



IBM Software Group | IMS

Accelerating IMS Database Enterprise Development with DLIModel Utility

Danny Nguyen
IMS Development
Silicon Valley Laboratory
IBM Corporation

Important Disclaimer

© Copyright IBM Corporation 2009. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. THE INFORMATION ON NEW PRODUCTS IS FOR INFORMATIONAL PURPOSES ONLY AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. THE INFORMATION ON ANY NEW PRODUCTS IS NOT A COMMITMENT, PROMISE, OR LEGAL OBLIGATION TO DELIVER ANY MATERIAL, CODE OR FUNCTIONALITY. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR OUR PRODUCTS REMAINS AT THE SOLE DISCRETION OF IBM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, Information Management, IMS, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.

Why Java and XML for IMS DB enterprise development?

- **Rapid Application Development**
 - Reduce total cost of ownership and time-to-market
- **Leverage existing marketplace, industry-sanctioned standards**
 - Persistent
 - Well-documented and understood
 - Standard APIs minimize specific back-end knowledge of IMS
 - Our drivers will handle that on behalf of the application
- **Leverage new and abundant skills in the marketplace**
 - Help to mitigate the reducing numbers of mainframe (COBOL, PL/I, etc.) skills
- **Integrate with other products**

How did IMS do it?

Two new IMS dependent regions introduced in IMS V7

- **JMP (Java Message Processing)**
 - For message-driven Java applications
 - New IMSJMP JOB that EXECs the new DFSJMP procedure
 - DFSJMP procedure added to IMS.PROCLIB
 - Similar to the DFSMPR procedure for MPPs

- **JBP (Java Batch Processing)**
 - For non-message driven Java applications
 - New IMSJBP JOB that EXECs the new DFSJBP procedure
 - DFSJBP procedure added to IMS.PROCLIB
 - Similar to the IMSBATCH procedure for BMPs

What else?

XML Database introduced in IMS V9, XQuery and IMS DB Web Services in IMS V10

- Introduces a way to view/map *native* IMS hierarchical data to XML documents
- Aligns IMS Database (DBD) with XML Schema
- Allows the retrieval and storage of IMS Records as XML documents with no change to existing IMS databases
- Enables query of IMS data using XQuery
 - More natural fit for hierarchical data querying
 - Enables customers to leverage emerging standard skill set
 - Enhanced product and tooling integration
- Provides ability to expose a database query as a callable web service

IMS Java, XML DB and DB web service, now what?

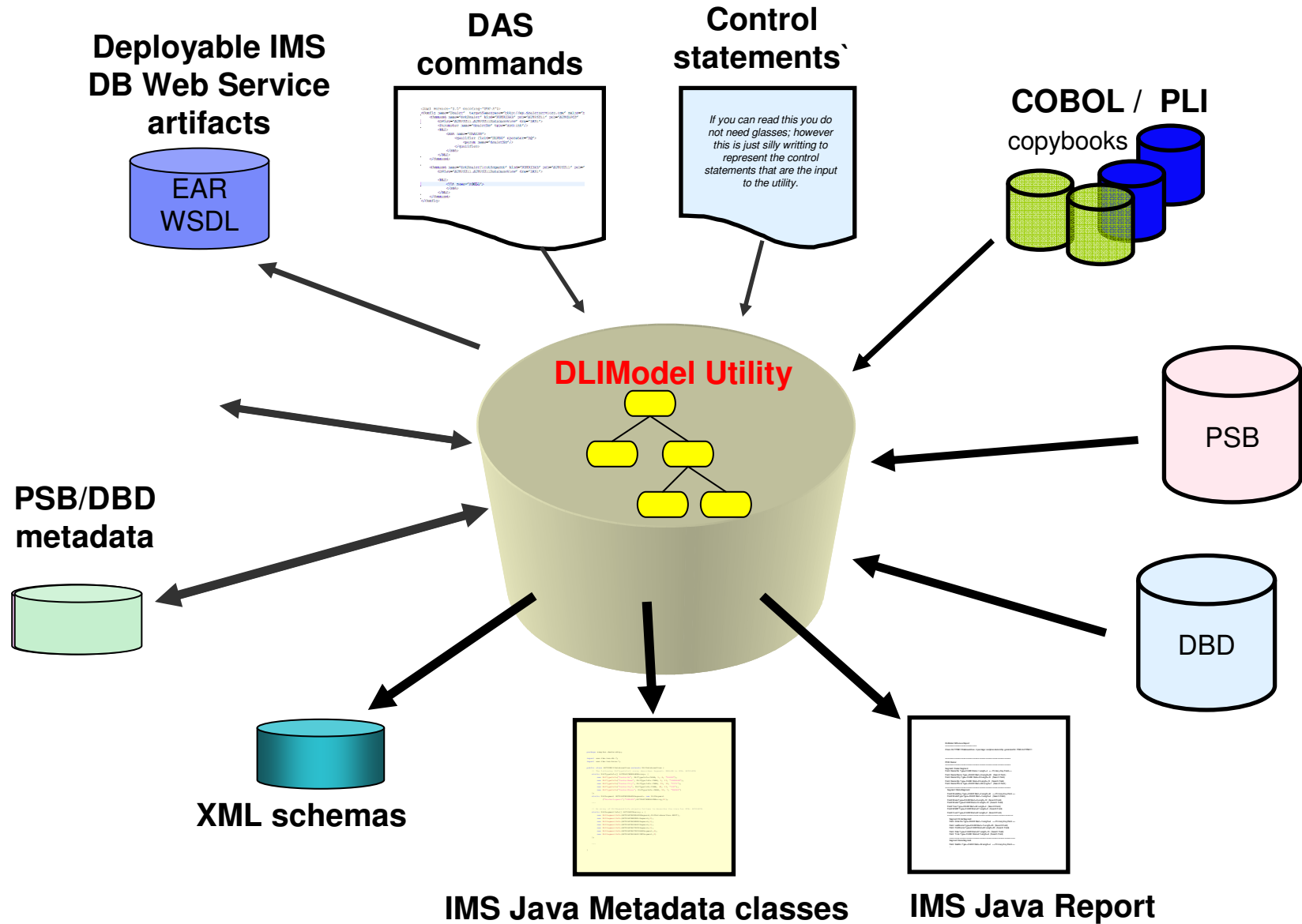
- IMS does not have a runtime metadata catalog
 - Needed a way to represent IMS database metadata (PCBs, segments, fields) for consumption at runtime.

- Development cost is a big issue
 - Skills gap
 - Manual creation of metadata extremely difficult and error prone
 - Very tedious
 - Hard to maintain and update

Solutions? DLIModel Utility

- **IMS database GUI-based visualization tool**
 - Visualizes an entire IMS PSB
 - shows each PCB and the hierarchy of the referred DBD
 - shows segments, fields, data types, etc.
- **IMS database metadata generation tool**
 - Generates the necessary metadata that is consumed at runtime by IMS JDBC drivers, XML database and XQuery
 - Generates the EAR and WSDL files for IMS DB Web Services
- **Bottom up tooling approach**
 - Parses PSB and DBD source
 - Optionally imports COBOL and PL/I copybooks
- **An Eclipse plug-in**

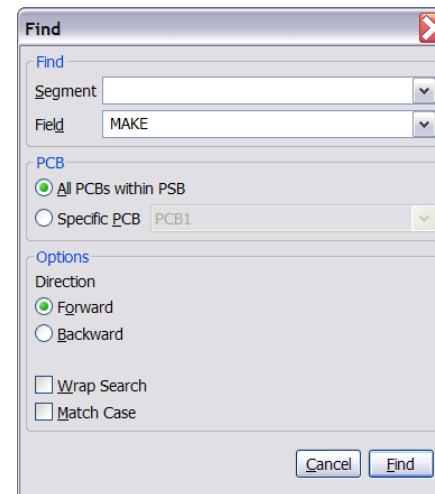
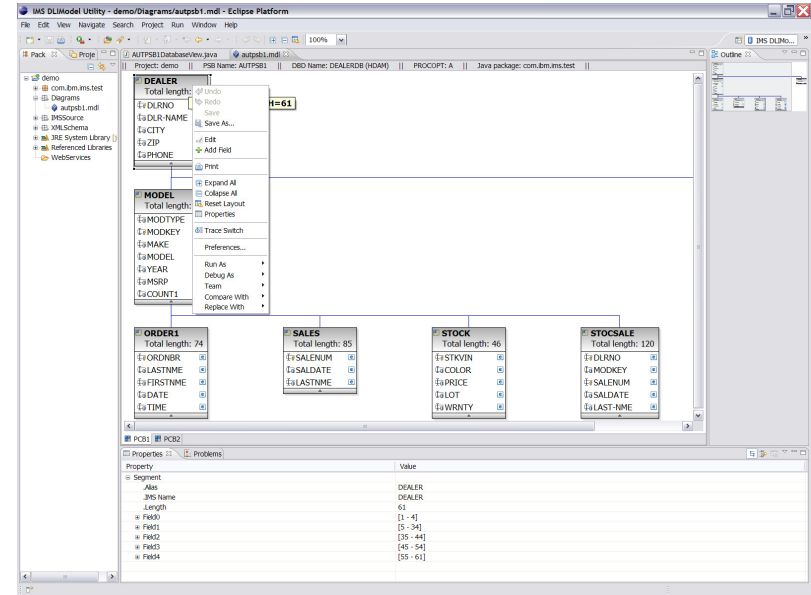
DLIModel Utility Enterprise Suite



DLIModel Utility – A visualization tool

- User can visualize an entire IMS PSB in a multi-page graphical editor.
 - Each PCB can be viewed, saved and printed individually.
 - Each PCB editor shows the hierarchy of the sensitive segments of an IMS database.
 - Each PCB, senseg, and senfld can be edited using the context menu or Properties view.

- User can also search an entire IMS PSB for a specific PCB, segment, and/or field.



DLIModel Utility – An IMS database metadata generation tool

- It has been used to generate the necessary metadata that is consumed at runtime by the IMS JDBC drivers, XML DB, XQuery and IMS DB Web services using the IMS sources as input.
 - DLIDatabaseView for IMS JDBC Universal driver
 - XML schema for XML DB and XQuery
 - Deployable artifacts (EAR and WSDL files) for IMS DB Web services via the DAS commands in a syntax assist and syntax highlight editor.

DLIModel Utility – Wizards for input IMS sources

New DLIModel Utility Project

DLIModel Utility Project
Create a new DLIModel utility project.

Project name:

Java package:

Project contents

Use default

Directory:

Select optional output files

XML schema

DLIModel utility result report

Trace log

Run utility from an IMS control statement [Find out more](#)

Location of control statement file:

New DLIModel Utility Project

Import IMS PSB and DBD Source
Click Finish to open the IMS DLIModel Utility perspective.

PSB source

From directory:

demo3

DBD

PSB

autpsb1

DBD source

Same directory as above

DLIModel Utility – DLIDatabaseView output

```

AUTPSB1DatabaseView.java x autpsb1.mdl
package autpsb1;

import com.ibm.ims.db.*;

public class AUTPSB1DatabaseView extends DLIDatabaseView {

    // This class describes the data view of PSB: AUTPSB1
    // PSB AUTPSB1 has database PCBs with 8-char PCBNAME or label:
    //     PCB1
    //     PCB2

    // The following describes Segment: DEALER ("DEALER") in PCB: PCB1 ("PCB1")
    static DLTypeInfo[] PCB1DEALERArray= {
        new DLTypeInfo("DLRNO", DLTypeInfo.CHAR, 1, 4, "DLRNO", DLTypeInfo.UNIQUE_KEY),
        new DLTypeInfo("DLR-NAME", DLTypeInfo.CHAR, 5, 30, "DLR-NAME"),
        new DLTypeInfo("CITY", DLTypeInfo.CHAR, 35, 10, "CITY"),
        new DLTypeInfo("ZIP", DLTypeInfo.CHAR, 45, 10, "ZIP"),
        new DLTypeInfo("PHONE", DLTypeInfo.CHAR, 55, 7, "PHONE")
    };
    static DLISegment PCB1DEALERSegment= new DLISegment
        ("DEALER", "DEALER", PCB1DEALERArray, 61);

    // The following describes Segment: MODEL ("MODEL") in PCB: PCB1 ("PCB1")
    static DLTypeInfo[] PCB1MODELArray= {
        new DLTypeInfo("MODTYPE", DLTypeInfo.CHAR, 1, 2, "MODTYPE"),
        new DLTypeInfo("MODKEY", DLTypeInfo.CHAR, 3, 24, "MODKEY", DLTypeInfo.UNIQUE_KEY),
        new DLTypeInfo("MAKE", DLTypeInfo.CHAR, 3, 10, "MAKE"),
        new DLTypeInfo("MODEL", DLTypeInfo.CHAR, 13, 10, "MODEL"),
        new DLTypeInfo("YEAR", DLTypeInfo.CHAR, 23, 4, "YEAR"),
        new DLTypeInfo("MSRP", DLTypeInfo.CHAR, 27, 5, "MSRP"),
        new DLTypeInfo("COUNT1", DLTypeInfo.CHAR, 32, 2, "COUNT")
    };
    static DLISegment PCB1MODELSegment= new DLISegment
        ("MODEL", "MODEL", PCB1MODELArray, 37);

    // The following describes Segment: ORDER ("ORDER1") in PCB: PCB1 ("PCB1")
    static DLTypeInfo[] PCB1ORDERArray= {
        new DLTypeInfo("ORDNBR", DLTypeInfo.CHAR, 1, 6, "ORDNBR", DLTypeInfo.UNIQUE_KEY),

```

DLIModel Utility – XML schema output

```
AUTPSB1-PCB1.xsd
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ims="http://www.ibm.com/ims"
  xmlns="http://www.ibm.com/ims/AUTPSB1/PCB1"
  targetNamespace="http://www.ibm.com/ims/AUTPSB1/PCB1"
  elementFormDefault="qualified">

  <xsd:annotation>
    <xsd:appinfo>
      <ims:DLI version="2.0" mode="retrieve" topLevelElement="PCB1" />
    </xsd:appinfo>
  </xsd:annotation>

  <!-- PCB1 -->
  <xsd:element name="PCB1">
    <xsd:annotation>
      <xsd:appinfo>
        <ims:pcb name="PCB1" alias="PCB1"/>
      </xsd:appinfo>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="DEALER" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <!-- DEALER -->
  <xsd:element name="DEALER">
    <xsd:annotation>
      <xsd:appinfo>
        <ims:segment name="DEALER" alias="DEALER"/>
      </xsd:appinfo>
    </xsd:annotation>
    <xsd:complexType>
```

DLIModel Utility – DAS command editor

```

AUTPSB1-PCB1.xsd *Dealer.config
<?xml version="1.0" encoding="UTF-8"?>
<Config name="Dealer" targetNamespace="http://my.DealerServices.com" xmlns="http://www

  <Command name="getSimpleOrder"
    kind="RETRIEVE"
    psb = "AUTPSB1"
    pcb = "DEALERDB"
    dbView ="autpsb1.AUTPSB1DatabaseView"
    dra="IMS1">
  <!-- Parameters -->
  <Parameter name="city" type="xsd:string"/>
  <
  <cc></cc>
  <Command name="" kind="" psb="" pcb="" dbView="" dra="">
  <Config name="" targetNamespace="" version="">
  <DLI occurrences=""> </DLI>
  <field name=""></field>
  <insert></insert>
  <Parameter name="" type="" />
  <param name="" />
  <qualifier field="" operator=""> </qualifier>
  <segment name=""></segment>
  <SSA name="" lockClass="" commandCode="">
  </Com
  <Comm
    psb = "DEALERDB"
    dbView ="autpsb1.AUTPSB1DatabaseView"
    dra="IMS1">
  <!-- Parameters -->
  <Parameter name="city" type="xsd:string"/>
  <Parameter name="model" type="xsd:string"/>
  <Parameter name="year" type="xsd:string"/>
  <Parameter name="lName" type="xsd:string"/>

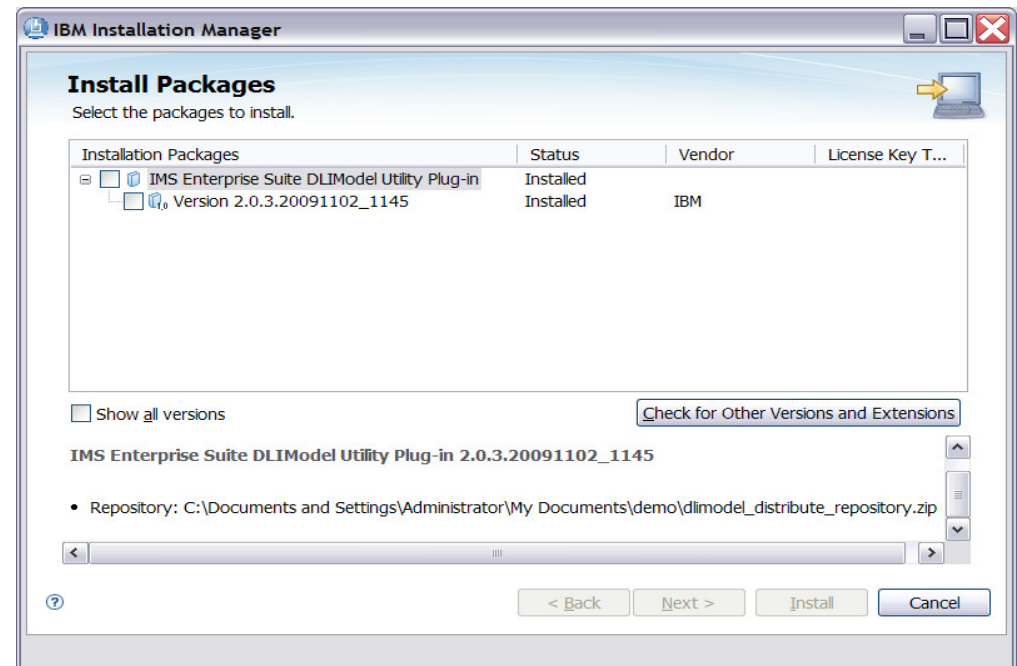
```


DLIModel Utility – WSDL output

```
Dealer.config Dealer.wsdl x
<?xml version="1.0" encoding="UTF-8"?><wsdl:definitions xmlns="http://my.DealerServices.com" xmlns:wsdl="http://
<!--Types-->
<wsdl:types>
<xsd:schema targetNamespace="http://my.DealerServices.com">
<import xmlns="http://www.w3.org/2001/XMLSchema" schemaLocation="AUTPSB1-PCB1.xsd" namespace="http://www.ibm.com
<xsd:element name="getSimpleOrder">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="city" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="getSimpleOrderResponse">
<xsd:complexType>
<xsd:sequence>
<element xmlns="http://www.w3.org/2001/XMLSchema" xmlns:pcb="http://www.ibm.com/ims/AUTPSB1/PCB1" ref="pcb:DEALE
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="getOrder">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="city" type="xsd:string"/>
<xsd:element name="model" type="xsd:string"/>
<xsd:element name="year" type="xsd:string"/>
<xsd:element name="lName" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="getOrderResponse">
<xsd:complexType>
<xsd:sequence>
<element xmlns="http://www.w3.org/2001/XMLSchema" xmlns:pcb="http://www.ibm.com/ims/AUTPSB1/PCB1" ref="pcb:ORDER
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="getOrderByDate">
<xsd:complexType>
```


DLIModel Utility – An Eclipse plug-in

- It is a GUI that comprises of Eclipse, Eclipse Modeling Framework (EMF) and Graphical Editor Framework (GEF).
- It can be installed as a stand-alone Eclipse application or on top of other Eclipse based products (i.e. RAD, RDz, Data Studio) using IBM Installation Manager.



How do I get it?

- IMS SOA Integration Suite - IMS Enterprise Suite
 - <http://www-01.ibm.com/software/data/ims/soa-integration-suite/enterprise-suite/dlimodel-utility>, or