Java Data Dictionary – Demystified, Part 1

Applies to:

SAP NW 04/04S, Composition Environment

Summary

Often, JAVA dictionary seems quite mystifying or feature-less to its various users.

The questions that often arise are how to leverage on the capabilities of a Dictionary project within and outside of Web Dynpro Java applications.

This document, in the first of many details answers to same basic questions when beginning to work on a dictionary project and help understand the usefulness of dictionary aspects.

Author(s): **Vidya Gugnani**

Company: **SAP Labs India Pvt Ltd.**

Created on: **27 June 2007**

Author Bio

Vidya Gugnani is working with the Web Dynpro Tools design team in SAP Labs India Private Ltd. Prior to this, was also working with ABAP Workbench team involved in development of test tools based on SAP NetWeaver.

Table of Contents

To begin with, following is a list of questions which should serve to solve some of the most basic and fundamental queries regarding java dictionary and what purpose it can serve you…

Java Dictionary FAQ's:

⇨ Why simple types and why structures? What purpose do these two serve differently?

Simple types are data types for which you can specify the type e.g. binary, boolean, integer etc and also specify additional properties for them as you might have noticed in the Simple Type Editor (e.g. length, decimals, labels, representation, enumeration etc).

- Within a Web Dynpro project, these are used to bind the context of UI elements. UI elements properties can be bound to Dictionary Simple Types or Core Data Types (in Web Dynpro implemented as Dictionary Types). For example the text property of FormattedTextView has to be bound to a Dictionary Simple Type FormattedText.

- Data type descriptions in the form of structures are used in Web applications and can be created in the Java Dictionary independently of the program code. Structures created in the Java Dictionary can be reused in different Web Dynpro contexts.

- Structure data type may use the simple types internally.
  You do not need to specify length,decimals,etc other such properties if you choose to use a simple type for your structure elements.
  Of course, you could specify all these individually .
  In addition, a structure has columns which consist of these individual types, either using simple types or specified explicitly.

⇨ I delete tables but they are still visible. Can they be re-created with the same name?

When you delete a table in the dictionary perspective,for all purposes this table is deleted.

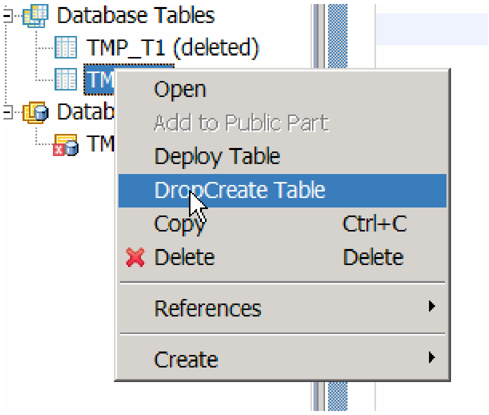
Tables in 04/04s are marked as deleted when you choose the option “Delete “ in design time.

The creation of Java dictionary projects is verified against the name server.

Hence, even after the deletion of the table, if the table is already transported and being used in another project, the corresponding design(xlf) files for the deleted tables are still retained .

e.g. imagine a situation where a table is already deployed and in use and then this is deleted locally and the deletion is not deployed, then this can cause corpses on the database. This is the reason creation of tables with the same name is then not allowed.

In CE and in versions greater than 04/04s,the “DropCreate” table option in the explorer gives you a more convenient mechanism of re-creating the table as it drops the table on the database and then re-creates it.



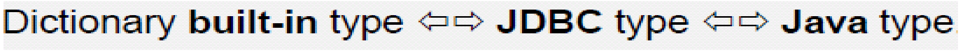
⇨ Why base types in simple types? Why not use java or JDBC types directly?

Simple types are application specific data types built upon predefined JDBC-like data types of the Java Dictionary.

The mapping is from dictionary built in types to JDBC types which in turn are mapped to Java types.

By creating a simple type and checking in, re-use is already enabled for other developers.

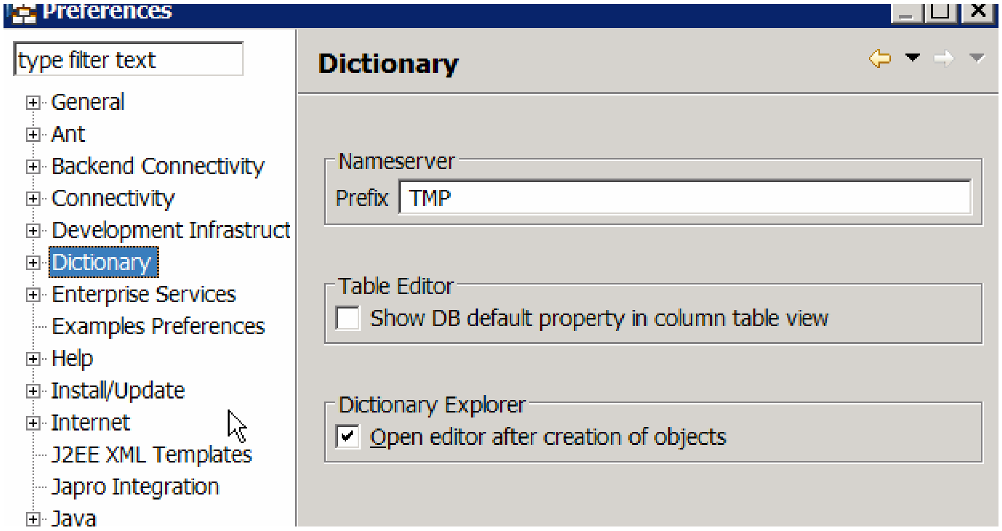
- However, using an existing simple type instead of creating a new one i.e. using a base type simple type A to create a simple type B:
- Enhances re-use
- Properties are inherited from original simple type which can then later be changed.
- N simple types can be created from a single base type.



Dictionary **built-in type ⇦⇨ JDBC type ⇦⇨ Java type**

⇨ My tables always seem to have a prefix...where can I change this?

The name reservation takes care of reserving tables/views on the name server; the prefix which is set in the preferences takes care of this.



Dictionary

Nameserver

Prefix TMP

Table Editor

☐ Show DB default property in column table view

Dictionary Explorer

☑ Open editor after creation of objects

- Go to Window->Preferences->Dictionary.
- Change the prefix e.g. from TMP to say com.xyz….

Tables/Views will then get created with this prefix.

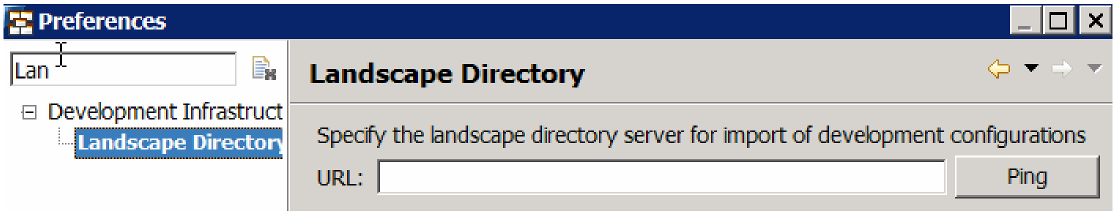
⇨ How can I remove the reservation created? Re-use the name of a deleted table?

If you really know that this table name is not used somewhere else, you can remove the name reservation.

- This has to be done in the SLD that acts as the Name Reservation Server (this is mostly the SLD that you also use for creating your SCs).
- Navigate to the section "Name Reservation" and have a look at the reserved names of type "DB table".
- Delete the desired entry.
- Restart the NWDS as this is caching the info from the name server.

  On the DB, there is no issue re-using the table. But you have to keep in mind that this will result in a drop of the existing table with all the data before it is recreated.

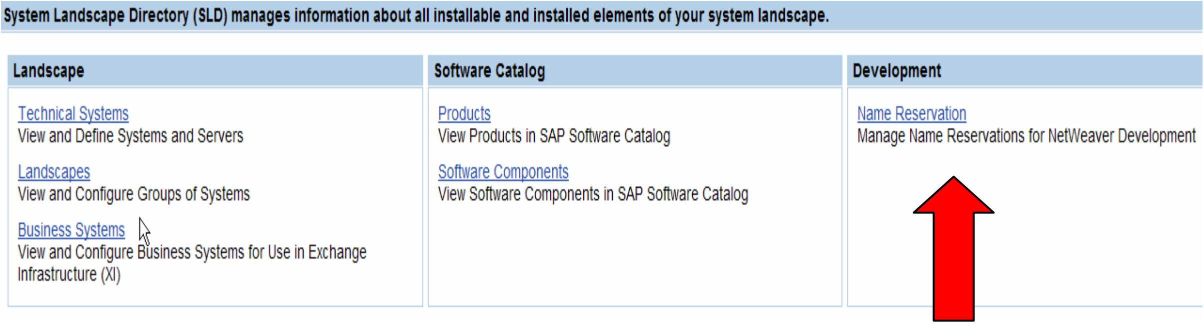
Where one can specify the SLD:



Landscape Directory

Specify the landscape directory server for import of development configurations

URL:



System Landscape Directory (SLD) manages information about all installable and installed elements of your system landscape.

Landscape	Software Catalog	Development
Technical Systems View and Define Systems and Servers	Products View Products in SAP Software Catalog	Name Reservation Manage Name Reservations for NetWeaver Development
Landscapes View and Configure Groups of Systems	Software Components View Software Components in SAP Software Catalog	
Business Systems View and Configure Business Systems for Use in Exchange Infrastructure (XI)		

⇨ What is the purpose of a local dictionary which is seen in Web Dynpro projects?

Data types defined in a Dictionary DC can be used as:

- Types for database columns
- Data Types for UI elements within Web Dynpro(used in context binding for
- Java views)

Dictionary familiar users can identify with the use-cases of:

- Developing application specific data types
- Web Dynpro developers can then check these out (assuming these are checked in to the Design time Repository (DTR) of NWDI. And reuse them in Web Dynpro DC's.

This is what the local dictionary in your Web Dynpro projects showcases.

Web Dynpro offers the ease of integrating the types that you create within the Web Dynpro project itself in your Web Dynpro application.

⇨ What are simple type format properties in the external representation section?

Formats are used to represent simple data types referring values in a special fashion.

This becomes especially relevant in the following scenarios:

- When data needs to be represented in a certain format using pre-defined conversion exits. Such a pre-defined one is the ABAP conversion exit *ALPHA* (and this currently is the only one) which is recognized by the java dictionary and the formatting of the special type is then done according to the conversion exit specified*. This is only for string data types*

  e.g. a string value 00023456 which would be converted to a format without the leading zeroes to some value such as 23456 for application logic which would then require to be converted back to the original value for screen output.

- The other requirement for this formatting comes from specifying numeric values such as decimals, float etc in a representation with comma, decimal separator etc. E.g. a number specified as 123734 may require to be displayed as 1, 23734 with the number formatting (for English locales).

Guidelines for Formats for numeric fields and how they need to be defined are specified here:

http://java.sun.com/j2se/1.3/docs/api/java/text/DecimalFormat.html

A user may require a date specified in dd/mmy/yyyy, dd/mm/yy or mm/dd/yy format,

The same is possible for Simple Types with builtInType time and timestamp, e.g.: h:m:ss for time. This should be defined as described in.

http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html

Some more examples for the formatting can be found under:

http://www.exampledepot.com/egs/java.text/FormatNum.html

If a format is chosen for a Simple Type and values entered in a Web Dynpro field referring this type, these values are checked against the format. An error message appears if the values do not satisfy the format.

⇨ Are foreign keys supported in java dictionary?

No, not as of now.

The Dictionary design time of the NetWeaver IDE doesn't support defining foreign key constraints.

If you want a portable application running on all databases supported by the SAP 6.40 J2EE engine, there are following alternatives for using foreign keys

- Retrieve a native vendor connection instead of an OpenSQL connection from the engine pool, and run DDL and DML on that, not interfering with standard SAP NetWeaver tables, e.g. avoid using the same table name prefixes.
- Use entity beans to access data from Java dictionary and in entity beans you can define relationships between these tables.
- Deploy java tables on the server then run the foreign key constraints script on the database explicitly.

⇨ Is there a character limit on the table length?

Yes, there is a limit on the maximum length for a table i.e. 18 which is inclusive of the minimum 2 required for the prefix.

⇨ Where all is the dictionary editor visible?

The Dictionary Editors are available in several kinds of projects:

- Dictionary DC'S
- Web Dynpro DC'S
- Local Dictionary and Web Dynpro projects

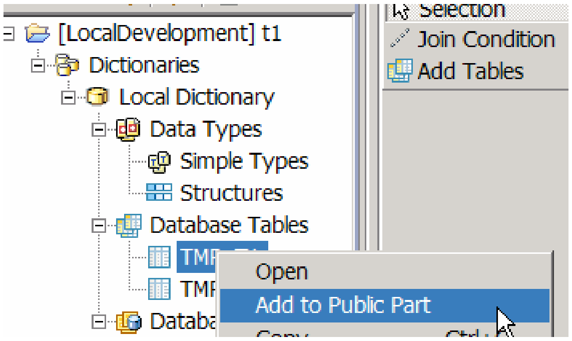
⇨ How can I use/re-use types declared in a different WebDynpro/Dictionary project?

The interface of a DC to the outside world is a public part.

- Create a public part for the dictionary project, the types of which you wish to expose.
- Add the types to the public part. Check in to DTR.
- Reference this in DC which wishes to use it.

Another DC project can then re-use the types defined in the public parts of this DC.

This will be visible then in the selection dialog for types in the second Wweb Dynpro application.



The usage of types in Web Dynpro programmatically is also well explained in:

https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/com.sap.km.cm.docs/library/webdynpro/WD%20Java/WD%20Tutorials/Value%20Help%20in%20Web%20Dynpro%20Applications.pdf

⇨ Is it possible to import table structures directly?

Table structure import into Java Dictionary is not possible.

⇨ How can I configure databases other than the system with Web Dynpro?

Accessing DB from Web Dynpro:

To work with databases other than system database:

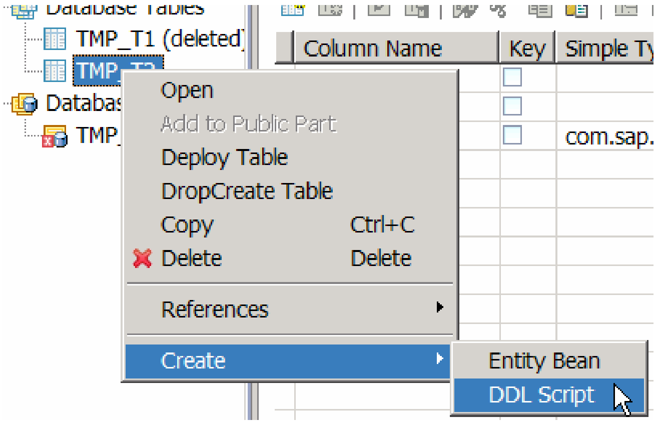
- Configure a server side data source for your DB (this can be done using the Visual Administrator tool - JDBC connector service). When a DB is created and used by Web AS, it is possible to create tables in that DB with dictionary project using the standard connection parameters specified in visual admin.
- Look up the appropriate data source up using JNDI interfaces in Web Dynpro.

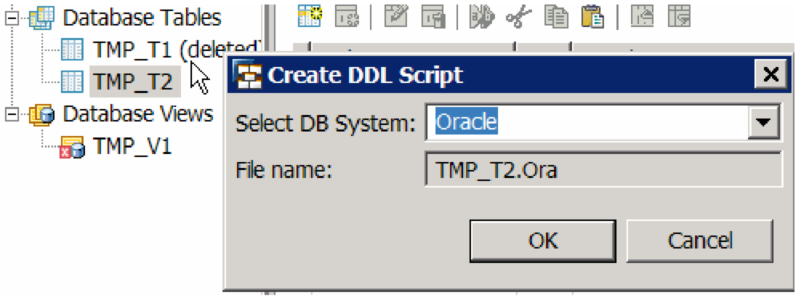
⇨ Is the default schema only used with Java Dictionary?

Yes, by default the SAP DB is used with the Java Dictionary. The DD is, until now designed for the default schema with 04/04s.

Tables and views from Java Dictionary project are deployed to the engine's Data Source, what means that the default engine's schema is supported.

One can generate the DDL per table for the different chosen databases or all of them.





Select DB System: Oracle

File name: TMP_T2.Ora

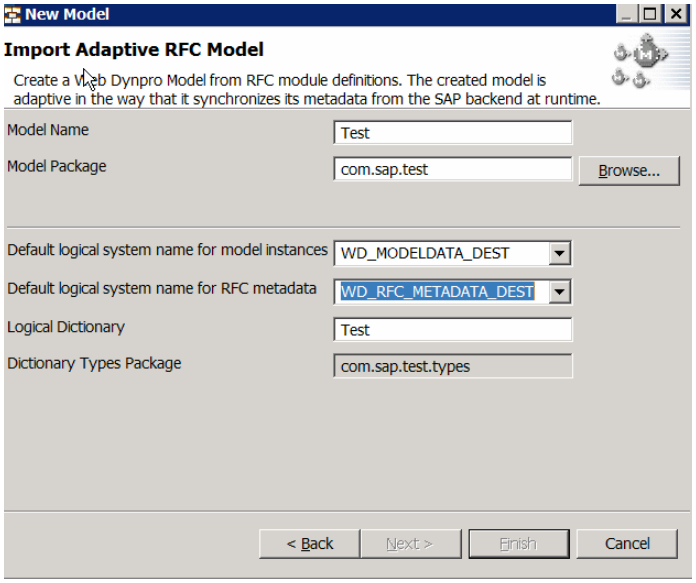
Besides providing all of the features with its type-set described above, Java Dictionary's purpose is also to create an abstraction of the database by providing a standard interface to create tables whatever the DBMS is.

With NW 04, Java Dictionary creates tables in the system database of the target SAP NW AS Java; regardless on the DBMS you use (MaxDB, Oracle, etc).

- A data dictionary project allows for an automatic deployment to any db using a SDM ANT task.
- There is not usage of proprietary db idioms or sequences and keeps the schema db independent of the database. Dictionary definitions are portable.
- Database tables can be defined “offline”. On deployment of the dictionary project, the framework cares for creating the database tables in the system database.

⇨ How can I import the data types created in ABAP in my Web Dynpro application?

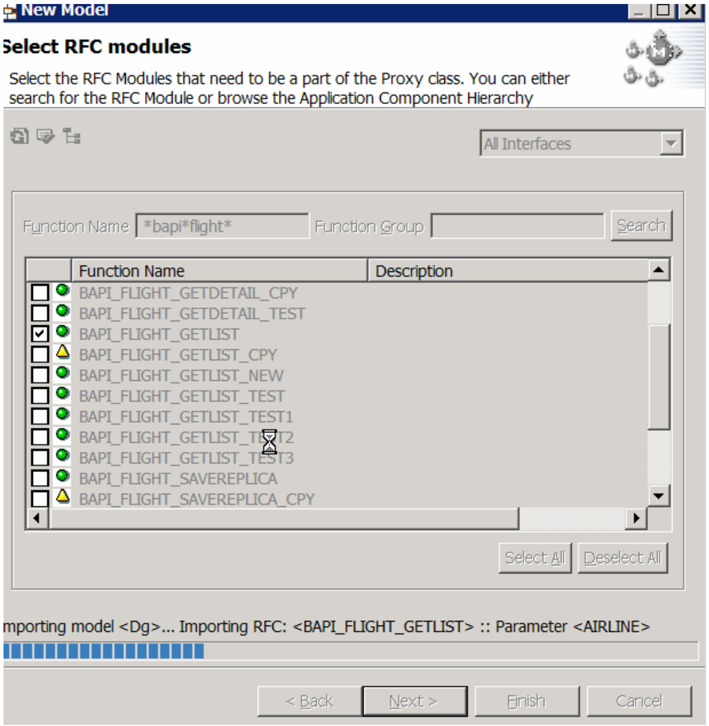
- You can import the dictionary data type that you create in ABAP Dictionary SE11 using the adaptive RFC model in a WebDynpro application in NW04S.
- Specify the model name, package, specify default logical names, and provide the R/3 system logon information and the function modules for importing the data.
- Once this is done, the R/3 data types get imported into the logical dictionary which is visible in your application nodes.



Import Adaptive RFC Model

Create a Web Dynpro Model from RFC module definitions. The created model is adaptive in the way that it synchronizes its metadata from the SAP backend at runtime.

Model Name	Test
Model Package	com.sap.test
Default logical system name for model instances	WD_MODELDATA_DEST
Default logical system name for RFC metadata	WD_RFC_METADATA_DEST
Logical Dictionary	Test
Dictionary Types Package	com.sap.test.types



Select RFC modules

Select the RFC Modules that need to be a part of the Proxy class. You can either search for the RFC Module or browse the Application Component Hierarchy

Function Name *bapi*flight* Function Group

	Function Name	Description
□ ●	BAPI_FLIGHT_GETDETAIL_CPY	
□ ●	BAPI_FLIGHT_GETDETAIL_TEST	
☑ ●	BAPI_FLIGHT_GETLIST	
□ △	BAPI_FLIGHT_GETLIST_CPY	
□ ●	BAPI_FLIGHT_GETLIST_NEW	
□ ●	BAPI_FLIGHT_GETLIST_TEST	
□ ●	BAPI_FLIGHT_GETLIST_TEST1	
□ ●	BAPI_FLIGHT_GETLIST_TEST2	
□ ●	BAPI_FLIGHT_GETLIST_TEST3	
□ ●	BAPI_FLIGHT_SAVEREPLICA	
□ △	BAPI_FLIGHT_SAVEREPLICA_CPY	

mporting model <Dg>... Importing RFC: <BAPI_FLIGHT_GETLIST> :: Parameter <AIRLINE>

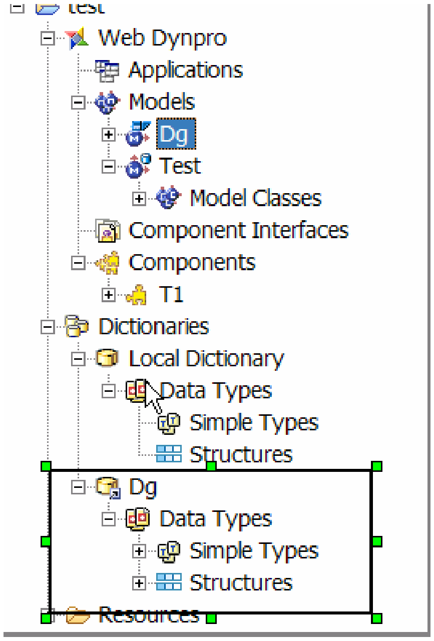
⇨ What is a logical dictionary?

When data types are imported using ARFC model in Web Dynpro, a logical dictionary gets created besides the existing dictionary in the Web Dynpro project.

Logical dictionaries are visible in the dictionary explorer under the Web Dynpro application node with the node name same as that of the logical dictionary (Simple type by default).

All the imported data types are then mapped to JAVA DDIC simple types.

Once the logical dictionaries are created, they are persisted as a Web Dynpro Model class and corresponding simple type design time files are generated for the imported types as shown below for logical dictionary TEST.



⇨ How can I update records in my database?

Dictonary project will helps you in creating tables in the backend and creating the same context structure in your WebDynpro application..

When you deploy the (.sda file) which has been created when you have "Create Archive" of the Dictionary project to the J2EE engine, then the tables are created in the database.

- Create a java bean where you can access the dictionary types and modify records.
- To add/write records, one would require to use their own JDBC controller classes for queries.
- One can alternatively use the EJB model also...in newer versions of NWDS.

⇨ Are there some good links for me to begin with Dictionary usage?

Following are some useful links:

[Web Dynpro Sample Applications and Tutorials]

[Reuse Java Dictionary Types in Web Dynpro DCs]

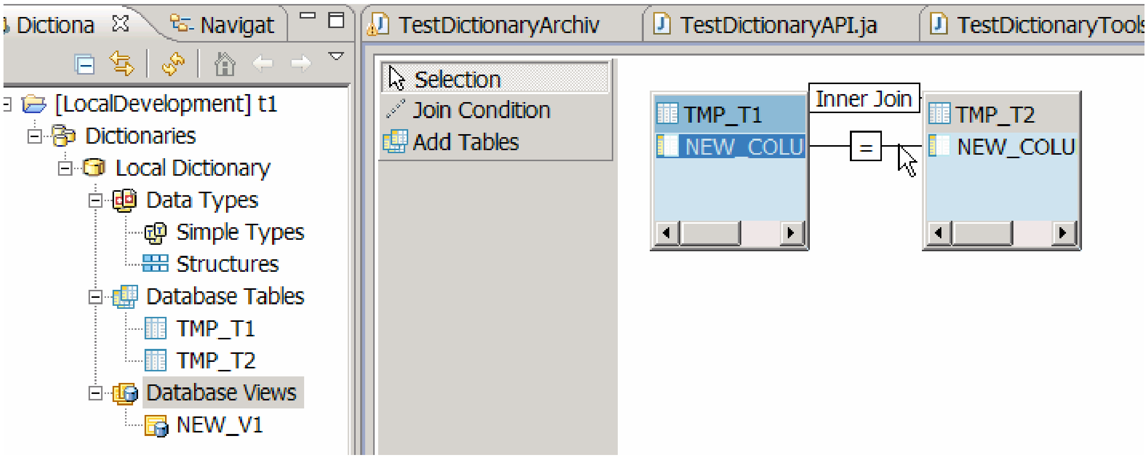
⇨ Is there support for database views in Dictionary?

With 04/04s unfortunately no there is no possibility in design perspective.

However, with later versions such as in the Composition Environment release (CE), similar to simple types and structures and database tables,

Views can also be created on tables and SQL statements generated as a result of the view can also be seen, a handy addition to the Dictionary project.

Please note this is not available in 04/04s but in the Composition Environment. (CE 7.0)



Related Content

- Simple Java Bean Generator for DB
- Reuse Java Dictionary Types in Web Dynpro DCs

Copyright

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.