



# Schnittstellendesign mit IBM Tivoli Directory Integrator am Beispiel SAP

Andreas Artner  
IBM Software Consultant

**FRITZ & MACZIOL**

Software, Systeme und Dienstleistungen

<http://www.fum.de>



# Agenda

- TDI Übersicht
  - Einsatzgebiete
  - Architektur und Komponenten
  - Grundlagen
  - Verbindung zu den Quell-, Zielsystemen
- TDI Verbindung zu Domino & SAP
- TDI Konfigurationseditor
- TDI SAP Schnittstellen Beispiele
- TDI Tipps und Tricks



# TDI Übersicht

- **TDI** ermöglicht die synchronisation von **Daten und Directories**
- Lassen Sie sich nicht von dem Namen täuschen – **TDI** kann mit nahezu jeder **Datenquelle** arbeiten
- TDI bietet eine **grafische Oberfläche** zur Konfiguration der Schnittstellen
- IBM setzt TDI ein z.b. zusammen mit:
  - IBM Connections
  - IBM LotusLive
- Der Einsatz des TDI ist für Lotus Domino Bestandskunden **lizenzkostenfrei !**

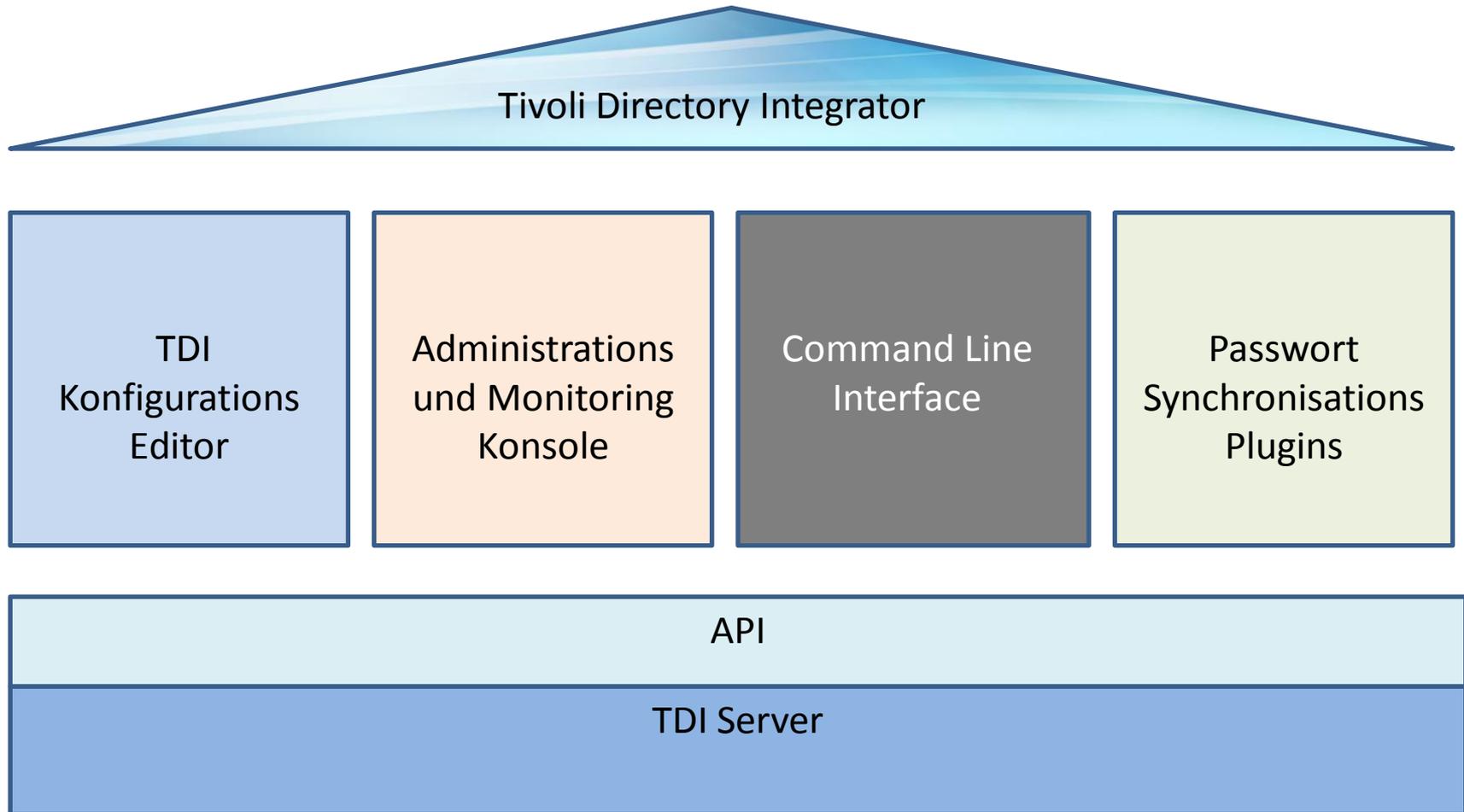


# TDI Anwendungsfälle

- Domino Directory – Synchronisation mit LDAP Directory (Active Directory)
- Domino Datenbank – Import von einer Oracle Tabelle
- SAP Materialanforderung aus Domino Applikation erzeugen
- Verbindung zu SQL Datenbank um Domino web page oder Datenbank zu aktualisieren (gescheduled oder eventgesteuert)
- Zwei Domino Datenbanken miteinander abgleichen
- XML-Export aus Domino Datenbank
- .....

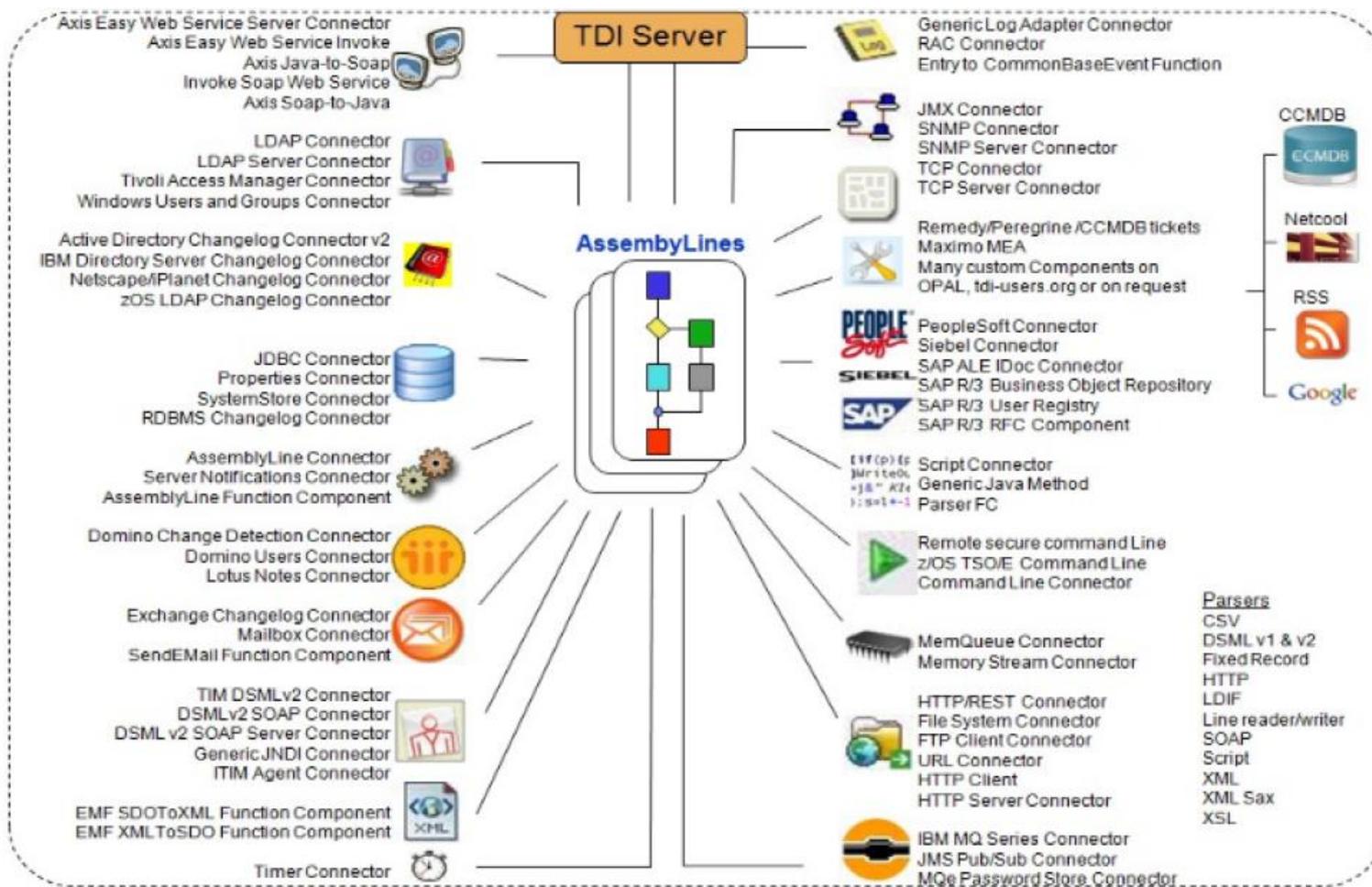


# TDI Komponenten





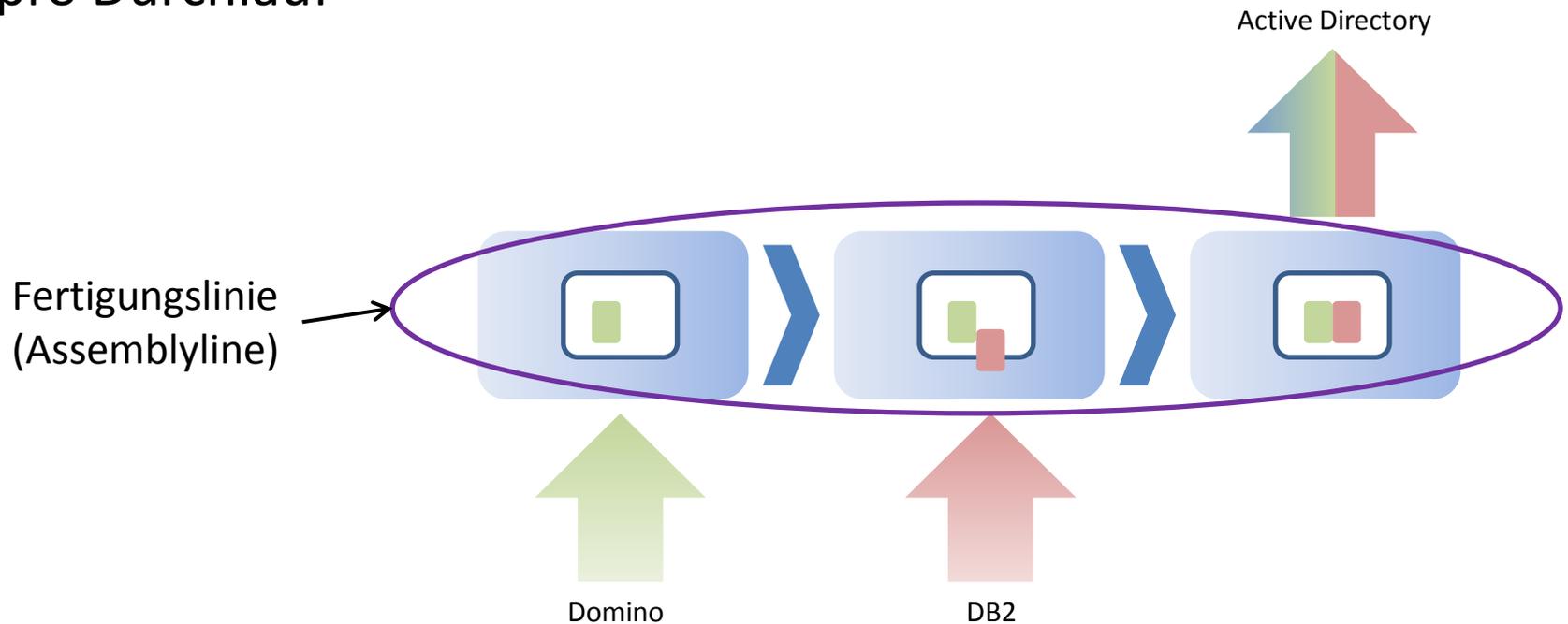
# TDI Übersicht





# TDI Grundlagen

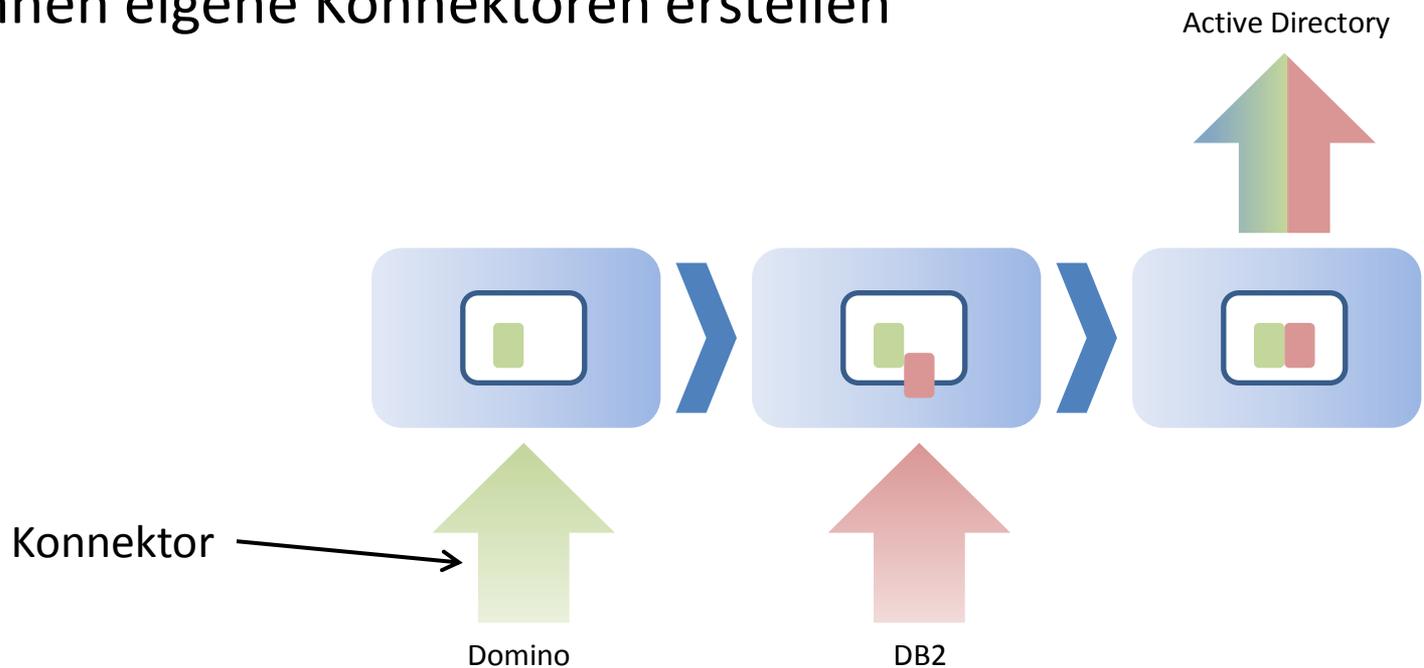
- Die **Fertigungslinie** (AssemblyLine) beschreibt den **logischen Fluss** der Daten
- Eine Fertigungslinie verarbeitet immer nur **einen Datensatz** pro Durchlauf





# TDI Grundlagen

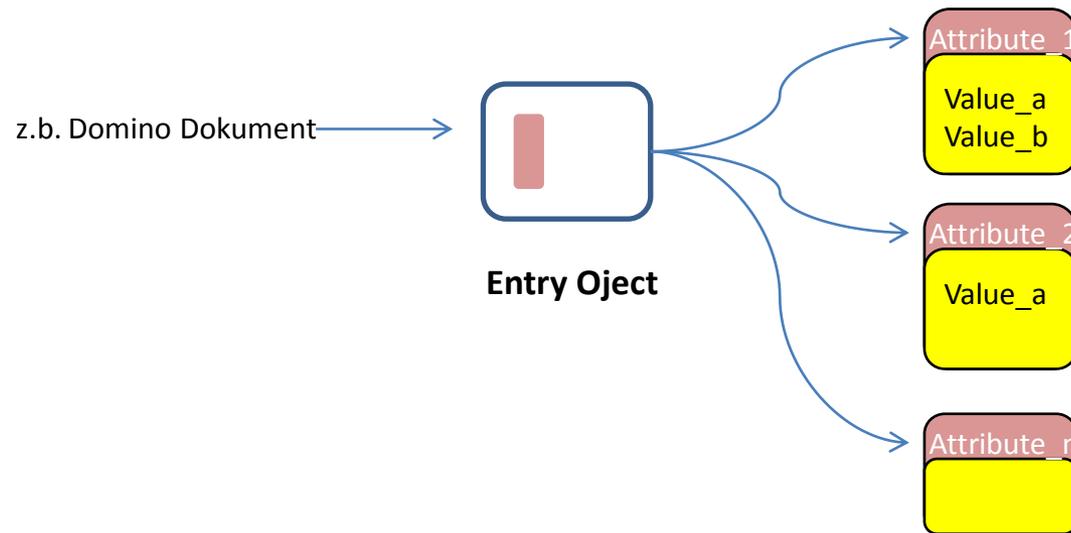
- **Konnektoren** stellen die Verbindung zu den einzelnen Datenquellen her
- Ein Konnektor läuft immer in einem bestimmten Modus (Iterate, Update, Delete, etc.)
- Sie können eigene Konnektoren erstellen





# TDI Grundlagen

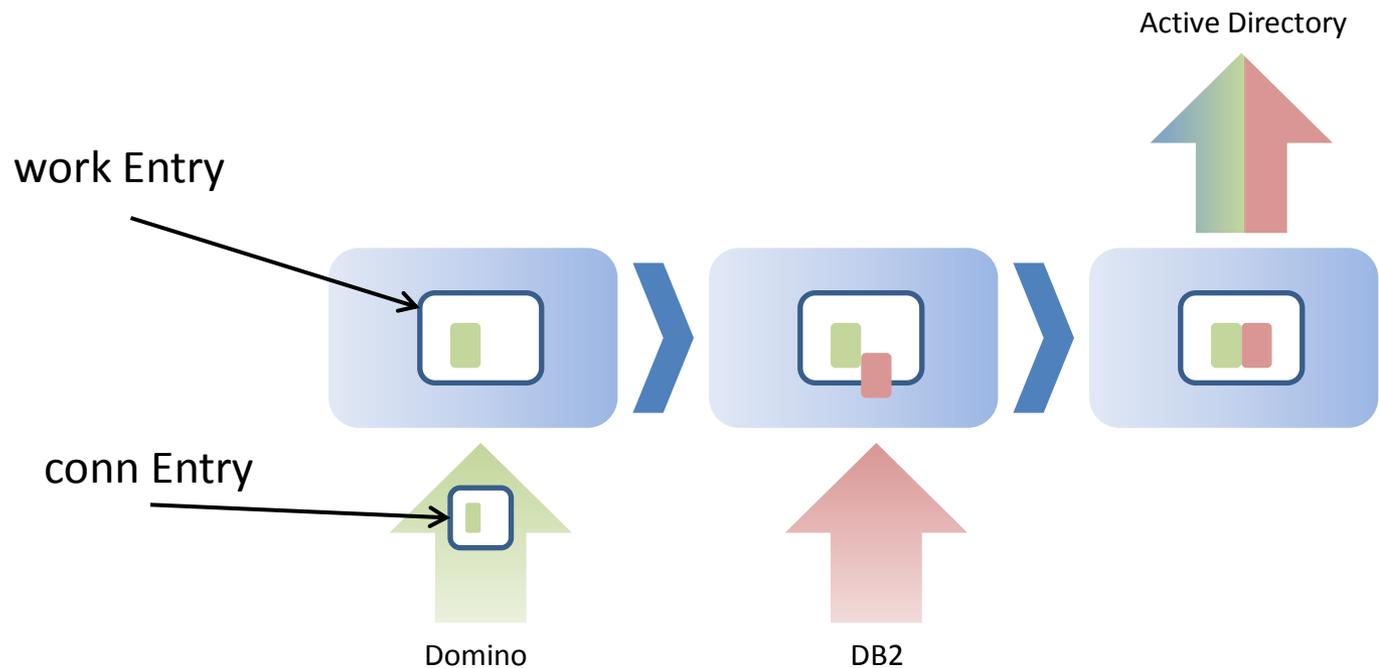
- Wie Daten abgelegt werden ist von System zu System verschieden (z.b. Domino vs. Relationale Datenbanken)
- Der gemeinsame Nenner - das **TDI Datenmodell**





# TDI Grundlagen

- **work Entry** – Der Datensatz der durch die Fertigungslinie läuft und von ihr modifiziert wird
- **conn Entry** – lokaler Cache des jeweiligen Konnektors



# TDI Verbindung zu den Quell- und Ziel Systemen



- Es stehen über 50 Konnektoren zur Verfügung
- Sie stellen die Verbindung zwischen der Datenquelle und der Fertigungslinie her
  - Informationen wie Hostnamen, Benutzer, Passwörter, View oder Tabellennamen etc. werden hier konfiguriert
- Einige Konnektoren benötigen zusätzliche Libraries oder Java-Archive des Herstellers. Z.B.:
  - Relationale Datenbanken -> JDBC-Treiber
  - SAP -> JCo



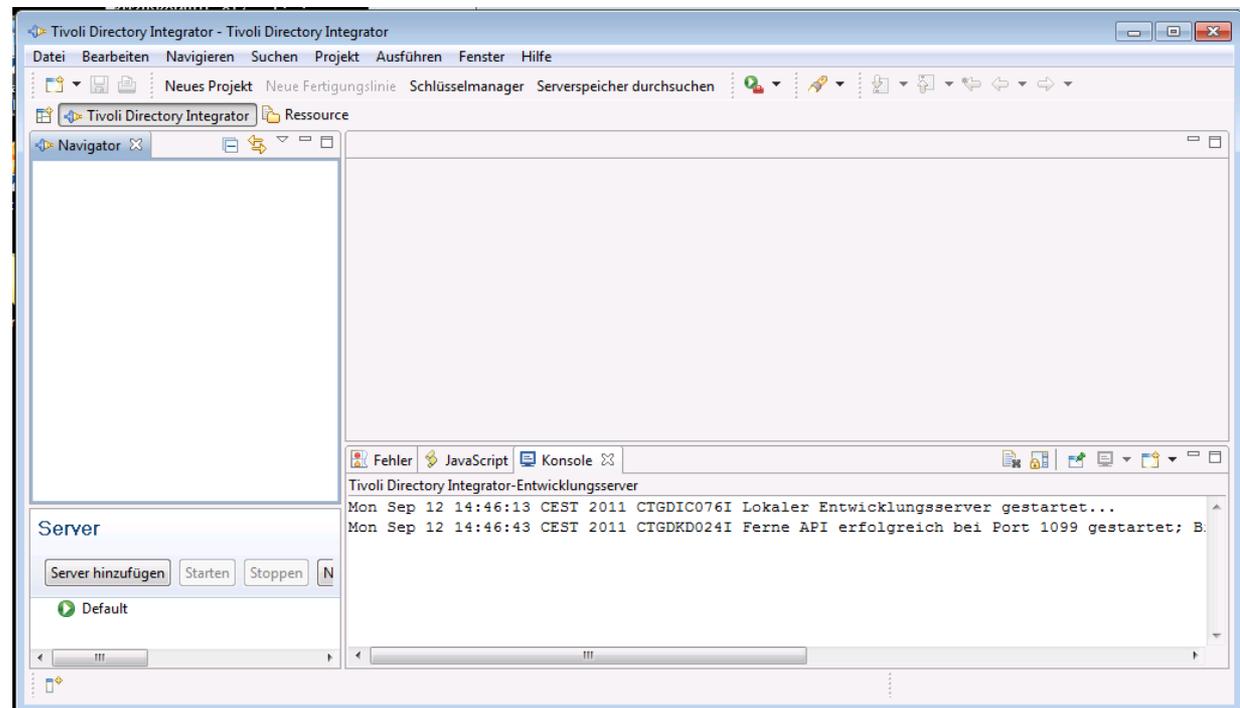
# Agenda

- TDI Übersicht
- TDI Konfigurationseditor
  - Bestandteile des Konfigurationseditors
  - Fertigungslinien und deren Komponenten
  - Tipps
- TDI Schnittstellen zwischen Domino & SAP
- TDI SAP Schnittstellen Beispiele
- TDI Tipps und Tricks



# TDI – Konfigurationseditor

- Eclipse basierend
- Wird in Dokumentationen oft als auch CE (von **C**onfiguration **E**ditor) bezeichnet.

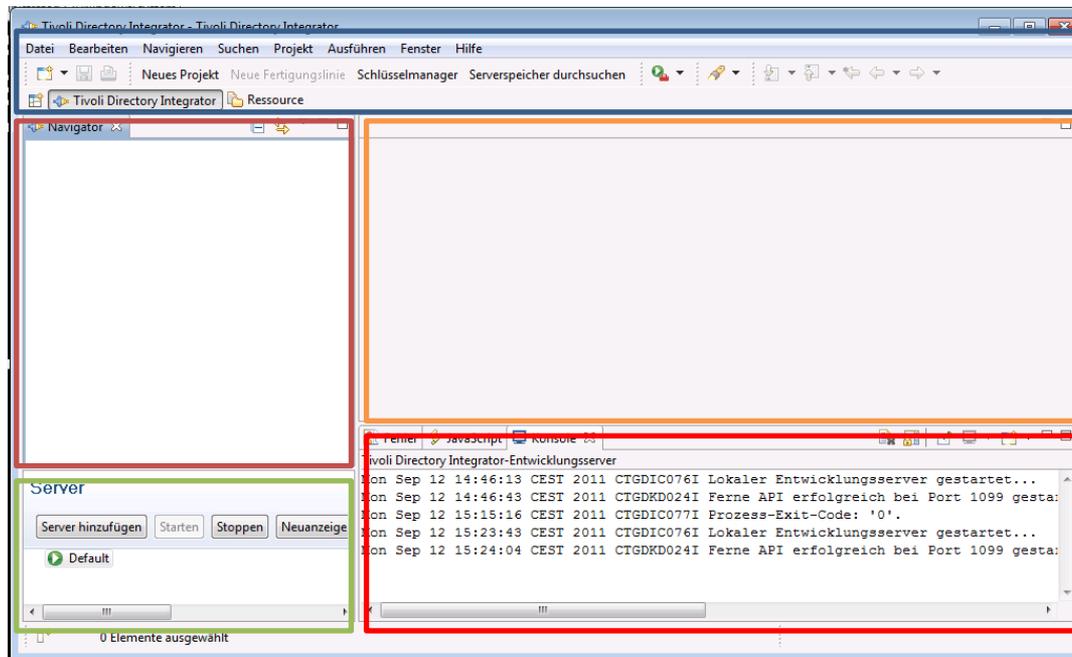




# TDI CE Komponenten

- Der Konfigurationseditor gliedert sich in 5 Bereiche:

## Menü-Leiste



Navigationsbereich

Haupt-Fenster

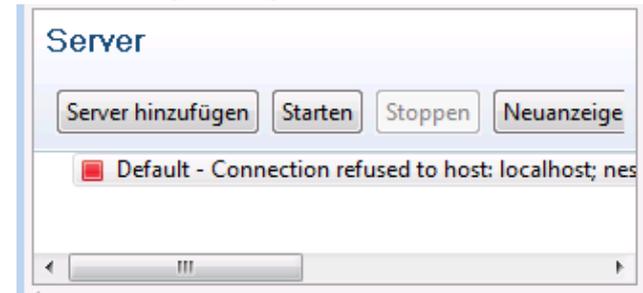
Entwicklungsserver

Konsole und  
Logs



# TDI CE - Entwicklungsserver

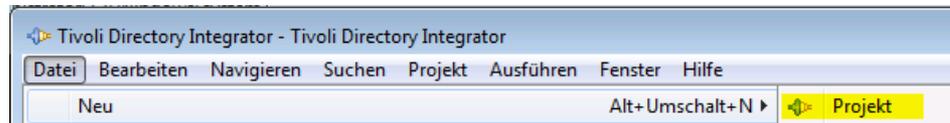
- Wichtig: Der Konfigurationseditor startet im Hintergrund einen eigenen TDI Entwicklungsserver.
- Startet dieser nicht erfolgreich kann die Fertigungslinie im Konfigurationseditor nicht laufen.
- Häufige Ursachen sind hier:
  - ein TDI Windows Service auf der selben Maschine
  - ein nicht sauber beendeter Konfigurationseditor
- Deshalb, bitte immer zuerst den TDI Windows-Service beenden bevor der Konfigurationseditor gestartet wird
- Task-Manager auf vorhandene Prozesse mit java.exe prüfen und beenden falls der Konfigurationseditor mal abstürzt.





# TDI CE – Ein Projekt erstellen

- Ein Projekt:
  - Beschreibt eine komplette Datenschnittstelle
  - Kann aus einer oder mehreren Fertigungslinien bestehen
  - Bildet am Ende **eine** „Konfigurationsdatei“
- Um ein neues Projekt zu erstellen:
  - Wählen Sie im Menü Datei > Neu > Projekt



- Oder klicken Sie auf den Button „Neues Projekt“



- Vergeben sie einen eindeutigen sprechenden Namen



# TDI CE – Ein Projekt erstellen

- Neben den Fertigungslinien können in einem Projekt auch Ressourcen definiert werden.
- Ressourcen sind z.B.:
  - Eigenschaften wie Usernamen und Passwörter
  - Konnektoren die mehrmals verwendet werden
  - Script Komponenten für z.B. Logging oder Errorhandling
  - .....
- Änderungen an Ressourcen greifen sofort an allen Stellen im Projekt an denen sie verwendet wurden !



# TDI CE – Neue Fertigungslinie

- Fertigungslinien sind die zentralen Komponenten im TDI
- Um eine Neue Fertigungslinie anzulegen verwenden Sie:
  - Rechte Maustaste > Neu Fertigungslinie



- oder den Button „Neue Fertigungslinie“





# Komponenten der Fertigungslinie

- Fertigungslinien bestehen aus:
  - **Konnektoren**
    - Konnektoren stehen für eine Vielzahl von Systemen zur Verfügung
  - **Funktionen**
    - Wie z.B. die „Funktionskomponente der Fertigungslinie“ um eine andere Fertigungslinie zu starten
  - **Steuerungs- und Datenfluss Komponenten**
    - Schleifen, IF – ELSE Abfragen usw.
  - **Skripts**
    - Eigen erstellte oder vordefinierte Skript-Blöcke
  - **Attributzuordnungen**
    - Mapping von Attributen
- Der Einsatz aller Komponenten ist optional !



# Komponenten der Fertigungslinie

- Eine Fertigungslinie gliedert sich in zwei Bereiche:

- Zuführung:

- Hier sind nur Konnektoren erlaubt, die Daten an die Fertigungslinie übergeben
- Ist keine Zuführung definiert wird die Fertigungslinie einmal durchlaufen
- Sind mehrere Zuführungen definiert werden die Daten aus den Konnektoren der Reihe nach abgearbeitet. Zuerst Iterator1 dann Iterator2, usw.

- Datenfluss

- Hier sind alle Komponenten erlaubt
- Im Datenfluss wird gesteuert wie die Daten weiter verarbeitet werden
- So können z.B. Weitere Informationen aus einem anderen System abgerufen, Schleifen ausgelöst oder Bedingungen überprüft werden
- Die Reihenfolge der Kpomonenten im Datenfluss ist hierbei entscheidend !





# TDI CE - Konnektoren

- Sind Batch oder Event getriggert
- Jeder Konnektor unterstützt einen oder mehrere der folgenden Modi:
  - Iterator:
    - geht durch alle Einträge einer Datenquelle
    - liest Attribute aus und übergibt diese an eine Fertigungslinie
  - Lookup:
    - Liest zusätzliche Informationen von weiteren Datenquellen
  - AddOnly:
    - Fügt Datensätze an ohne zu überprüfen ob der Eintrag schon existiert
  - Update:
    - Fügt Datensätze hinzu oder führt updates auf bereits existierenden Einträgen aus
  - Delete:
    - Löscht Datensätze
  - Delta:
    - Stellt Änderungen in der Datenquelle fest und liefert nur die geänderten Daten zurück



# TDI CE - Konnektoren

- Verknüpfungsbedingungen:
  - Konnektoren im **Update**, **Lookup** oder **Delete** Modus benötigen die Information über welchen **Schlüssel** eine Verbindung der Einträge in den Datensystemen erfolgen soll.
  - Beliebige Und / Oder Bedingungen können im Dialog definiert werden.
  - Spezielle Bedeutung von \$ und @ vor dem Namen des Work-Attribute:
    - » \$ - nur der erste Wert wird verglichen
    - » @ - alle Werte
  - Oder Bedingung in eigenem Script
    - » `ret.filter= .....`

Hinzufügen  Beliebige Übereinstimmung

CONNECTID	gleich	\$SAP_RESPONSE_CONNECTID
FLIGHTDATE	gleich	SAP_RESPONSE_FLIGHTDATE

kleiner als  
größer als  
größer oder gleich

Bedingungen mit angepasstem Script erstellen

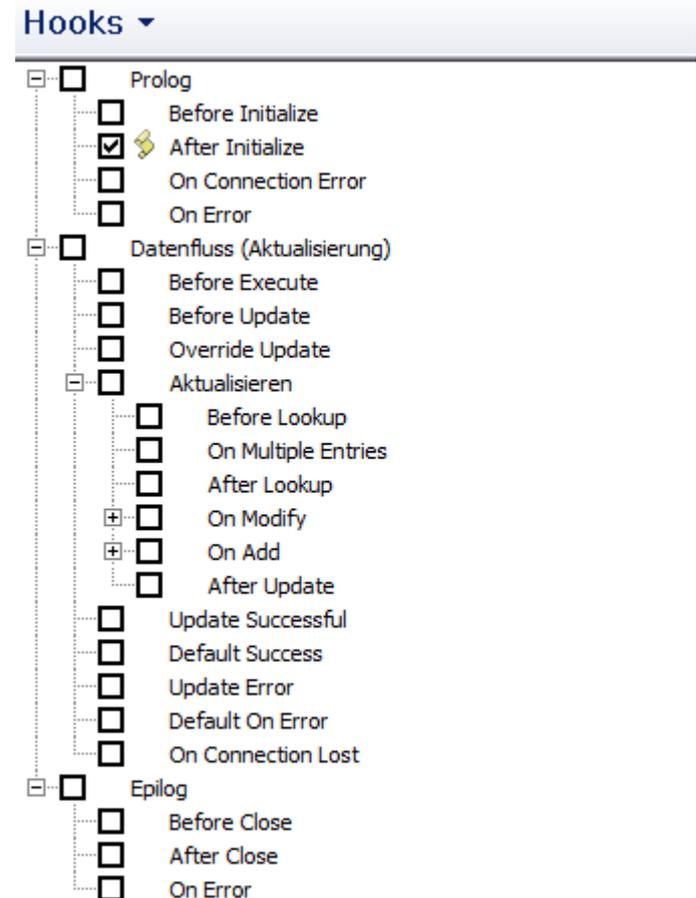
Script

```
ret.filter="CONNECTID=\" + work.getString("SAP_RESPONSE_CONNECTID") +  
\" & FLIGHTDATE=\" + work.getString("SAP_RESPONSE_FLIGHTDATE") +\"";
```



# TDI CE - Konnektoren

- Hooks:
  - Je nach Konnektor Modus sind verschiedene Hooks verfügbar
  - Erlauben das **gezielte Eingreifen** in den Ablauf einer Fertigungslinie um Fehler abzufangen oder den Datenfluss zu steuern.  
z.b. Das Skript „system.exitFlow();“ im „Override Add“ Hook eines Konnektors unterbindet das neue Datensätze angelegt werden.
- [Hook Flow diagrams](#)





# TDI CE - Funktionskomponenten

- Funktionskomponenten:
  - Stellen gekapselte Funktionen zur Verfügung
  - Z.b. Senden einer email, Aufruf Webservice, Aufruf SAP-BAPI
  - Ähnlich wie Konnektoren
  - Haben keine unterschiedlichen Modi
  - Stellen ebenfalls Hooks zur Verfügung

Name
CBE-Funktionskomponente
Deltafunktionskomponente
Funktionskomponente Axis2WSCClient
Funktionskomponente AxisEasyInvokeSoapWS
Funktionskomponente AxisJavaToSoap
Funktionskomponente AxisSoapToJava
Funktionskomponente CastorJavaToXML
Funktionskomponente CastorXMLToJava
Funktionskomponente der Fertigungslinie
Funktionskomponente EMFSDToXML
Funktionskomponente EMFXMLToSDO
Funktionskomponente für den Generator für komplexe Typen
Funktionskomponente für die Datenbereinigung
Funktionskomponente für die ferne Befehlszeile
Funktionskomponente für die Initialisierung der IT-Registry
Funktionskomponente für IdML-Ausgabe in mehreren Dateien
Funktionskomponente für Java-Klassen
Funktionskomponente für SOAP-Generierung
Funktionskomponente für Speicherwarteschlangen
Funktionskomponente InvokeSoapWS
Funktionskomponente zum Schließen von IdML
Funktionskomponente zum Öffnen von IdML
Parser-Funktionskomponente
SAP ABAP AS RFC-Funktionskomponente
Script-Funktionskomponente
SendEMail-Funktionskomponente
z/OS-TSO/E-Befehlszeile



# TDI CE - Skript

- TDI ist pure Java !
- Alle TDI-Funktionen liefern Java-Objekte zurück
- Darüber hinaus können Sie jederzeit Java Klassen aufrufen
  - `var myfilereader = new java.io.FileReader("Datainame");`
- Java + Script != Java Script
  - Z.b. String vergleiche
    - `if(str_a == str_b) <-> if (str_a.equals(str_b))`



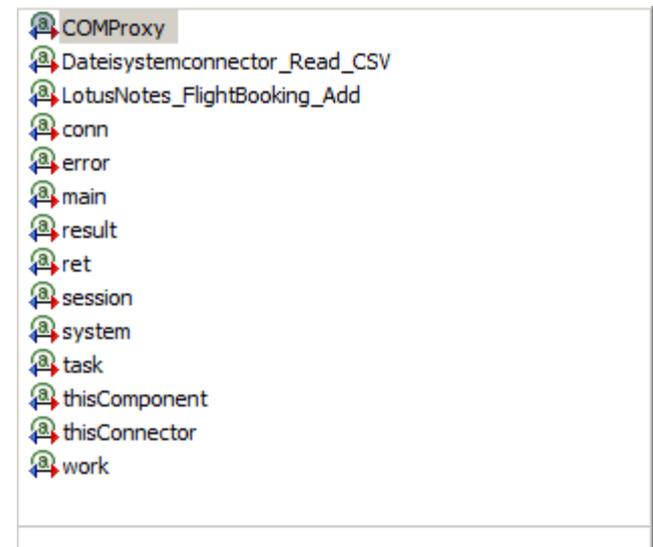
# TDI CE - Skript

- Skripte können definiert werden in:
  - Skript Komponenten
  - Fertigungslinien Hooks
  - Server Hooks
  - Attributzuordnungen
  - Verknüpfungsbedingungen
  - Konnektor Eigenschaften
  - .....
- Viele nützliche Funktionen sind in TDI implementiert:
  - `system.arrayToString();`
  - `system.copyFile();`
  - `system.parseDate();`
  - `system.getFTP();`
  - .....



# TDI CE - Skript

- Der Zugriff auf die TDI-Umgebung, die Fertigungslinie, etc. erfolgt über definierte Objekte.
- Entry Objekte:
  - work
    - Datencontainer der durch die Fertigungslinie läuft
  - conn
    - Lokaler Cache des jeweiligen Konnektors
    - Nur Verfügbar während des AttributMappings
  - current
    - aktueller Eintrag im Konnektor
    - Nur verfügbar im Modus Add und Delete
  - error
    - Details zum letzten Fehler





# TDI CE - Skript

- Sonstige Objekte:
  - main
    - Top Level Thread
  - system
    - Viele nützliche Skript Funktionen
  - task
    - Das Fertigungslinien Objekt
- Spezielle Objekte:
  - thisComponent
  - thisConnector
  - <Name eines Konnektors>



# TDI CE - Skript

- Jede Fertigungslinie hat ihre eigene Script Engine
- Jede variable ist per Default global für die Script Engine
- Lokale variablen können nur innerhalb einer Funktion definiert werden:

```
function myFunction(){  
    var myVar = "This is a local variable";  
}
```



# TDI CE - Skript

- Fertigungslinie im Debugger

**Fertigungslinienentwurf**

**Fertigungslinienkomponenten, Attribute und Werte**

**Überwachungsliste**

**Breakpoints setzen**

**Live Konsole für Skript**

`task.logmsg("INFO","DEBUG: " + work.getString("AIRLINEID")); >> null`

Quellena...	Quellenwert
AIRLINE	American...
AIRLINEID	AA
AIRPORTFR	JFK
AIRPORTTO	SFO
ARRDATE	20110914
ARRTIME	140100
CITYFROM	NEW YORK
CITYTO	SAN FRAN...

Zielattr...	Zielwert
AIRPORTTO	SFO
ARRDATE	20110914
ARRTIME	140100
CITYFROM	NEW YORK
CITYTO	SAN FRAN...
CONNECTID	0017
CURR	USD
PRICE	422.9400

Name/Ausdruck	Wert
work	{"FLIGHTDATE": "20110914", "AIRPO...
AIRLINE	"AIRLINE": "American Airlines"
AIRLINEID	"AIRLINEID": "AA"
0	AA
AIRPORTFR	"AIRPORTFR": "JFK"
AIRPORTTO	"AIRPORTTO": "SFO"
ARRDATE	"ARRDATE": "20110914"
ARRTIME	"ARRTIME": "140100"
CITYFROM	"CITYFROM": "NEW YORK"
CITYTO	"CITYTO": "SAN FRANCISCO"
CONNECTID	"CONNECTID": "0017"
CURR	"CURR": "USD"
CURR_ISO	"CURR_ISO": "USD"
DEPTIME	"DEPTIME": "110000"
FLIGHTDATE	"FLIGHTDATE": "20110914"
PRICE	"PRICE": "422.9400"

```
task.logmsg("INFO","DEBUG: " + work.getString("AIRLINEID")); >> null
16:00:23,363 INFO - DEBUG: AA
```



# Agenda

- TDI Übersicht
- TDI Konfigurationseditor
- TDI Schnittstellen zwischen Domino & SAP
  - Verbindung zu Domino
  - Verbindung zu SAP
  - SAP ABAP RFC Funktions Komponente
  - SAP Requests erzeugen
  - SAP Response verarbeiten
- TDI SAP Schnittstellen Beispiele
- TDI Tipps und Tricks



# TDI Die Verbindung zu Domino

- TDI bietet verschiedene Konnektoren zu Lotus Domino:
  - Domino Change Detection
  - Domino User
  - Lotus Notes
  - Domino AdminP
- Der Lotus Notes Konnektor bietet hierbei die größte Flexibilität
- Der Domino User Konnektor und auch der AdminP Konnektor bieten leider nicht die Möglichkeit mit Certificate Authorities zu arbeiten



# TDI Die Verbindung zu Domino

- Je nach Konnektor werden unterschiedliche Methoden zum Verbindungsaufbau unterstützt:
  - Lokaler Client
  - Lokaler Server
  - IIOP

Supported Sessions ► Connectors ▼	Local Client Session	Local Server Session	IIOP session
Domino Change Detection Connector	Yes	No	Yes
Domino Users Connector	Yes	Yes	Yes
Lotus Notes Connector	Yes	Yes	Yes
Domino AdminP Connector	No	Yes	Yes



# TDI Die Verbindung zu Domino

- IIOP bietet die größte Flexibilität !
- Um eine Verbindung zu Lotus Domino per IIOP zu ermöglichen benötigen Sie:
  - DIIOP muss als Servertask auf dem Domino Server laufen und konfiguriert sein
  - Einen Internet Benutzer mit den entsprechenden Berechtigungen
  - Ein jar-Archiv von Ihrem Domino Server. Kopieren Sie bitte von `\Domino\Data\Domino\java\NCSO.jar` nach `TDI_Install\jars\3rdparty\IBM`



# TDI Die Verbindung zu Domino

- Im Lookup, Update oder Delete Mode verwendet der Konnektor eine Volltextsuche um den entsprechenden Eintrag zu finden. Dies führt zu:
  - inkonsistenten Daten (wenn der FT-Index) nicht aktuell ist
  - oder schlechter Performance (temp. FT-Index wird bei jeder Suche aufgebaut)
  - Die Lösung: Die Option „Immer Formelsuche verwenden“ in den Advanced Einstellungen des Konnektors.

Advanced

SSL verwenden  ?

Rich Text-Elemente unterstützen  ?

Dokumentauswahl  ?

**Immer Formelsuche verwenden**  ?

Datenbankansicht  Auswählen... ?



# TDI Die Verbindung zu SAP

- Verbindungen zu SAP sind möglich über:
  - **SAP ABAP AS RFC-Funktionskomponente**
  - SAP-ALE-Idoc Konnektor
    - nur SAP ausgehend
  - SAP ABAP AS Business Object Repository-Konnektor
  - SAP ABAP AS-Benutzerregistry-Konnektor
  - WEB-Service Konnektor
  - ....



# TDI Die Verbindung zu SAP

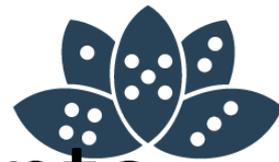
- TDI benutzt den SAP Java Connector (JCo) um die Verbindung zu SAP herzustellen
- Um Verbindungen per JCo herzustellen benötigen Sie:
  - Einen SAP Benutzer mit den entsprechenden Berechtigungen
  - JCo Version 2.1.6 oder höher (nicht V 3.x)
    - Download über SAP Marketplace <http://service.sap.com/connectors>
      - > SAP Java Connector
      - > Tools & Services
      - > Download SAP JCo Release 2.1.10



# TDI Die Verbindung zu SAP

- JCo nach dem Download in ein beliebiges Verzeichnis entpacken
- sapjco.jar kopieren nach  
>> TDI\_Install\jars\3rdparty\others
- Librfc32.dll und sapjcorfc.dll kopieren nach  
>> TDI\_Install\libs

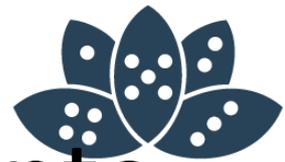
Name	Date modified
demo	08.03.2012 12:52
docs	08.03.2012 12:52
librfc32.dll	10.05.2011 14:59
Readme.txt	10.05.2011 14:59
sapjco.jar	10.05.2011 14:59
sapjcorfc.dll	10.05.2011 14:59



# TDI SAP RFC-Funktionskomponente

- Ermöglicht den Aufruf von SAP BAPI's oder RFC-Funktionen
- Die Funktionskomponente benötigt als Input zwei Attribute:
  - requestType
    - Die Art des Aufrufs
  - request
    - Der eigentliche Aufruf

Eingabezuordnung	Ausgabezuordnung	Hooks	Verbindung
Zuordnung	Hinzufügen	Löschen	Mehr...
Zuordnung	Komponentenattribut		
work.request	request		
work.requestType	requestType		

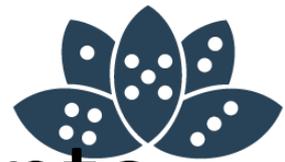


# TDI SAP RFC-Funktionskomponente

- Es werden 3 Request Typen unterstützt:

Typ	requestType	request
XML Dokument	xmlDomDocument	org.w3c.dom.Document
XML String	xmlString	java.lang.String
Multi-value Attribut	multiValuedAttributes	com.ibm.di.entry.Attribute

- Analog zum requestType wird auch das Ergebnis des SAP Aufrufs zurückgeliefert



# TDI SAP RFC-Funktionskomponente

- Welchen requestType soll ich verwenden ?
  - Welche Daten liefert der SAP Aufruf zurück und wie sollen diese weiter verarbeitet werden ?
  - Generell gilt:
    - multiValuedAttributes ist ideal für einfache Funktionsaufrufe
    - xmlString bzw. xmlDomDocument für komplexe Datenstrukturen
- TDI kann XML-Daten über verschiedene Wege weiter verarbeiten. [TDI Users - How To's](#)



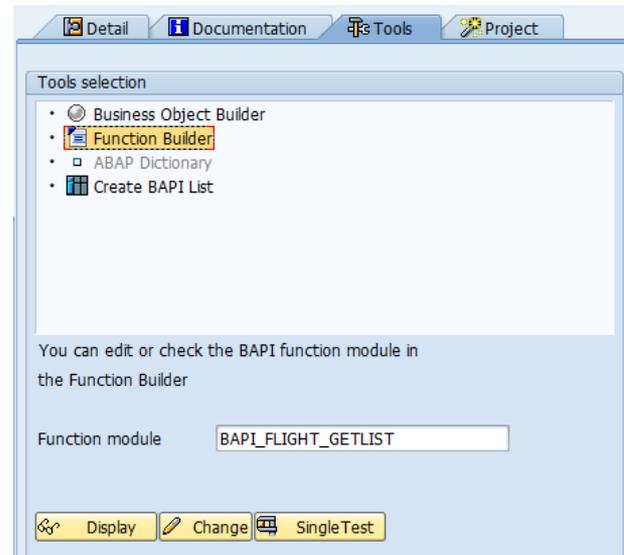
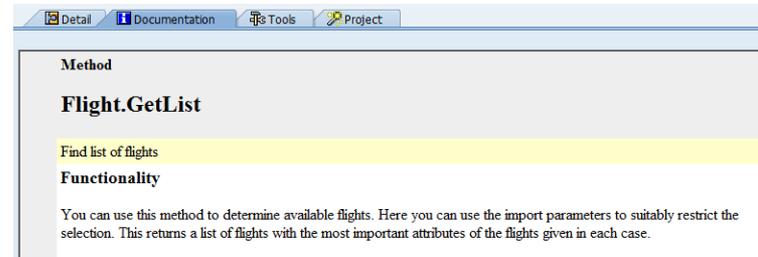
# TDI SAP Requests erzeugen

- Wie fülle ich den request mit Werten?
- Wichtig ist zunächst die definierten Ein- und Ausgabeparameter des SAP Aufrufs zu ermitteln
- 4 Tools die hierbei helfen:
  - Bapi Explorer (Bapi)
  - Function Builder (SE37)
  - SAPJcoUtil.jar
  - TDI AssemblyLine 0.98\_SAP\_GETFC\_Interface



# TDI SAP Requests erzeugen

- BAPI Explorer
  - Dokumentation
  - Name des Function Modules
  - Link zum Function Builder





# TDI SAP Requests erzeugen

- Function Builder
  - Test Aufruf
  - Zugriff auf den Sourcecode

**Test Function Module: Initial Screen**

Debugging Test data directory

Test for function group: SAPBC\_BAPI\_SFLIGHT  
Function module: BAPI\_FLIGHT\_GETLIST  
Uppercase/Lowercase:   
RFC target sys: \_\_\_\_\_

Import parameters	Value
AIRLINE	LH
DESTINATION_FROM	FR
DESTINATION_TO	FR
MAX_ROWS	0

Tables	Value
DATE_RANGE	0 Entries
EXTENSION_IN	0 Entries
FLIGHT_LIST	0 Entries
EXTENSION_OUT	0 Entries
RETURN	0 Entries

85 Entries

AIR	AIRLINE	CONN	FLIGHTDATE	AIR	CITYFROM	AIR	CITYTO	DEPTIME	ARRTIME	ARRDATE	PRICE	CURR	CUR
LH	Lufthansa	0400	17.09.2011	FRA	FRANKFURT	JFK	NEW YORK	10:10:00	11:34:00	17.09.2011	666,0000	EUR	EUR
LH	Lufthansa	0400	15.10.2011	FRA	FRANKFURT	JFK	NEW YORK	10:10:00	11:34:00	15.10.2011	666,0000	EUR	EUR
LH	Lufthansa	0400	12.11.2011	FRA	FRANKFURT	JFK	NEW YORK	10:10:00	11:34:00	12.11.2011	666,0000	EUR	EUR
LH	Lufthansa	0400	10.12.2011	FRA	FRANKFURT	JFK	NEW YORK	10:10:00	11:34:00	10.12.2011	666,0000	EUR	EUR
LH	Lufthansa	0400	07.01.2012	FRA	FRANKFURT	JFK	NEW YORK	10:10:00	11:34:00	07.01.2012	666,0000	EUR	EUR
LH	Lufthansa	0400	04.02.2012	FRA	FRANKFURT	JFK	NEW YORK	10:10:00	11:34:00	04.02.2012	666,0000	EUR	EUR



# TDI SAP Requests erzeugen

SAPJcoUtil.jar Simple Java Programm

→Aufruf über getRFCStructure.bat

```
C:\Windows\system32\cmd.exe

C:\DeveloperCamp2012>Echo off
SAP Host:
minisap
SAP Language (en):
en
SAP System No.:
00
SAP Client:
001
SAP User:
bcuser
SAP Password:

SAP RFC or BAPI Name:
BAPI_FLBOOKING_CREATEFROMDATA
Write Output to Directory e.g C:/temp/ (must exist):
c:/developmentcamp2012/
Output written to c:/developmentcamp2012/BAPI_FLBOOKING_CREATEFROMDATA.html
Output written to c:/developmentcamp2012/BAPI_FLBOOKING_CREATEFROMDATA.xml
Press any key to continue . . . _
```



# TDI SAP Requests erzeugen

## SAPJcoUtil.jar – Ergebnis

BAPI\_FLBOOKING\_CREATEFROMDATA Date: Thu Mar 08 17:02:15 CET 2012

**Input Parameters**

Parameters: INPUT

Index:	1	2	3
Name:	BOOKING_DATA	RESERVE_ONLY	TEST_RUN
Type:	STRUCTURE	CHAR	CHAR
Size:	12	1	1
Offset:	0	12	14
Decimals:	0	0	0
Default:		SPACE	SPACE
Value:	BAPISBONEW		

Structure: BOOKING\_DATA (BAPISBONEW)

Index:	1	2	3	4	5	6	7	8	9	10
Name:	AIRLINEID	CONNECTID	FLIGHTDATE	CUSTOMERID	CLASS	COUNTER	AGENCYNUM	PASSNAME	PASSFORM	PASSBIRTH
Type:	CHAR	NUM	DATE	NUM	CHAR	NUM	NUM	CHAR	CHAR	DATE
Size:	3	4	8	8	1	8	8	25	15	8
Offset:	0	6	14	30	46	48	64	80	130	160
Decimals:	0	0	0	0	0	0	0	0	0	0
Value:	" "	0000	0000-00-00	00000000	" "	00000000	00000000	" "	" "	0000-00-00

**Output Parameters**

Parameters: OUTPUT

Index:	1	2	3
Name:	AIRLINEID	BOOKINGNUMBER	TICKET_PRICE
Type:	CHAR	NUM	STRUCTURE
Size:	3	8	12
Offset:	0	6	22
Decimals:	0	0	0
Value:			

Structure: TICKET\_PRICE (BAPISBOPRI)

Index:	1	2	3	4
Name:	PRICE	TAX	CURR	CURR_ISO
Type:	BCD	BCD	CHAR	CHAR
Size:	12	12	5	3
Offset:	0	12	24	34
Decimals:	4	4	0	0
Value:	0	0	" "	" "





# TDI SAP Requests erzeugen

- DEMO



# TDI SAP Requests erzeugen

## TDI AssemblyLine 0.98\_SAP\_GETFC\_Interface - Ergebnis

```
<?xml version='1.0' encoding='UTF-16'?>
<doc:BAPI_FLBOOKING_CREATEFROMDATA xmlns:doc="urn:sap-com:document:sap:business:rfc">
  <AIRLINEID><!-- MANDATORY ## EXPORT ## Airline Code-->
  <BOOKINGNUMBER><!-- MANDATORY ## EXPORT ## Booking number-->
  <TICKET_PRICE><!-- MANDATORY ## EXPORT ## Price-->
  <BOOKING_DATA><!-- MANDATORY ## IMPORT ## Booking Data-->
    <AIRLINEID></AIRLINEID>
    <CONNECTID></CONNECTID>
    <FLIGHTDATE></FLIGHTDATE>
    <CUSTOMERID></CUSTOMERID>
    <CLASS></CLASS>
    <COUNTER></COUNTER>
    <AGENCYNUM></AGENCYNUM>
    <PASSNAME></PASSNAME>
    <PASSFORM></PASSFORM>
    <PASSBIRTH></PASSBIRTH>
  </BOOKING_DATA>
  <RESERVE_ONLY><!-- OPTIONAL ## IMPORT ## Reservation only-->
  <TEST_RUN><!-- OPTIONAL ## IMPORT ## Switch to simulation mode-->
  <EXTENSION_IN><!-- OPTIONAL ## TABLE ## Import customer enhancements-->
  <RETURN><!-- OPTIONAL ## TABLE ## Return Messages-->
</doc:BAPI_FLBOOKING_CREATEFROMDATA>
```



# TDI SAP Requests erzeugen

## Am Beispiel BAPI\_FLIGHT\_GETLIST

```
<?xml version='1.0' encoding='UTF-16'?>
<doc:BAPI_FLIGHT_GETLIST xmlns:doc="urn:sap-com:document:sap:business:rfc">
  <AIRLINE><!-- OPTIONAL ## IMPORT ## Select airline-->
  <DESTINATION_FROM><!-- OPTIONAL ## IMPORT ## Select departure city-->
  <DESTINATION_TO><!-- OPTIONAL ## IMPORT ## Select destination-->
  <MAX_ROWS><!-- OPTIONAL ## IMPORT ## Maximum Number of Lines of Hits-->
  <DATE_RANGE><!-- OPTIONAL ## TABLE ## Selection range for flight date-->
  <EXTENSION_IN><!-- OPTIONAL ## TABLE ## Import customer enhancements-->
  <EXTENSION_OUT><!-- OPTIONAL ## TABLE ## Export customer enhancements-->
  <FLIGHT_LIST><!-- OPTIONAL ## TABLE ## List of flights-->
  <RETURN><!-- OPTIONAL ## TABLE ## Return Messages-->
</doc:BAPI_FLIGHT_GETLIST>
```



# TDI SAP Requests erzeugen

1. Entfernen aller Return und Optionalen Parameter die wir nicht verwenden wollen

```
<?xml version='1.0' encoding='UTF-16'?>
<doc:BAPI_FLIGHT_GETLIST xmlns:doc="urn:sap-com:document:sap:business:rfc">
  <AIRLINE><!-- OPTIONAL ## IMPORT ## Select airline-->
  <DESTINATION_FROM><!-- OPTIONAL ## IMPORT ## Select departure city-->
  <DESTINATION_TO><!-- OPTIONAL ## IMPORT ## Select destination-->
  <MAX_ROWS><!-- OPTIONAL ## IMPORT ## Maximum Number of Lines of Hits-->
  <DATE_RANGE><!-- OPTIONAL ## TABLE ## Selection range for flight date-->
  <EXTENSION_IN><!-- OPTIONAL ## TABLE ## Import customer enhancements-->
  <EXTENSION_OUT><!-- OPTIONAL ## TABLE ## Export customer enhancements-->
  <FLIGHT_LIST><!-- OPTIONAL ## TABLE ## List of flights-->
  <RETURN><!-- OPTIONAL ## TABLE ## Return Messages-->
</doc:BAPI_FLIGHT_GETLIST>
```



# TDI SAP Requests erzeugen

## 2. Parameter füllen

wichtig: Bei XML-Requests müssen alle Werte als CDATA-Sektionen übergeben werden

## 3. Fertig

```
<?xml version='1.0' encoding='UTF-16'?>
<doc:BAPI_FLIGHT_GETLIST xmlns:doc="urn:sap-com:document:sap:business:rfc">
<AIRLINE><!-- OPTIONAL ## IMPORT ## Select airline-->
<![CDATA[LH]]>
</AIRLINE>
</doc:BAPI_FLIGHT_GETLIST>
```

Name/Ausdruck	Wert	Java-Klasse
work	{ "requestType": "multiValuedAttributes", "request": ""BAPI_FLIGH...	com.ibm.di.entry.Entry
request	"request": ""BAPI_FLIGHT_GETLIST": ""AIRLINE": "LH""	com.ibm.di.entry.Attribute
0	"BAPI_FLIGHT_GETLIST" ""AIRLINE": "LH""	com.ibm.di.entry.Attribute
0	"AIRLINE": "LH"	com.ibm.di.entry.Attribute
0	LH	java.lang.String
requestType	"requestType": "multiValuedAttributes"	com.ibm.di.entry.Attribute



# TDI SAP Requests erzeugen

- Im TDI-Projekt zum Developercamp sind verschiedene Beispiele enthalten wie Requests per Script erzeugt und manipuliert werden können.

```
var reqAttr = system.newAttribute("request");
var rfc = system.newAttribute("BAPI_FLIGHT_GETLIST");
var attr1 = system.newAttribute("AIRLINE");
attr1.addValue("LH");
rfc.addValue(attr1);
reqAttr.addValue(rfc);
return reqAttr;
```

```
var l_requestxml = parseXMLStringtoDOM(work.XML_TEMPLATE_STRING);
var l_dateformatter = new java.text.SimpleDateFormat("yyyyMMdd");
var l_Calendar = new java.util.GregorianCalendar(1956,java.util.Calendar.MARCH,17);

.....
l_passbirth_items=l_requestxml.getElementsByTagName("PASSBIRTH");
....
l_tmpCDATA = l_requestxml.createCDATASection(l_dateformatter.format(l_Calendar.getTime()));
l_passbirth_items[0].appendChild(l_tmpCDATA);
....
ret.value=l_requestxml;
```



# TDI SAP Response auswerten

- Der SAP requestType bestimmt den responseType !
- Die Antwortstruktur eines SAP-Aufrufs ist immer identisch
- Welche Daten liefert der Aufruf zurück ?
  1. multiValuedAttributes

doc:BAPI\_FLIGHT\_GETLIST

+ AIRLINE
+ DESTINATION_FROM
+ DESTINATION_TO
+ MAX_ROWS
+ DATE_RANGE
+ EXTENSION_IN
+ EXTENSION_OUT
+ FLIGHT_LIST
+ RETURN

[-] response		com.ibm.di.entry.Attribute
[-] 0		com.ibm.di.entry.Attribute
+ 0	"AIRLINE": ""	com.ibm.di.entry.Attribute
+ 1	"DESTINATION_FROM": [""AIRPORTI...	com.ibm.di.entry.Attribute
+ 2	"DESTINATION_TO": [""AIRPORTID":...	com.ibm.di.entry.Attribute
+ 3	"MAX_ROWS": "0"	com.ibm.di.entry.Attribute
+ 4	"DATE_RANGE": []	com.ibm.di.entry.Attribute
+ 5	"EXTENSION_IN": []	com.ibm.di.entry.Attribute
+ 6	"EXTENSION_OUT": []	com.ibm.di.entry.Attribute
+ 7		com.ibm.di.entry.Attribute
+ 8	"RETURN": ""item0": [""TYPE": "S", ""...	com.ibm.di.entry.Attribute





# TDI SAP Response auswerten

- Über den Default Success Hook kann das relevante Ergebnis z.B. in eine eigenes Work Attribut gespeichert werden.

```
call_SAP_FC  
  Default Success  
  Default On Error  
work.setAttribute("FlightList",work.response[0][7]);
```

- Das Verarbeiten der Daten kann dann z.B. mit einer „FOR-Each-Attribut“ Schleife erfolgen

Name/Ausdruck	Wert	Java-Klasse
work		com.ibm.di.entry.Entry
request	"request": "BAPI_FLIGHT_GETLIST":...	com.ibm.di.entry.Attribute
requestType	"requestType": "multiValuedAttributes"	com.ibm.di.entry.Attribute
response		com.ibm.di.entry.Attribute
0		com.ibm.di.entry.Attribute
0	"AIRLINE": ""	com.ibm.di.entry.Attribute
1	"DESTINATION_FROM": ["AIRPORTID":...	com.ibm.di.entry.Attribute
2	"DESTINATION_TO": ["AIRPORTID":...	com.ibm.di.entry.Attribute
3	"MAX_ROWS": "0"	com.ibm.di.entry.Attribute
4	"DATE_RANGE": []	com.ibm.di.entry.Attribute
5	"EXTENSION_IN": []	com.ibm.di.entry.Attribute
6	"EXTENSION_OUT": []	com.ibm.di.entry.Attribute
7		com.ibm.di.entry.Attribute
0	"item0": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute
1	"item1": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute
2	"item2": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute
3	"item3": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute
4	"item4": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute
5	"item5": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute
6	"item6": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute
7	"item7": ["AIRLINEID": "AA", "AIRLI...	com.ibm.di.entry.Attribute



# TDI SAP Response auswerten

## 2. XML Response

- xmlDomDocument zu XML-String umwandeln
  - Function serializeDOM in XmlUtil
- Weitere Verarbeitung über:
  - Parser Funktionskomponente
  - Script (XMLTag2Entry)
  - .....

```
doc:BAPI_FLIGHT_GETLIST
├── AIRLINE
├── DESTINATION_FROM
├── DESTINATION_TO
├── MAX_ROWS
├── DATE_RANGE
├── EXTENSION_IN
├── EXTENSION_OUT
├── FLIGHT_LIST
└── RETURN
```

```
doc:BAPI_FLIGHT_GETLIST.Response
├── AIRLINE LH
├── DESTINATION_FROM
├── DESTINATION_TO
├── MAX_ROWS 0
├── DATE_RANGE
├── EXTENSION_IN
├── EXTENSION_OUT
├── FLIGHT_LIST
│   └── item
│       ├── AIRLINEID LH
│       ├── AIRLINE Lufthansa
│       ├── CONNECTID 0400
│       ├── FLIGHTDATE 20110917
│       ├── AIRPORTFR FRA
│       ├── CITYFROM FRANKFURT
│       ├── AIRPORTTO JFK
│       ├── CITYTO NEW YORK
│       ├── DEPTIME 101000
│       ├── ARRTIME 113400
│       ├── ARRDATE 20110917
│       ├── PRICE 666.0000
│       ├── CURR EUR
│       └── CURR_ISO EUR
└── item LH Lufthansa 0400 20111015 FRA FRANKFURT JFK
└── RETURN S BC IBF 000 Method was executed successfully 000
```



# TDI Errorhandling

- Zwei unterschiedliche Fehlerarten müssen beachtet werden:
  - Fehler die von der SAP Funktionskomponenten stammen
    - Wie z.b.:
      - Falscher Username/Passwort
      - Account deaktiviert
      - Fehlende Attribute
      - Feldlänge zu kurz oder zu lange
    - Können wie gewohnt über Hooks abgefangen werden
      - „Default On Error“ oder „Funktionsfehler“

## Default On Error

```
errorMsg("Error in: " + task.getShortName());  
work.addAttributeValue("SAP_RESPONSE_MESSAGE", error.message);  
work.addAttributeValue("SAP_RESPONSE_TYPE", "E");  
system.skipEntry();
```



# TDI Errorhandling

- Fehler die vom BAPI bzw. RFC-Aufruf in SAP zurückgeliefert werden:
  - Werden in der Return Tabelle des SAP Response ausgegeben
  - Type „E“
  - Die Fehlermeldung befindet sich im Attribut „Message“
  - Erfolgreiche SAP Aufrufe haben den Type „S“

```
doc:BAPI_FLBOOKING_CREATEFROMDATA.Response
├── BOOKING_DATA LH 0400 20120818 00000001 C 00000000 000
├── RESERVE_ONLY
├── TEST_RUN
├── AIRLINEID
├── BOOKINGNUMBER 00000000
├── TICKET_PRICE 0 0
├── EXTENSION_IN
├── RETURN
│   ├── item
│   │   ├── TYPE E
│   │   ├── ID BC_IBF
│   │   ├── NUMBER 105
│   │   ├── MESSAGE No travel agency or counter specified
│   │   ├── LOG_NO
│   │   ├── LOG_MSG_NO 000000
│   │   ├── MESSAGE_V1
│   │   ├── MESSAGE_V2
│   │   ├── MESSAGE_V3
│   │   ├── MESSAGE_V4
│   │   ├── PARAMETER BOOKING_DATA
│   │   ├── ROW 0
│   │   ├── FIELD
│   │   └── SYSTEM
│   └── item
│       └── TYPE E
```



# Agenda

- TDI Übersicht
- TDI Installation
- TDI Konfigurationseditor
- **TDI SAP Schnittstellen Beispiele**
  - Von SAP generiertes CSV-File
  - BAPI oder RFC-Aufruf
- TDI Tipps und Tricks



# TDI SAP Schnittstellen

- Beispiel 1 – CSV Import
- Unser Ziel:
  - Daten aus einer CSV Datei in eine Domino Datenbank importieren
  - Nichts weltbewegendes
  - Ausführen und Debuggen einer Fertigungslinie im Konfigurationseditor
- Die verwendeten Datenquellen sowie das komplette TDI Projekt erhalten Sie zusammen mit der Präsentation



# TDI SAP Schnittstellen

- DEMO



# TDI SAP Schnittstellen

- Beispiel 2 – BAPI Aufruf
- Unser Ziel:
  - Aufruf von BAPI\_FLBOOKING\_CREATE\_FROM\_DATA
  - Request als xmlDomDocument
  - Verwenden des TaskCallBack um einen Parameter zu setzen
  - Verwenden der Parser Funktionskomponente
  - Ansätze zum Errorhandling
- Die verwendeten Datenquellen sowie das komplette TDI Projekt erhalten Sie zusammen mit der Präsentation



# TDI SAP Schnittstellen

- DEMO



# Agenda

- TDI Übersicht
- TDI Installation
- TDI Konfigurationseditor
- TDI SAP Schnittstellen Beispiele
- TDI Tipps und Tricks



# TDI – Tipps und Tricks

- Verwenden Sie eindeutige sprechende Namen für Ihre Komponenten
- Ändern Sie immer die „Default Namen“ der Komponenten
- Vermeiden Sie die Verwendung von Leer- und Sonderzeichen
- Komponenten können über „F2“ oder das Kontextmenü umbenannt werden
- Der Konfigurations Editor unterstützt in viele Bereichen Drag & Drop



# TDI – Tipps und Tricks

- Die Reihenfolge der Komponenten in der FL kann per Drag & Drop geändert werden. Oder Komponenten können in „Verzweigungen“ verschoben werden.

Sync\_Domino\_2\_AD

Komponente hinzufügen    Zuordnung anzeigen    Option

- ▲ Zuführung
  - ▲ LotusNotes\_AdminCamp\_Iterator
    - Default On Error
- ▲ Datenfluss
  - Log\_Entry
  - Prepare\_ActiveDirectory\_Attributes
  - ▲ LotusNotes\_USState\_Lookup
    - On Multiple Entries
    - On No Match
    - Default On Error
  - ▲ IF: IF\_State\_Long\_available
    - LDAP\_AD\_Update
    - Log\_Entry
  - ▲ ELSE: ELSE\_State\_Long\_not\_available

(leere Verzweigung - doppelt klicken, um eine Komponente...



# TDI – Tipps und Tricks

- Maximale Anzahl der „Iterationen“ kann in den Einstellungen der Fertigungslinie konfiguriert werden, z.B. für Tests

The screenshot shows the 'Einstellungen für Fertigungslinie' (Production Line Settings) dialog box. The title bar reads 'Einstellungen für Fertigungslinie'. The main area contains the following settings:

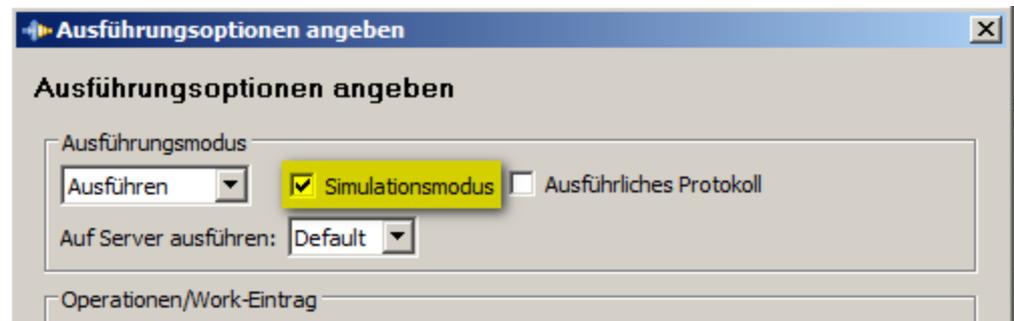
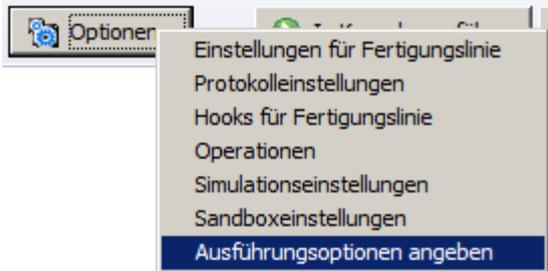
- Taskparameter laden aus: [Empty text box] [Auswählen...] [?]
- Taskparameter speichern in: [Empty text box] [Auswählen...] [?]
- Intervall für Statistikprotokollierung: [Empty dropdown menu] [?]
- Maximale Anzahl Lesevorgänge (Iterator): [10] [?]
- Maximale Anzahl Fehler: [Empty dropdown menu] [?]
- Die maximal zulässige Anzahl der Einträge, die bei Suchvorgängen zurückgegeben wurden: [10] [?]
- Alle globalen Prologe einschließen:  [?]
- Zusätzliche Prologe einschließen: [Empty list box] [Auswählen...] [?]
- Alle Attribute automatisch zuordnen:  [?]

An 'OK' button is located at the bottom right of the dialog.



# TDI – Tipps und Tricks

- Simulation einer Fertigungslinie:
  - Keine Änderung in den Zielsystemen





# TDI – Tipps und Tricks

- Der Datenbrowser bietet eine einfache Möglichkeit die gewünschten Attribute zu identifizieren.
- Stellen Sie die Verbindungsoptionen im ersten Schritt so ein dass er Ihnen nur einen „gepflegten“ Datensatz zurückliefert.

LotusNotes\_AdminCamp\_Iterator

Modus **Iterator** Status **Aktiviert** Übernehmen von **system:/Connectors/ibmdi.Notes** **Mehr...**

Eingabezuordnung | Hooks | Delta | Verbindung | Verbindungsfehler

Zuordnung **Hinzufügen** **Löschen** **Mehr...**

Work-Attribut	Zuordnung
FirstName	conn.FirstName
Form	conn.Form
FullName	conn.FullName
InternetAddress	conn.InternetAddress
LastName	conn.LastName
NotelD	conn.NotelD
OfficeCity	conn.OfficeCity
OfficeCountry	conn.OfficeCountry
OfficeFAXPhoneNumber	conn.OfficeFAXPhoneNumber
OfficePhoneNumber	conn.OfficePhoneNumber
OfficeState	conn.OfficeState
OfficeStreetAddress	conn.OfficeStreetAddress
OfficeZIP	conn.OfficeZIP
Type	conn.Type
UNID	conn.UNID
WebSite	conn.WebSite

Verbindung herstellen | **Nächster** | **Schließen** | **Mehr...**

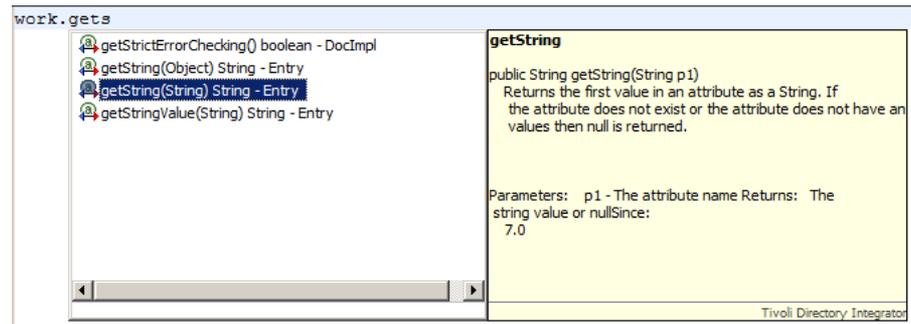
Schema

Name	Musterwert
SUpdatedBy	CN=Andreas Artner/OU=UL/OU=DE/O=FuM
FirstName	Abraham
Form	Person
FullName	Abraham Arview/AdminCamp
InternetAddress	abraham@arview.com
LastName	Arview
NotelD	3866
OfficeCity	Trenton
OfficeCountry	Mercer
OfficeFAXPhoneNumber	609-890-6446
OfficePhoneNumber	609-890-4008
OfficeState	NJ
OfficeStreetAddress	4441 Nottingham Way
OfficeZIP	8690
Type	Person
UNID	807B7AC1FB414326C125790C004971A6
WebSite	http://www.abrahamarview.com

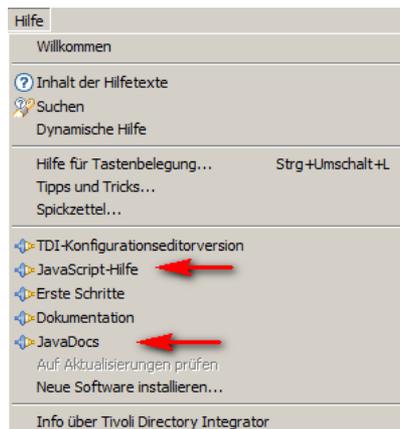


# TDI – Tipps und Tricks

- Der Konfigurationseditor unterstützt
  - typeahead und JavaDoc



- Zugriff auf die Dokumentation über das Hilfe Menü





# TDI – Tipps und Tricks

- In den Verbindungsoptionen eines jeden Konnektors oder einer Funktionskomponente kann ein „Ausführliches Protokoll“ aktiviert werden.

A screenshot of the LotusNotes\_AdminCamp\_Iterator configuration window. The window title is "LotusNotes\_AdminCamp\_Iterator". It features a "Modus" dropdown set to "Iterator" and a "Status" dropdown set to "Aktiviert". Below these are tabs for "Eingabezuordnung", "Hooks", "Delta", "Verbindung", and "Verbindungsfehler". The "Verbindung" tab is active, showing fields for "IP-Adresse des Domino-Servers" (127.0.0.1), "Name des Domino-Servers", "HTTP-Port" (80), "Sitzungstyp" (IIOP), "Benutzername" (aartner), "Kennwort" (\*\*\*), "Datenbank" (AdminCamp\demo.nsf), and "Kommentar". At the bottom, there is a checkbox for "Ausführliches Protokoll" which is currently unchecked.

LotusNotes\_AdminCamp\_Iterator

Modus **Iterator** Status **Aktiviert** Übernehmen von [ ]

Eingabezuordnung | Hooks | Delta | **Verbindung** | Verbindungsfehler

IP-Adresse des Domino-Servers \* 127.0.0.1

Name des Domino-Servers

HTTP-Port 80

Sitzungstyp \* IIOP

Benutzername aartner

Kennwort \* \*\*\*

Datenbank \* AdminCamp\demo.nsf

Kommentar

Ausführliches Protokoll



# TDI – Tipps und Tricks

- Zugriff auf Domino-Java Objekte:
  - z.b. um alle Datensätze zu löschen

## After Initialize

```
//Beispiel für Zugriff auf Lotus Domino Klassen aus TDI
var db = thisConnector.connector.getDominoDatabase(null);
var dc = db.getAllDocuments();
task.logmsg("INFO","Deleting: " + dc.getCount() + " Documents from Target DB: " + db.getTitle());
dc.removeAll(true);
```



# TDI – Tipps und Tricks

- Delta Engine:
  - Verfügbar für alle Konnektoren
  - Optimal um z.B. Löschungen in der Quelle zu identifizieren
  - Performancegewinn bei großen Datenmengen nur ein Subset der Datensätze durchläuft die Fertigungslinie
  - Inhalt wird in einer relationalen DB gecached um das Delta zu ermitteln
  - `work.getOperation()` liefert die aktuelle Operation (add, modify, delete...) zurück.
  - Beispiel: 1.01\_SAP\_CSV\_2\_Domino\_DELTA

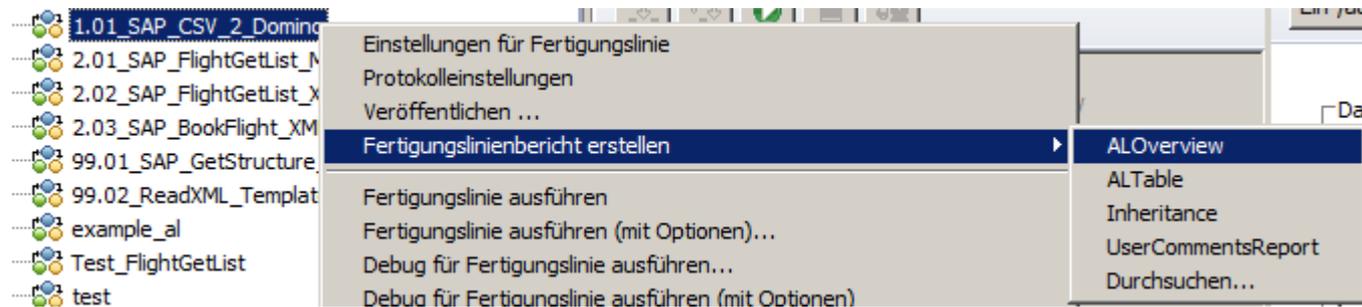
The screenshot shows the configuration window for the Dateisystemconnector\_Read\_CSV connector. The 'Delta' tab is selected, and the 'Delta konfigurieren' section is visible. The settings are as follows:

Option	Value
Delta aktivieren	<input checked="" type="checkbox"/>
Eindeutiger Attributname	CONNECTID+FLIGHTDATE
Deltaspeicher	FLIGHT_CSV
Gelöschte lesen	<input checked="" type="checkbox"/>
Gelöschte entfernen	<input checked="" type="checkbox"/>
Nicht geänderte zurückgeben	<input type="checkbox"/>
Festschreiben	Nach jeder Datenbankoperation
Zeilensperre	Serialisierbar
Schnellerer Algorithmus	<input type="checkbox"/>
Doppelte Deltaschlüssel zulassen	<input type="checkbox"/>
Änderungserkennungsmodus	Alle Attribute für die Änderungserkennung verwenden
Attributliste	



# TDI – Tipps und Tricks

- Dokumentation automatisch erstellen:



- Einfache XSLT Transformation kann angepasst werden !



# TDI – Tipps und Tricks

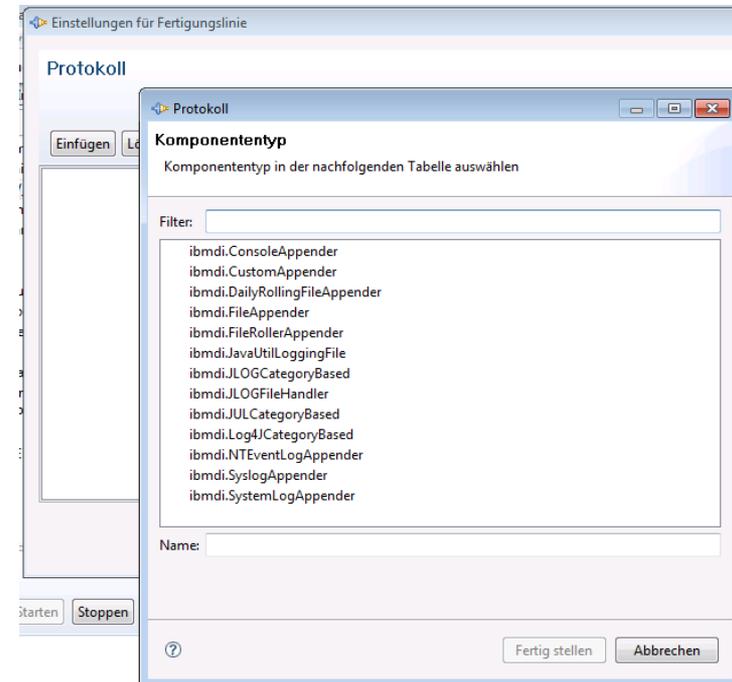
- Achtung beim Aufruf von Java Klassen:  
return java.lang.**Duble**.parseDouble(conn.PRICE)  
führt zu:

```
10:34:48,796 ERROR - CTGDIS266E Fehler in NextIteratorEntry. Es ist eine Ausnahmebedingung aufgetreten: com.ibm.js:  
com.ibm.jscript.InterpretException: Script interpreter error, line=1, col=24: Reference error: 'java' not found  
  at com.ibm.jscript.types.FBSReferenceByName$JavaPackageVariable.getValue (FBSReferenceByName.java:64)  
  at com.ibm.jscript.ASTTree.InterpretResult.getFBSValue (InterpretResult.java:40)  
  at com.ibm.jscript.ASTTree.ASTCall.interpret (ASTCall.java:86)  
  at com.ibm.jscript.ASTTree.ASTReturn.interpret (ASTReturn.java:43)  
  at com.ibm.jscript.ASTTree.ASTProgram.interpretEx (ASTProgram.java:102)  
  at com.ibm.jscript.JSEExpression.interpretExpression (JSEExpression.java:399)  
  at com.ibm.jscript.JSEExpression.evaluateValue (JSEExpression.java:249)  
  at com.ibm.jscript.JSEExpression.evaluateValue (JSEExpression.java:236)  
  at com.ibm.jscript.JSEExpression.evaluateValue (JSEExpression.java:239)  
  at com.ibm.jscript.JSInterpreter.interpret (JSInterpreter.java:53)
```



# TDI – Tipps und Tricks

- Tivoli Directory Integrator benutzt *log4j*
- Default: *ibmdi.log*
- Etliche Standard Methoden sind vorhanden:
  - *Unix syslog oder Windows eventlog*
  - *Täglich rollierend*
  - .....
- Die Einstellungen zum Logging werden über die Protokoll-einstellungen vorgenommen:





# TDI – zum Mitnehmen

- Ein klasse Tool um Schnittstellen zu realisieren
- TDI ist nicht nur für Directories geeignet
- Konnektoren zu nahezu allen Systemen
- Von einfachen bis hin zu komplexen Schnittstellen – alles ist möglich
- Hoch flexibel
- TDI ist Lizenzkostenfrei
- Holen Sie sich TDI von PassportAdvantage
- Viel Spass damit !



# TDI – ist cool !

- Vielen Dank für die Aufmerksamkeit !
- Fragen ?
  - jederzeit gerne
  - auch per email

Andreas Artner  
aartner@fum.de





# TDI – zusätzliche Quellen

- IBM Tivoli Directory Integrator Users Group  
<http://www.tdi-users.org>
- IBM Tivoli Directory Integrator - Infocenter  
[http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc\\_7.1/welcome.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc_7.1/welcome.htm)
- Consultant in Your Pocket Webcasts:  
TDI Admin Perspective & TDI Developer Perspective  
<http://consultantinyourpocket.com/ciyp/ciyp.nsf/>
- Domino Integration Specific Info: <http://sites.google.com/site/dominointegration>
- IBM DeveloperWorks – Tivoli  
<https://www.ibm.com/developerworks/tivoli/>
- TDI User Forum  
<http://groups.google.com/group/ibm.software.network.directory-integrator/topics?gvc=2>
- Learning TDI  
<http://www.tdi-users.org/twiki/bin/view/Integrator/LearningTDI>
- Domino Integration with TDI  
<http://www.tdi-users.org/twiki/bin/view/Integrator/Domino>
- SAP JCo Utility Cookbook  
<http://searchsap.techtarget.com/feature/Resource-The-JCo-utility-cookbook>
- TDI Redbooks/Redpieces  
<http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=tivoli+AND+directory+AND+integrator>