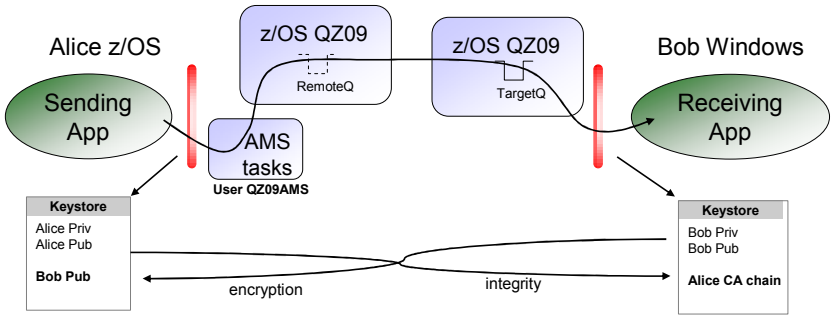
© 2010 IBM Corporation

AMS z/OS installation

- SMP/E installation
  - Program 5655-W50, FMID HMS7701
  - Requires 120 tracks (Target), 150 tracks (Distribution), 1MB (zFS)
- Post-installation tasks
  - Update LPA for AMS modules in SDRQLINK
  - Update Authorized Program Facility (APF) list for SDRQLOAD
  - Update Program Properties Table (PPT) update
  - Update Product Registration list (IFAPRDxx)
  - Possible update to DIAG member for allocating in user storage key
- Create AMSM & AMSD procedures for the two AMS STCs
- Create profiles for STCs
- Note: userids that will be putting & getting protected messages will require:
  - An OMVS segment associated with their userid (or set default with FACILITY class, BPX.DEFAULT.USER)
  - SAF UPDATE permission for the FACILITY class, IRR.DIGTCERT.LISTRING

© 2010 IBM Corporation

WebSphere MQ AMS –config z/OS -> Windows



1. Install AMS Interceptor
2. Configure AMS
3. Create keystores with public / private key pairs
   a) copy sender's CA key chain to receiver's keystore for integrity
   b) copy receiver's public key to sender's keystore for encryption

2

© 2010 IBM Corporation

RACF on z/OS (sender side, aka Alice=FARKAS)

Data task on sender uses CA to validate

PUTer userid on sender uses certif to sign

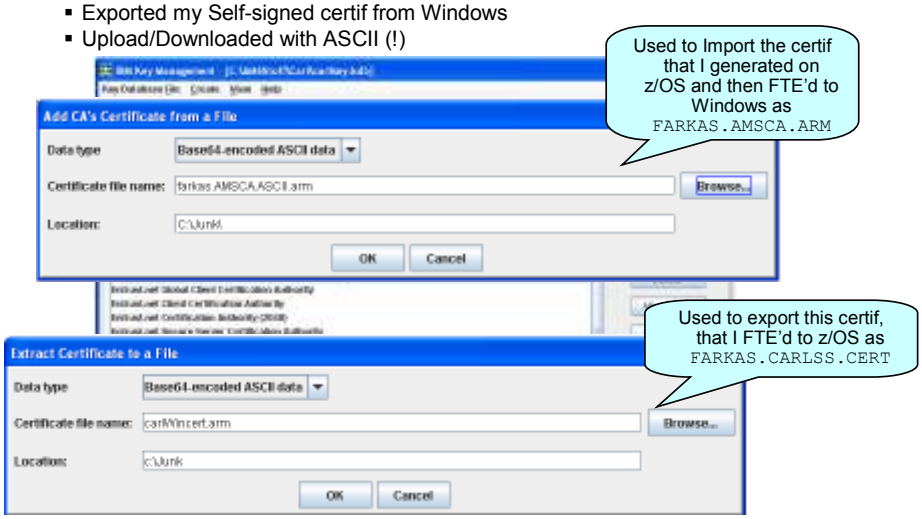
Export this CA so others can check my certif

```
//SYSTSIN  DD *

 RACDCERT ID(QZ09AMS) ADDRING(drq.ams.keyring)

/* Create a CA certificate good thru my retirement :> */
RACDCERT CERTAUTH GENCERT -
  SUBJECTSDN( CN('AMS CertAuth') O('IBM') ) WITHLABEL('AMSCA') -
  KEYUSAGE(CERTSIGN) TRUST NOTAFTER(DATE(2018/12/31) )

/* Connect the CA certificate to the AMSD Data task STC */
RACDCERT ID(QZ09AMS) CONNECT(CERTAUTH LABEL('AMSCA') -
   RING(drq.ams.keyring)

 RACDCERT EXPORT( LABEL('AMSCA') ) CERTAUTH FORMAT(CERTB64) -
    DSN('FARKAS.AMSCA.CERT')

/* Make a keyring for a userid that will be a MQPUTer or MQGETer */
RACDCERT ID(FARKAS) ADDRING(drq.ams.keyring)

/* Create a Certificate for a MQPUTer or MQGETer id */
RACDCERT ID(FARKAS) GENCERT -
 SUBJECTSDN( c('FR') O('IBM France') CN('Carl on Z') ) -
 WITHLABEL('CARLONZ') SIGNWITH(CERTAUTH LABEL('AMSCA') )  -
 NOTAFTER(DATE(2018/12/31) ) -
 KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

  RACDCERT ID(FARKAS) CONNECT(ID(FARKAS) LABEL('CARLONZ') -
 RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))

 SETROPTS RACLIST(FACILITY) REFRESH
/*
```

© 2010 IBM Corporation

iKeyMan on Windows (receiver side, aka Bob)

- Imported Signer certif into Windows keystore
- Exported my Self-signed certif from Windows
- Upload/Downloaded with ASCII (!)



© 2010 IBM Corporation

RACF on z/OS (sender side, aka Alice, part 2)

Import the Public certif of the Receiver so I can encrypt using this

```
//SYSTSIN  DD *

   RACDCERT ID(QZ09AMS) ADD('FARKAS.CARLSS.CERT') TRUST -
     WITHLABEL('CARLSS')

   RACDCERT ID(QZ09AMS) CONNECT( ID(QZ09AMS) LABEL('CARLSS') -
     RING(drq.ams.keyring) USAGE(SITE) )

  SETROPTS RACLIST(FACILITY) REFRESH
/*
```

© 2010 IBM Corporation