



IBM SOA

WebSphere Transformation Extender

WebSphere's Universal Transformation Engine for the Enterprise



Saad BENACHI

WebSphere Tech Sales
Saad.benachi@fr.ibm.com

ON DEMAND BUSINESS™

Agenda

- ▶ Présentation générale de WebSphere Transformation Extender (TX)
 - Transformation universelle
 - Fonctionnalités clés
 - Historique
 - Valeur apportée à nos clients
- ▶ WebSphere TX Architecture & Déploiement
- ▶ Comment développer une interface WebSphere TX
- ▶ Conclusion



Qu'est-ce que IBM® WebSphere® TX fait?

Lecture des données dans leur format natif

TOUTES structures des données



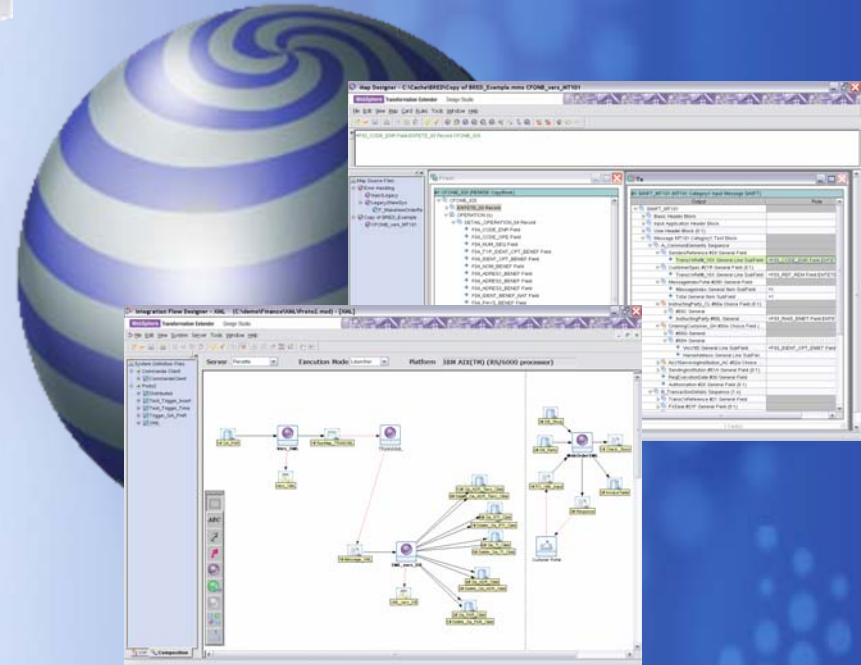
Transformation et routage sans programmation

Intégration N à N

Ecriture des données dans leur format cible



TOUTES structures des données



Ex. - Hierarchical Data , Binary Data, Packed Data, Tabular Data, Relational Data, Nested Structures, Mixed-Type Data, and on and on...

Vision de Transformation - Simplifié

- Transformer - tout
- Transformer - partout
- Transformer – tout le monde



What is the problem of Transformation

Processing Information with all the business rules and usage mechanisms in tact...

BINARY

```
00011110010010011010010
10010010010010010000011
11010100101010110010010
10010010010010001010010
10010010101010101000100
11100010100010010001001
00100100100100101010010
01010100100010010010010
01001001110001010010101
01010101010010010101001
00100010010100101000010
10101010001001010001001
01001010100101010100101
01100101000000000001111
10010010010101110010010
010101001010101101111
```

TABLE

Make	Model	PKG	Extended_Features
Ford	Prefect	34890	2984782g, 93847920, 3438084
Ford	Prefect	34890	2984782g, 93847920, 3438084
Ford	Prefect	34890	2984782g, 93847920, 3438084
Ford	Prefect	34890	2984782g, 93847920, 3438084
Ford	Prefect	34890	2984782g, 93847920, 3438084

COPYBOOK

```
01 TP-API-CB
03 TP-API-CB
05 TP-API-REQUEST PIC X(40).
88 TP-API-INITIALIZE-REQUEST
  VALUE INITIALIZE-MAPPING.
88 TP-API-PERFORM-MAPPING
  VALUE PERFORM-MAPPING.
88 TP-API-FINISH-MAPPING
  VALUE FINISH-MAPPING.
05 TP-API-VERSION PIC X(04).
88 TP-API-VERSION-VALID VALUES ARE '0100' '0200'.
88 TP-API-VERSION-0100 VALUE '0100'.
88 TP-API-VERSION-0200 VALUE '0200'.
05 TP-API-RESPONSE.
10 TP-API-RESPONSE-CODE PIC 9(04) COMP.
88 TP-API-ALL-OKAY VALUE 0.
88 TP-API-REQUEST-ERROR VALUE 1.
88 TP-API-INITIALIZE-ERROR VALUE 2.
88 TP-API-MAP-ERROR VALUE 3.
88 TP-API-FINISH-ERROR VALUE 4.
88 TP-API-UNKNOWN-LOOP-ID VALUE 5.
88 TP-API-NO-ALGORITHM VALUE 6.
88 TP-API-NO-PARTNER VALUES 7 15.
88 TP-API-NO-APPLICATION VALUE 8.
88 TP-API-ALGORITHM-IO-ERROR VALUE 9.
88 TP-API-FATAL-GATEWAY-ERROR VALUE 10.
88 TP-API-GATEWAY-WRITE-ERROR VALUE 11.
88 TP-API-PARTNER-IO-ERROR VALUE 12.
88 TP-API-BAD-VERSION VALUE 13.
88 TP-API-BAD-NUMERIC-TYPE VALUE 14.
88 TP-API-NO-ALIAS VALUE 15.
88 TP-API-ACCESS-ERROR VALUE 16.
10 TP-API-RESPONSE-MESSAGE PIC X(80).
05 TP-API-APPLICATION-ID PIC X(10).
03
01
```

CASH RECONCILIATION
27 files

```
10029847 - $100,000,000.00
13948589 - $679,495,094.98
13950967 - $588,345,058.00
13950968 - $000,000,000.00
14001321 - $098,957,038.12
```

AAA: HT4459
AAA: B33566
AAA: C4058G
AAB: 948409
AAB: 874931

STREAM

```
OH,257*IN,142*MI,1
54*WI,80*MT,5*ID,8
*WY,3*CO,21*NM,8
*AZ,15*UT,13*NV,4*
MN,48*MO,67*ND,9
*SD,9*KS,27
```

PROPRIETARY

```
DUNS 0123
F046000INV for 2
months 120799
12 718-339-1700143989D-2
1207999999-b
003000010000
DUNS 4445 P55590
120799499QR
000004004000
```

EBCDIC

```
ãÖ@Ô...™fll£-
™z@Ã-
•¥...™£@"...@£-
@ÀÂÃÄÉË
```

DDA Application Updates

```
<MSG 19934749>
<ACCT BAL RPT>
<ACCTNUM><"14001321">
<ENDBALANCE><" $098,957.038.12">
<DEBIT><"103048382$394,394.00">
<DEBIT><"103048383$001,293.65">
<CREDIT><"987463921$928,943.67">
</ACCTNUM>
</MSG>
```

413 msgs



Qu'est-ce qui rend WebSphere TX différent?

Moteur unique,
Multiples
modes d'exécution

Validation de
données
automatique
au cours de
traitement



Développement
graphique
sans
codage manuel

Transformations
complexes
de forts volumes

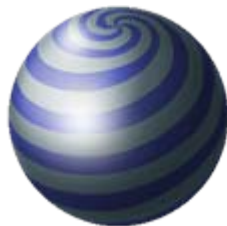
Supporte
toutes structures de
données
en format natif



Un Moteur, Multiples Méthodes d'Invocation

- WebSphere TX peut être déployé dans les environnements divers

- ESB
- SOA
- Web Services
- SOAP
- EJB
- HTTP
- Event driven
- Batch
- Java
- C/C++
- COBOL
- Unix
- Linux
- Windows
- z/OS
 - Batch
 - CICS
 - IMS
 - USS
- 30+ Adapters
- ETL
- Data quality
- Enterprise Applications
- Application Servers
- WebSphere Message Broker
- WebSphere ESB
- WebSphere Process Server
- WebSphere Partner Gateway
- WebSphere Application Server
- WebSphere DataStage



WebSphere TX Packs et Accélérateurs



Financial Services

- **SWIFTNet**
- **SWIFTNet Funds**
- Many Services Based Solutions
 - **SEPA**
 - FIX
 - FedWire, CHIPs
 - NACHA
 - BAI, BAI2
 - AL3
 - ACORD
 - Etc.



Health Care

- HIPAA
- NCPDP
- HL7
- Many Services Based Solutions
 - HealthCare Hub
 - Payment Processing
 - Clearing House Processing



CPG, Manufacturing

- ERP/CRM Integration
- EDI (ANSI X12)
- TRADACOMS
- ODETTE
- **EDIFACT**
- Many Services Based Solutions

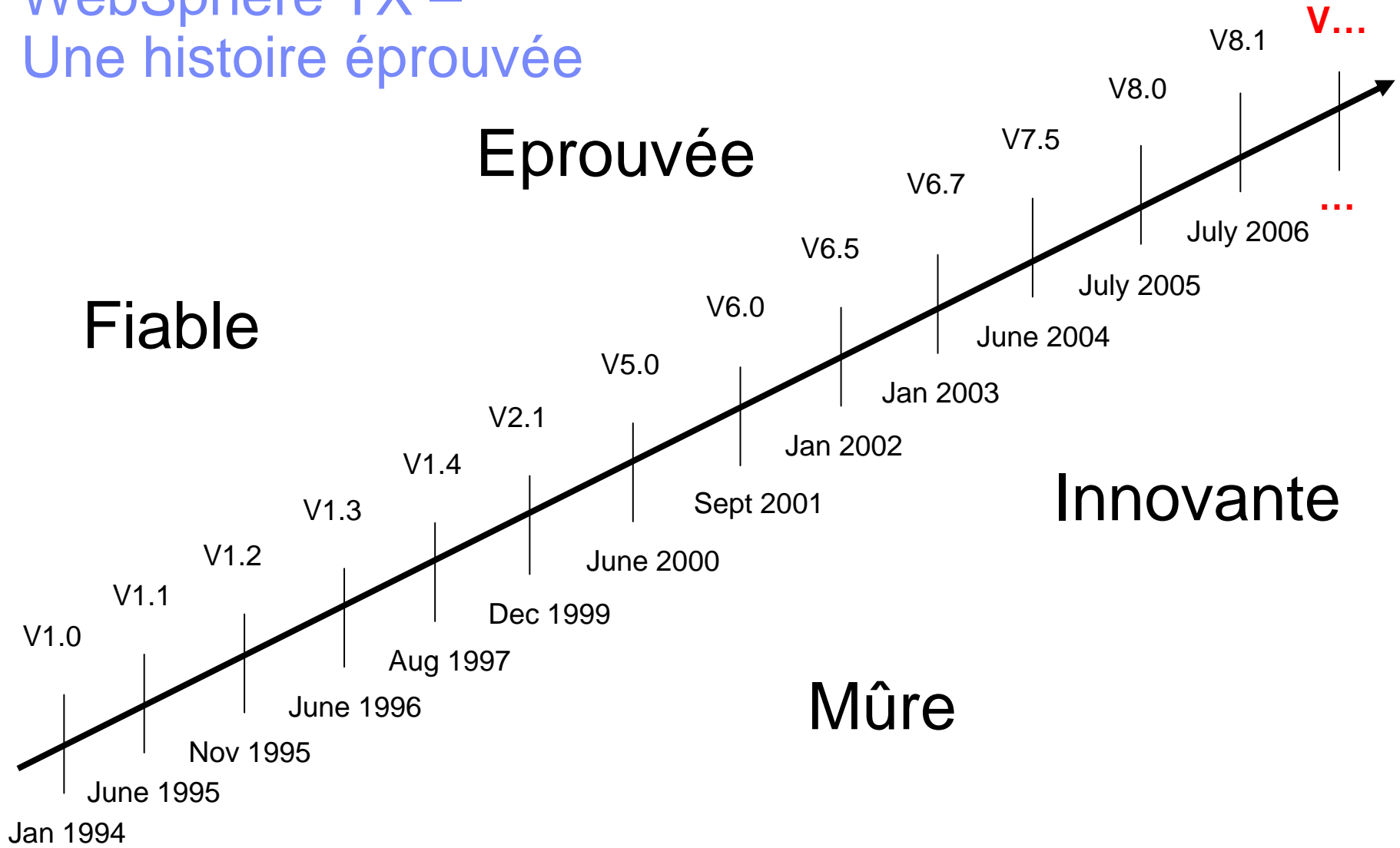


Embedded Software

- Integration to the Enterprise as an embedded service
- Telecom
- Enterprise Apps
- Shop Floor Systems, etc...



WebSphere TX – Une histoire éprouvée



Gestion des flux importants...

- On average, over U.S. \$3T* in transactions flows through WebSphere TX on a daily basis
 - ▶ Major Stock exchanges
 - ▶ 9 of 10 Major U.S. Financial Firms
 - ▶ 75% of BC/BS claims
 - ▶ 6 of 10 Major Insurance Firms
 - ▶ 3 of 3 Top U.S. Pharmacy Benefits Providers
 - ▶ 6 of 6 Top Global Banks
 - ▶ 6 of 10 Top U.S. Banks, 4 of 5 UK Top UK Banks
 - ▶ Major Global and National Telecoms
 - ▶ and on and on and on....

* \$1.6T is based on available, tangible measurements. Actual estimates are in excess of this amount.



Références WebSphere TX

FINANCIAL SERVICES



A Passion to Perform.



MANUFACTURING, RETAIL & DISTRIBUTION



AUSTRALIA POST



JOHN DEERE



HEALTHCARE / INSURANCE



Premera Blue Cross



SAFECO

Anthem.



EXPRESS SCRIPTS

MUTUAL of OMAHA



Group Health Cooperative

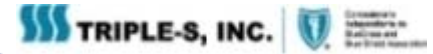


Blue Cross Blue Shield of Rhode Island



...et sur système Z

WTX z Customers



Zurich Financial Services



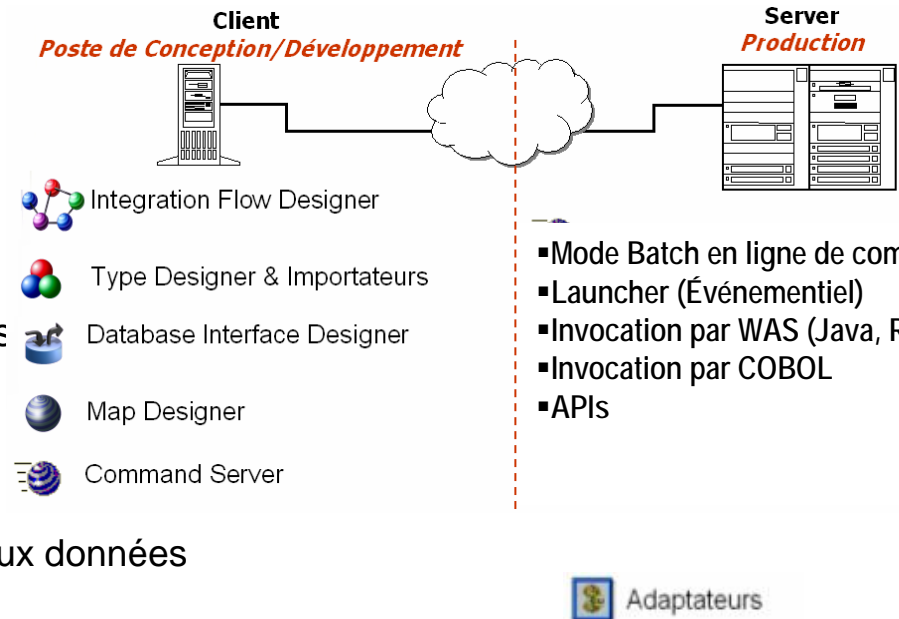
OKO Bank



nco Comercial Português SA



Du développement à l'exploitation...

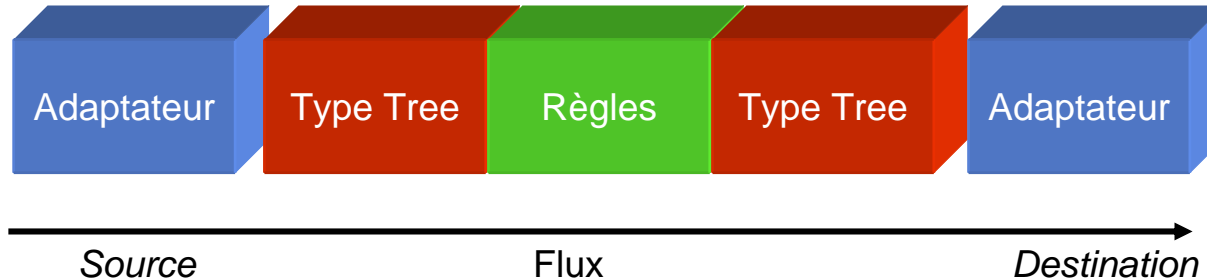


- Développement : le « Design Studio »
 - ▶ Outil totalement graphique sous Windows
 - ▶ Définition des :
 - Structures de données à manipuler
 - Transformations (interfaces) à assurer
 - Adaptateurs techniques pour accéder aux données
 - ▶ Exécuter les tests unitaires
- Exploitation : le « Transformation Server »
 - ▶ Moteur qui exécute les interfaces définies dans le Design Studio
 - événementiel Temps réel et/ou batch
 - ▶ Fonctionne sous Windows, Unix (AIX, Solaris, HP ...), Z/OS...



Une brève introduction de terminologie

- Un processus le plus simple se compose comme suit:

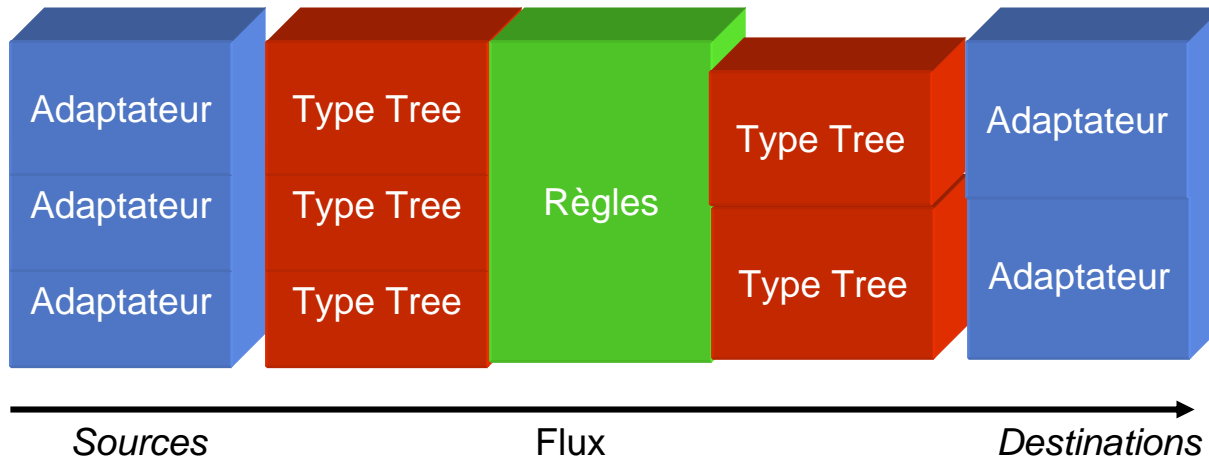


- ▶ C'est-à-dire...
 - ▶ Un adaptateur est un connecteur technique qui peut être pluggé à une application (source & destination) - sans parsing!
 - ▶ Un Type Tree est une représentation graphique de méta-données
 - ▶ Les règles définissent comment une Source est "transformée" à une Destination
- ▶ L'ensemble constitue une Interface ("Map")

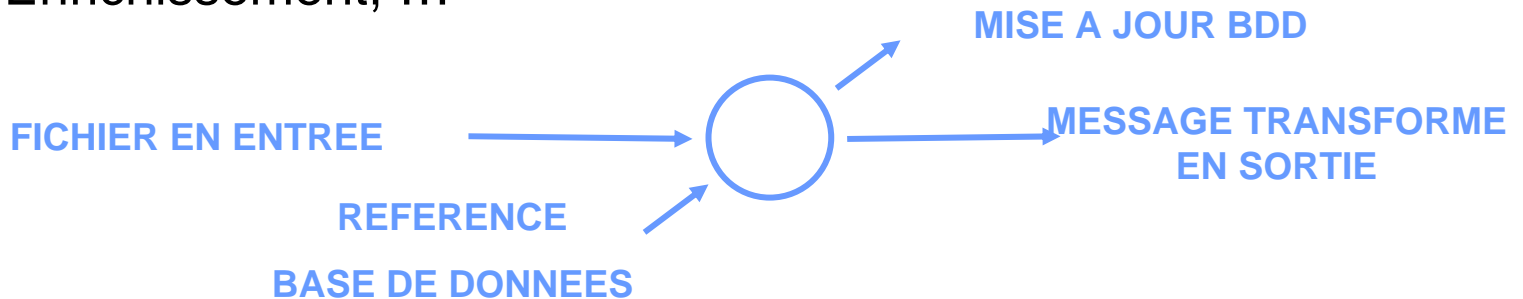


Une brève introduction de terminologie

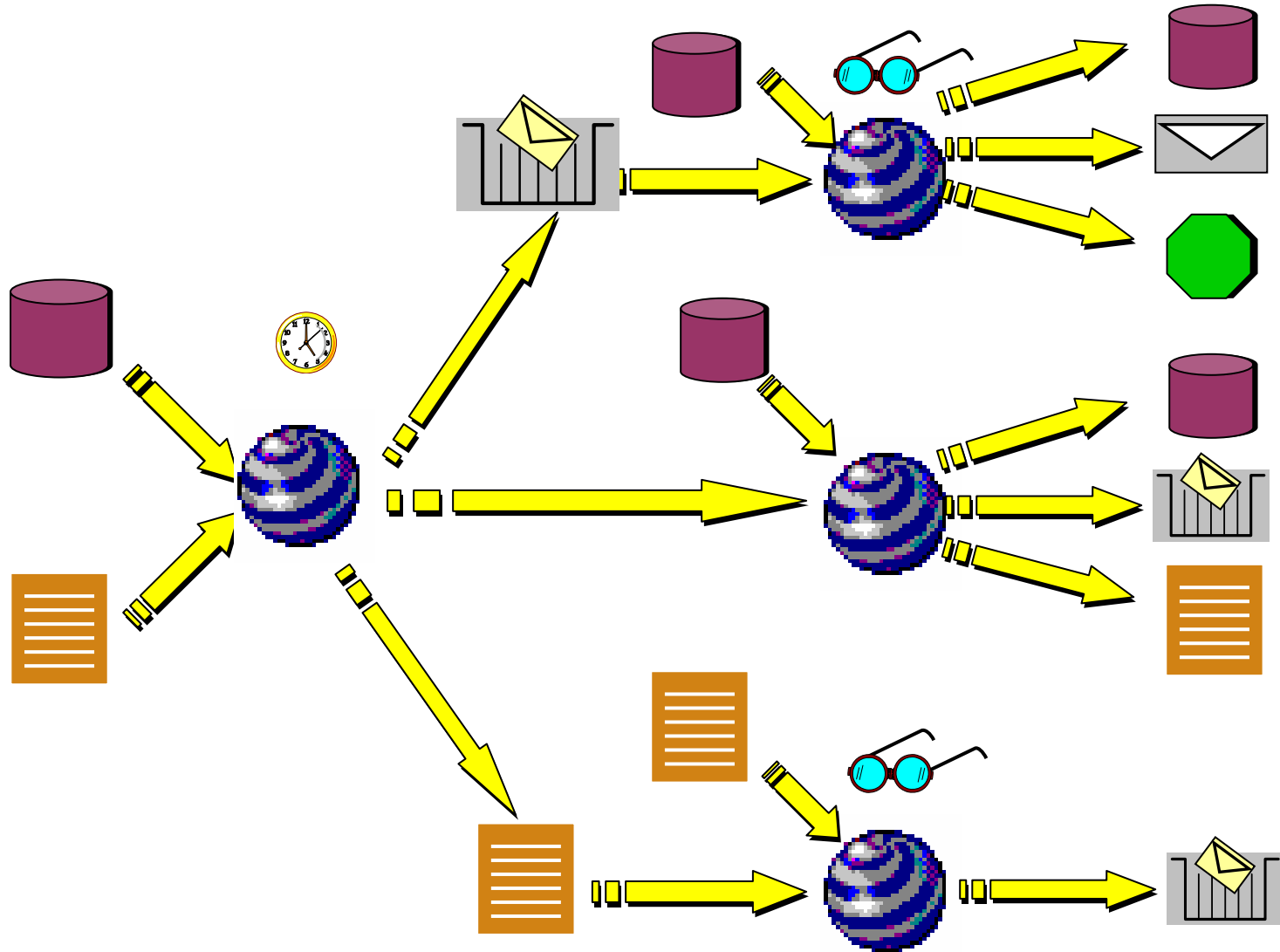
- Cependant... n à n est également possible:



- ▶ Une Interface peut faire :
 - ▶ Validation
 - ▶ Enrichissement, ...



Enchaînement des traitements (transformation et routage)



Comment développer un traitement dans WTX...

- 3 étapes simples:
 - ▶ Définir les structures de données
 - Génération de "Type Trees" (structure, syntaxe, sémantiques)
 - ▶ Définir les règles de transformation et de routage
 - Les règles sont définies entre source et destination par une approche "drag-and-drop" (glisser / déposer)
 - ▶ Déployer les flux et exécuter selon les méthodes différents
 - ✓ Batch
 - ✓ Événementiel
 - ✓ Couplage fort (par APIs)



Etape 1 – Définir les structures de données

- **Importateurs pré-définis permettant une génération automatique**
 - ▶ XML Schema, DTDs
 - ▶ COBOL Copybooks
 - ▶ Database (Tables, Vues, Procédures Stockées)
 - ▶ Applications Tiers
 - SAP: BAPI, Idoc, DXOB, BDC
 - PeopleSoft 7: Business Document, Message Agent
 - PeopleSoft 8: Application Messaging, Component Interface
 - Siebel: COM Business Object, EAI, EIM
 - ▶ WSDL (Web Services)
 - ▶ EJB API
 - ▶ Java Class
 - ▶ Java Messaging Service
 - ▶ Autres: Tuxedo, Fichier Texte, PL1, ...
- **Bibliothèques pré-définies**
 - EDIFACT
 - SWIFT
 - SEPA*
 - ACORD
 - HIPAA
 - HL7

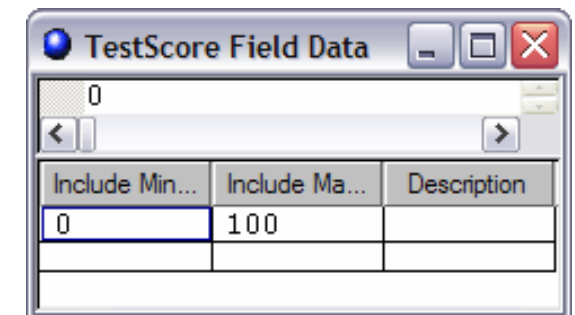
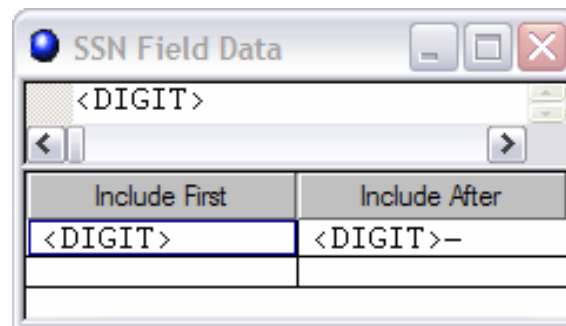
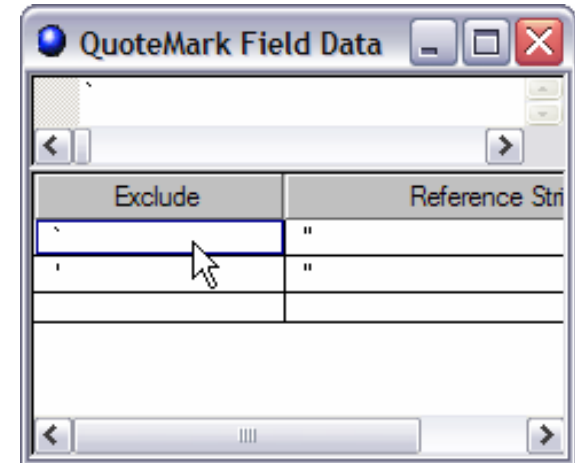
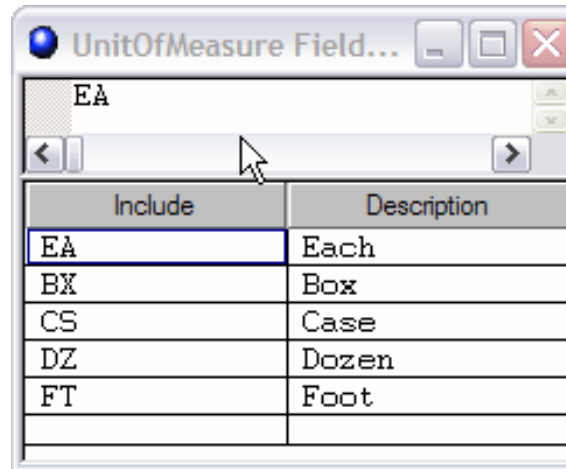
* En cours



Built-in Validation (Restrictions)

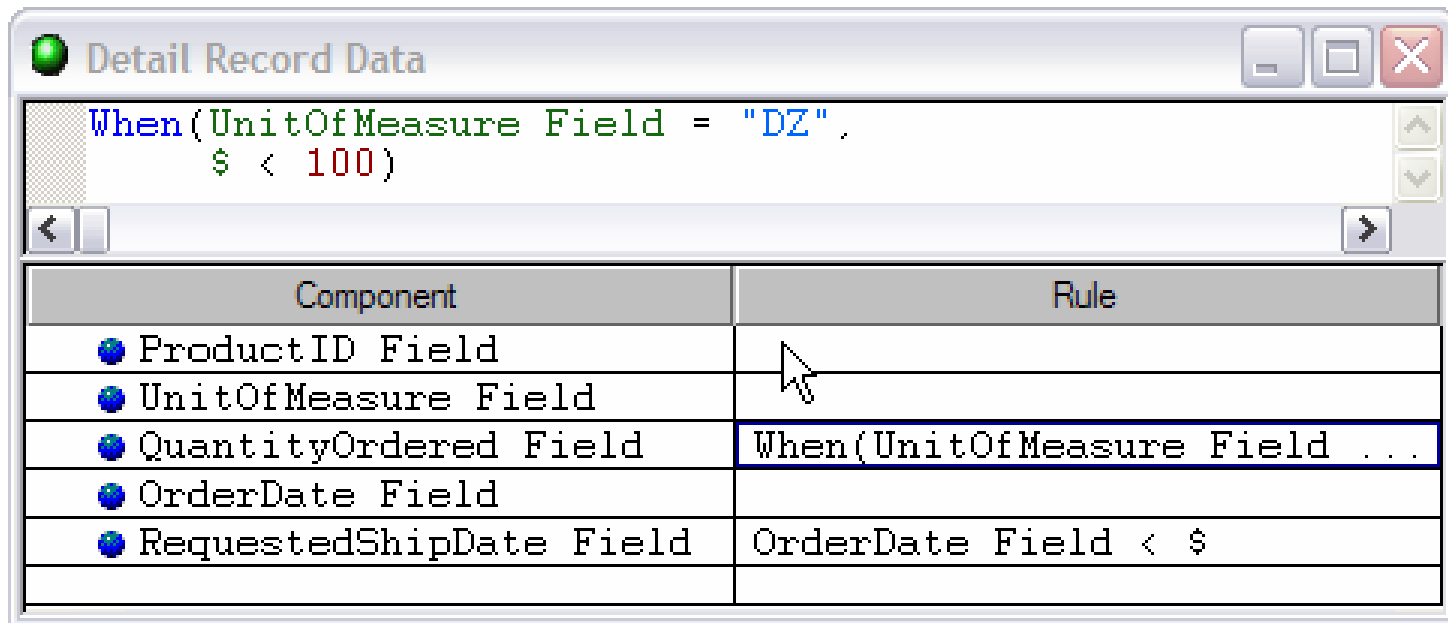
Items have extensive validation options

- Enumerations
- Exclusions (with substitution)
- Patterns
- Ranges



Additional Built-in Validation

- Component Rules allow cross-field validation
- Incorporates semantic validation directly into the parsing process



The screenshot shows a window titled "Detail Record Data" with a text area containing the following validation rule:

```
When(UnitOfMeasure Field = "DZ",  
      $ < 100)
```

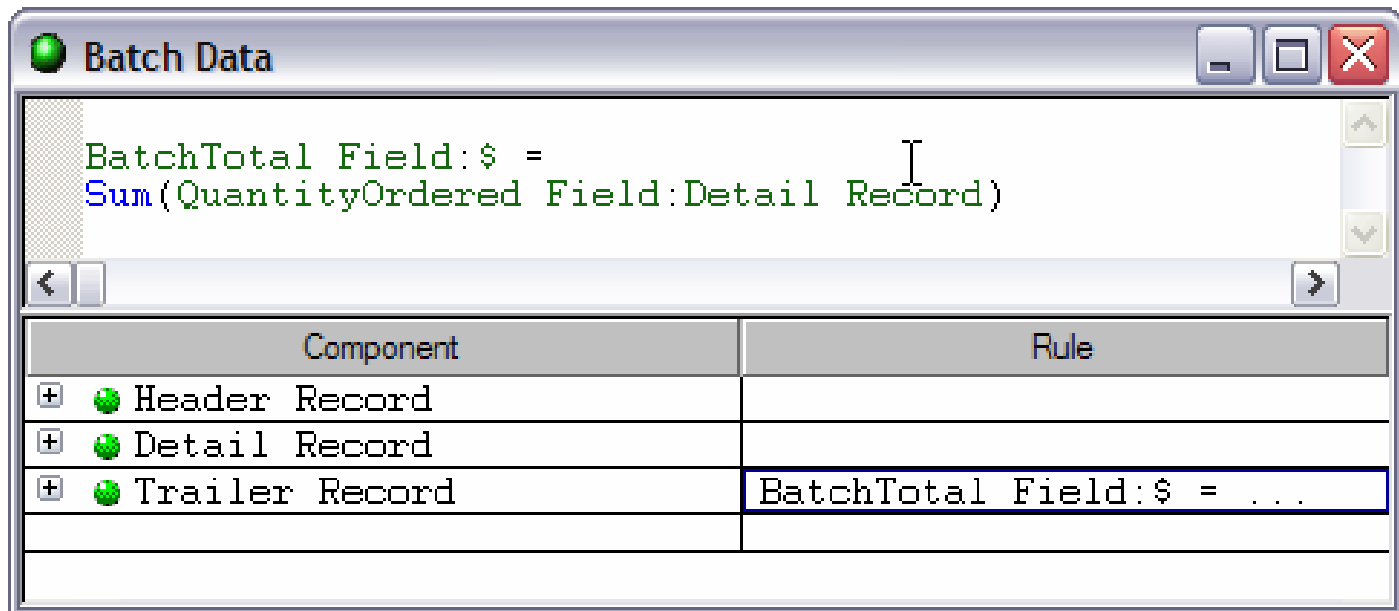
Below the text area is a table with two columns: "Component" and "Rule". The table lists several components and their associated rules.

Component	Rule
ProductID Field	
UnitOfMeasure Field	
QuantityOrdered Field	When(UnitOfMeasure Field ...
OrderDate Field	
RequestedShipDate Field	OrderDate Field < \$



Additional Built-in Validation

- Component Rules support aggregate functions
- Use component rules to specify conditions (business rules) that must be met for a particular component to be valid



The screenshot shows a window titled "Batch Data" with a text area containing the following code:

```
BatchTotal Field:$ =  
Sum(QuantityOrdered Field:Detail Record)
```

Below the text area is a table with two columns: "Component" and "Rule". The table lists three components: "Header Record", "Detail Record", and "Trailer Record". The "Trailer Record" row is selected, and its rule is "BatchTotal Field:\$ = ...".

Component	Rule
+ Header Record	
+ Detail Record	
+ Trailer Record	BatchTotal Field:\$ = ...

Etape 2 – Définir les règles de transformation

Fonctions
Pré Définies
(+ 170)

Map Designer - C:\demo\Generale\Legacy\Error Handling.mms F_MakeNewOrderRecord

WebSphere Transformation Extender Design Studio

File Edit View Map Card Rules Tools Window Help

=IF(WantDate Field:LegacyRecord < ADDDAYS(CURRENTDATE(), 5),
OrderDate Field:LegacyRecord,
WantDate Field:LegacyRecord)

Map Source Files

- Error Handling
 - InjectLegacy
 - Legacy2NewSys
 - F_MakeNewOrderRa

From

#2 XrefFile (XrefFile Data)

#1 LegacyRecord (LegacyRecord Data)

- LegacyRecord
 - OrderNumber Field
 - CustomerID Field
 - ShipToCode Field
 - ContactName Field (0:1)
 - OrderDate Field
 - WantDate Field (0:1)
 - CatalogNumber Field
 - QtyOrdered Field
 - UOM Field
 - UnitPrice Field
 - Message Field (0:2)

Insert Function

Category: All Categories

Functions:

- GETLOCALE
- GETPARTITIONNAME
- GETRESOURCEALIAS
- GETRESOURCENAME
- GETTXINSTALLDIRECTORY
- HANDLEIN
- HEXTEXTTOSTREAM
- IF**
- INDEX
- INDEXABS
- INT
- ISALPHA
- ISERROR
- ISLOWER

Evaluates a conditional expression, returning one value if true, another if false.

Syntax

IF (single-condition-expression , single-general-expression [, single-general-expression])

Meaning

IF (test_this , result_if_true [, result_if_false])

Returns (a single-item or single-group)

Design Insert Close

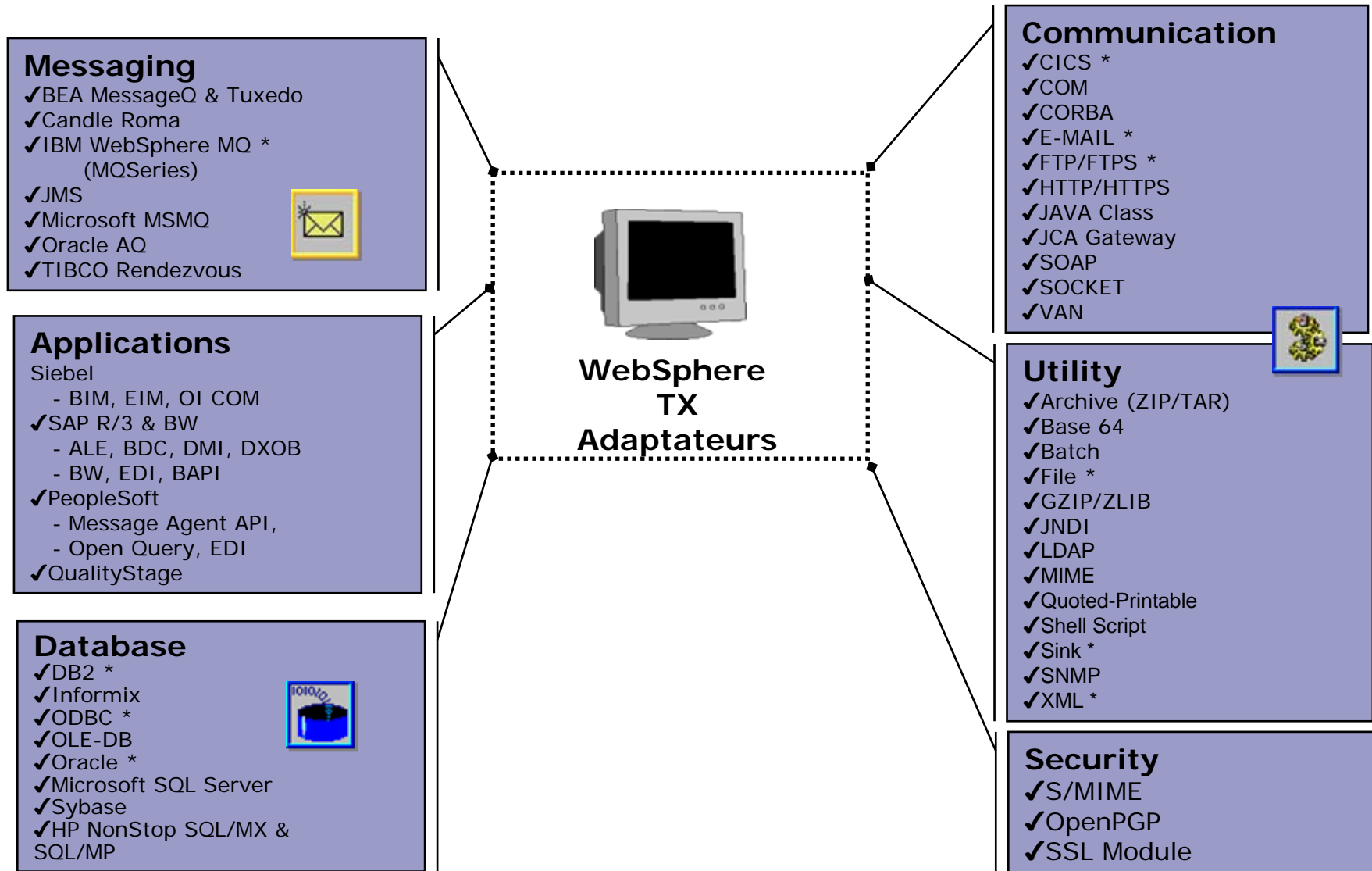
CustomerID Field	=LOOKUP(NewCustomerID Field::XrefFile, O...
ShipToCode Field	=TEXTTONUMBER(ShipToCode Field:Legac...
OrderDate Field	=OrderDate Field:LegacyRecord
WantDate Field	=IF(WantDate Field:LegacyRecord < ADDDA...
CatalogNumber Field	=CatalogNumber Field:LegacyRecord
QtyOrdered Field	=QtyOrdered Field:LegacyRecord
UOM Field	=UOM Field:LegacyRecord
UnitPrice Field	=UnitPrice Field:LegacyRecord
ExtendedPrice Field	=ROUND(QtyOrdered Field:LegacyRecord * U...
ReceivedDate Field	=CURRENTDATE()
RoutingInfo Field	=NONE

List Composition

Ready

- Glisser / Déposer
- Développement graphique!!

Acquisition et émission des flux par les adaptateurs WebSphere TX



Appel des fonctions / modules externe

The screenshot displays the IBM Map Designer interface. The main window shows a transformation rule with the following structure:

- #2 Create (TextGroup Root)
- #1 output (TextGroup Root)
- #3 UnallocDD (TextGroup Root)
 - Output: Rule
 - UnallocDD: =EXIT("BNPDYN","UNALLOC","CARD1B")
 - Testitem: =EXIT("BNPDYN","UNALLOC","CARD1B")

An "Insert Function" dialog box is open, showing the "EXIT" function selected in the "Functions" list. The dialog provides the following information:

- Category:** All Categories
- Functions:** DATETOTEXT, DBLOOKUP, DBQUERY, DDEQUERY, DEFAULT, ECHOIN, EITHER, **EXIT**, EXP, EXTRACT, FACTORIAL, FAIL, FILLLEFT, FILLRIGHT
- Description:** Allows you to interface with a function in an external library or application. Depending on the execution platform, there are two different methods for the EXIT function. Refer to the Using a Command Execution Engine documentation for information on the methods available for your platform.
- Syntax:** EXIT (single-text-expression , single-text-expression , single-text-expression)
- Meaning 1 - Library Method:** EXIT (library_name , function_name , input_to_the_function)
- Meaning 2 - Program Method:** (No specific syntax provided)

At the bottom of the dialog, there are buttons for "Design", "Insert", and "Close".

Gestion des erreurs (cycles et recyclage)

Map Designer - C:\demo\Generale\Legacy>Error Handling.mms Legacy2NewSys

WebSphere Transformation Extender Design Studio

File Edit View Map Card Rules Tools Window Help

=REJECT(LegacyRecord:LegacyFile)

Map Source Files

- Error Handling
- InjectLegacy
- Legacy2NewSys

List Composition

Item	Value
Intent	General
Item Sub...	Text
Implied D...	
Type Syn...	
Documen...	Default
Description	

From

#2 XrefFile (XrefFile Data)

#1 LegacyFile (LegacyFile Data)

- LegacyFile
 - LegacyRecord (s)
 - OrderNumber Field
 - CustomerID Field
 - ShipToCode Field
 - ContactName Field (0:1)
 - OrderDate Field
 - WantDate Field (0:1)
 - CatalogNumber Field
 - QtyOrdered Field
 - UOM Field
 - UnitPrice Field
 - Message Field (0:2)

To

#1 NewSysFile (NewOrderFile data)

#2 ErrorFile (ErrorFile Data)

Output	Rule
ErrorFile	
FileName Field	=GETFILENAME(LegacyFile)
ProcessDate Field	=CURRENTDATE()
ProcessTime Field	=CURRENTTIME()
Message Field	=IF(PRESENT(REJECT(LegacyRecord:LegacyFile)), "Les enr...")
ErrorText Field (s)	=REJECT(LegacyRecord:LegacyFile)

Insert Function

Category: All Categories

Functions:

- PACKAGE
- PARTITION
- POWER
- PRESENT
- PROPCASE
- PUT
- QUOTEDTOTEXT
- RAND
- REFORMAT
- REJECT**
- REVERSEBYTE
- RIGHT
- ROUND
- RUN

Returns the content of an object in error as a text item.

Syntax

REJECT (series-object-expression)

Meaning

REJECT (series_to_look_for_bad_objects)

Returns (a series-text-item)

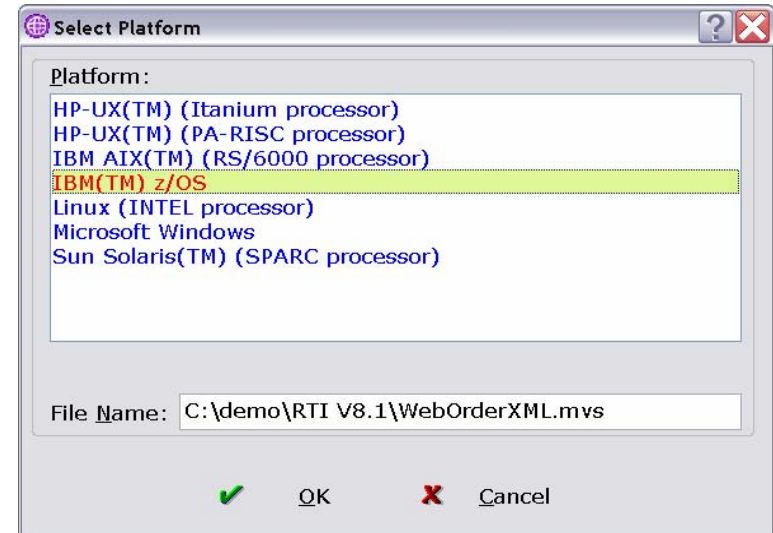
Design Insert Close

Ready



Etape 3 – Déployer et exécuter le traitement

- Option “Build” permet de créer une map compilée pour
 - ▶ la plate-forme d’exécution
 - ▶ avec une prise en charge automatique des différences de “byte-order” et de code page

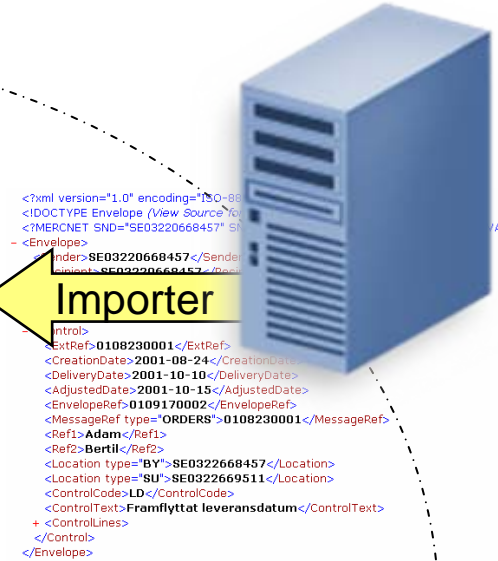
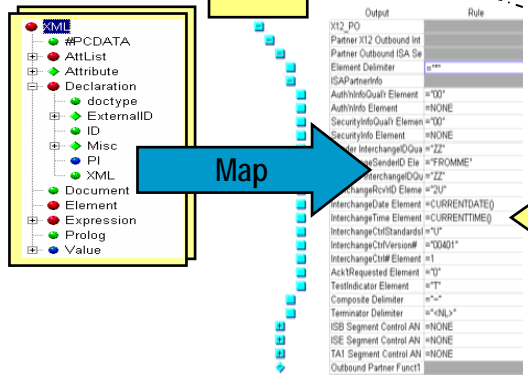
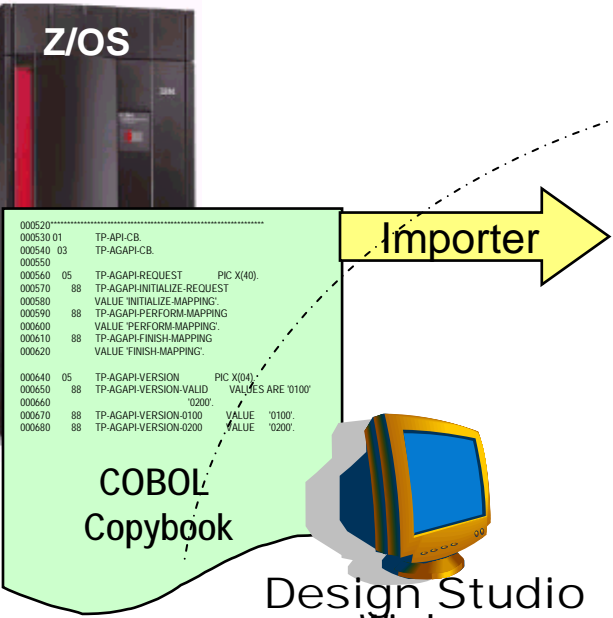
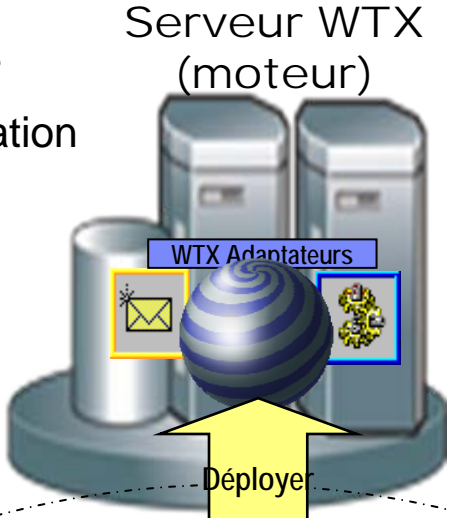


Exemple – développement d'un traitement WTX

1. Définir les structures de données
2. Définir les règles de transformation
3. Déployer et exécuter

Multiplés methodes d'invocation

- API
- Batch / JCL
- Événementiel
 - ▶ WebSphere Message Broker
 - ▶ WebSphere ESB
 - ▶ WebSphere Process Server
 - ▶ WebSphere Partner Gateway
 - ▶ WebSphere Application Server
 - ▶ WebSphere DataStage, etc.



Design Studio Windows

Conclusions

- Le meilleur compromis performance/ richesse fonctionnelle du marché
- Une solution qui décharge les systèmes métiers de la manipulation des flux
- Richesse des fonctionnalités
 - ▶ Transformation des flux
 - ▶ Mapping des données
 - ▶ Contrôles syntaxiques et sémantiques
 - ▶ Support des standards (SEPA, SWIFT, EDIFACT,)
 - ▶
- Des interfaces de développement à la fois performantes et conviviales
 - ▶ Définition graphique des règles d'intégration et de la logique métier
 - ▶ Élimination totale de toute programmation
- Des performances et une scalabilité démontrée
- Une solution ouverte, aisément intégrable dans l'architecture globale



धन्यवाद
Hindi

多謝
Traditional Chinese

Grazie
Italian

ขอบคุณ
Thai



Спасибо
Russian

Obrigado
Brazilian Portuguese

Merci
French

Gracias
Spanish

شكراً
Arabic

多谢
Simplified Chinese



Danke
German

நன்றி
Tamil

ありがとうございました
Japanese

감사합니다

WebSphere software

