

Hovitaga Report Generator

Free edition

User's manual

02.09.2011

INTRODUCTION	2
LIMITATIONS	3
USER INTERFACE	4
QUERYING THE DATABASE	8
GENERATING A REPORT.....	10
KEY MAPPING.....	11

Introduction

Hovitaga Report Generator is a powerful tool that helps SAP consultants, ABAP developers and basis administrators to work with the database of an SAP system. It provides an intuitive way to build ad-hoc reports and statistics with simple OpenSQL commands. No further ABAP programming is required.

Requirements and installation

Hovitaga Report Generator is entirely written in ABAP, so it is transparent (not a black-box development) and deeply integrated into the SAP system. No interfaces needed, no platform-dependency, no separate IT team to maintain. It does not expose the SAP system to any access from outside.

Installation is a process of few minutes, since it only consists of importing one transport with the TMS (Transport Management System). The only additional effort is to set up the authorizations for the users. Due to the intuitive user interface and extensive documentation no consulting or implementation project is needed.

Hovitaga Report Generator runs on SAP 4.6C but some features require SAP Netweaver 7.00 (aka. 2004s) or above.

Support and maintenance

We provide two levels of support. Standard support makes our customers eligible to receive regular support packages that contain all corrections and improvements. Customers who choose the premium support will receive every enhancement or correction immediately without having to wait for the new support package to be released. The software can be purchased without any support also, if required.

Limitations

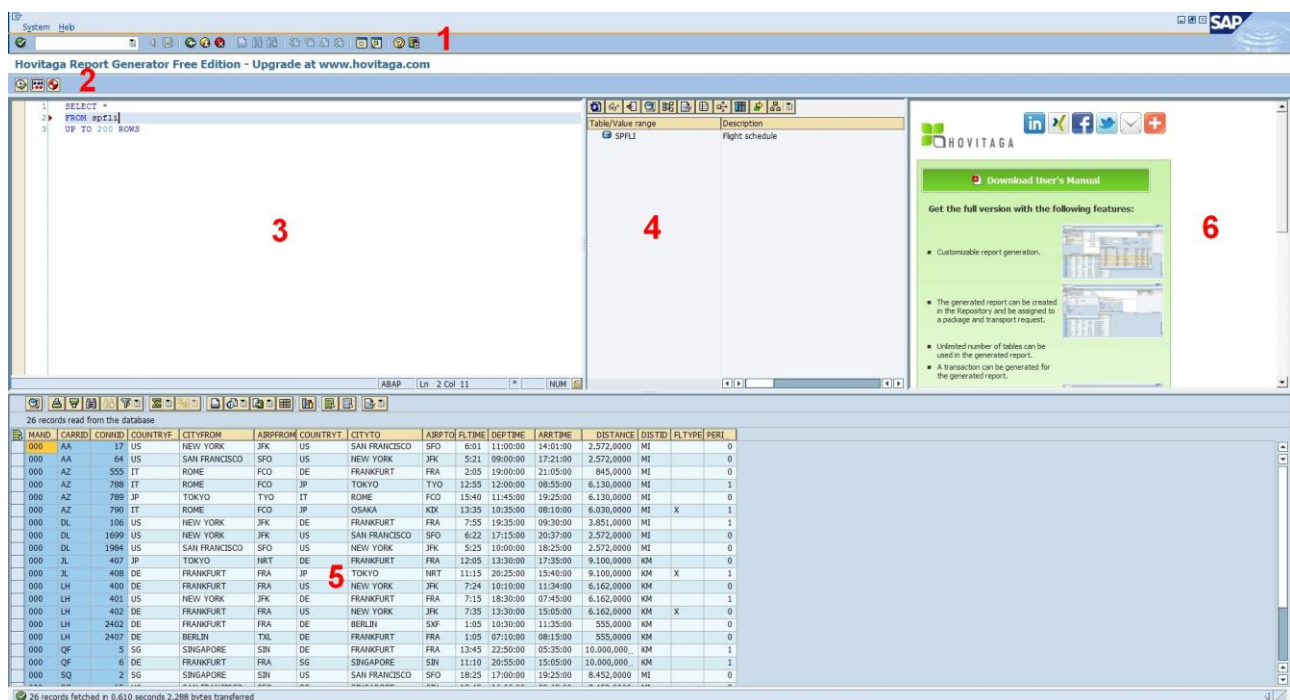
Here is a comparison chart showing what is missing from the free edition.

Feature	Report Generator	Report Generator
	Free version	Full version
Freely defined selections	✓	✓
Transport free	✓	✓
Inner joins	✓	✓
Outer joins	✓	✓
Subqueries	✓	✓
Group functions	✓	✓
"Select for all entries in" syntax	✗	✓
Background processing	✗	✓
Hierarchical display	✗	✓
Editing unlimited queries in one session	✗	✓
Saving queries to a repository	✗	✓
Record level authorizations	✗	✓
Field level authorizations	✗	✓
Report code generation	✓	✓
Selection screen customization	✓	✓
Assign the report to package/transport	✗	✓
Creation of a report in the Repository	✗	✓
Creation of a transaction for the report	✗	✓
Generation of event handler code	✗	✓
Generation of DDIC structure	✗	✓
ALV Tree display option for the report	✗	✓
Using multiple tables in the generated report	✗	✓
Many ALV options (class based, FM based etc.)	✗	✓

User Interface

The user interface consists of five major parts:

1. The Standard Toolbar
2. The Application Toolbar at the top
3. The Command Text Editor at the center
4. The Table List on the right side
5. The ALV List or Tree at the bottom that displays the Result Set
6. HTML displaying more information and news



Note: The borders that separate these screen areas are Splitters, so each screen area (excluding the HTML control) may be resized with the mouse.

The Standard Toolbar

There are three buttons to mention here:

- Back
- Exit
- Cancel

You can leave the transaction with pressing the Back, Exit or Cancel buttons.

The Application Toolbar

This screen area contains buttons for the following functions:

- performing actions on the selected command
 - o Execute
 - o Simulate
 - o Generate report

Performing actions on the select command

Simulation

Pressing the “Simulate” button will trigger the syntax check and parsing of the selected command. First some internal checks will be executed. If those were successful, then the standard ABAP syntax check is going to be executed on the command that is displayed.

If the command has any syntax errors, the appropriate error message will be displayed.

Note: the FOR UPDATE addition for a Select command is not supported.

Note: depending on what kind of syntax error the command has, the Table List may not be filled properly.

Execution

Pressing this button will execute the command. Execution contains an implicit parsing and checking of the command. If the command has syntax errors, the command is not executed.

The result is displayed by an ALV List by default. A green status message is displayed at the bottom of the screen that informs the user about the execution time, transferred bytes and the number of rows affected by the command.

Note: the display mode is also saved to the database for each command, so if a command was displayed with an ALV Tree for the last time, and the user restarts the transaction, the results of the command will be displayed with an ALV Tree, if executed again.

Execution

See chapter “Generating a report”.

Command Text Editor

In the Netveaver 7.00 (and above) version the Command Text Editor uses the standard ABAP Editor control and supports advanced code editing features like:

- Commenting and Uncommenting lines
- Saving to file/Loading from file
- Clipboard functionalities
- Undo/Restore function
- Find and Replace
- Pretty printer (using the user defined settings)
- ABAP documentation
- ABAP syntax diagram

These functions can be invoked with the context menu of the Command Text Editor. Many of these are accessible with the keyboard (clipboard functions, undo/restore, find, replace etc.).

Note: if the first character of a line is an asterisk (*), the whole line is considered as a comment. So for example if the valid command “SELECT * FROM spfli” is entered in such a way that the asterisk character is the first character of a line, a syntax error will occur.

Note: The “ symbol is not considered as a start of a comment!

Note: it is not necessary to close the OpenSQL commands with periods.

Table List

The Table List always contains all the tables used in the selected command. It is filled when the command is executed, or the “Recognize Tables” button is pressed on the toolbar of the Table List. If the command text is valid, the tables used in the command are listed in the Table List. A different icon is displayed next to the name of the table, depending on the type of the object:

- Transparent table
- View
- Clustered table
- Pooled table

After the list is filled, many operations can be carried out on the listed tables with dedicated buttons on the toolbar of the Table List:

- Display Table Definition (*no authorizations in the demo system*)
 - o Calls the appropriate transaction to display the Transparent Table/View/Pooled Table/Clustered Table
- Display Indexes
 - o Invokes the dialog for administering table indexes
- Display Technical Settings (*no authorizations in the demo system*)
 - o Shows technical settings of a transparent table
- Display Network Graphics (table relationships)
 - o Calls the graphical tool to display foreign key relationships between tables in the Data Dictionary
- Display Text Table (*no authorizations in the demo system*)
 - o Displays the definition of the Text table of a table (if exists)
- Display Runtime Object
 - o Calls the appropriate transaction to display the active runtime object of the selected table
- Where Used List (*no authorizations in the demo system*)
 - o Opens the Where Used List dialog
- Display Table Contents (*no authorizations in the demo system*)
 - o Calls up the generated selection screen of the table, and displays the table contents accordingly
- Define Value Ranges

- Dynamic Value Ranges can be defined for the selected table, and be used in the command text
- Linked Query Assistant
 - This feature helps to construct joins and subqueries automatically, by offering all table relations in a context menu (or in a popup window) and generating the join conditions automatically

Note: a table must be selected with the mouse before using these buttons.

The Linked Query Assistant can be also called up with the context menu of the Table List by right-clicking on a table in the list.

The Result List

The result of a query is instantly displayed after the execution of the command. The following features are instantly available with the ALV view:

- Sorting
- Filtering
- Detail view
- Find
- Totals/Subtotals, Average, Min and Max functions
- Print
- Export to Excel/HTML/etc...
- Display Graphic

The Data Dictionary transaction (SE11) is called when doubleclicking on a column of the ALV List to display the data element that the column uses.

Domain information is displayed as a tooltip when moving the mouse cursor to a column heading.

The format of column headings can be set with the “Display of column headings” button on the toolbar of the ALV Control:

- Field alias (for example in case of “CARRID as Carried Identifier” in the command text)
- original field name
- generated field name (for example generated for group functions like SUM, AVG etc...)
- short text (read from the Data Dictionary)

Querying the database

Select commands

To query the database, simply type the command text into the Command Text Editor. When it is finished, hit F8 or choose “Execute” from the Application Toolbar. The result will be displayed in an ALV list.

All syntax elements of the select command are permitted which do not incorporate the usage of local variables, except for the “for update” addition. For further information on OpenSQL statements, use the ABAP documentation or check the syntax diagram using the context menu of the Command Text Editor (above 4.6C only).

This implies that the INTO clause of the Select command must not be used. The program will automatically generate a suitable target for the command, and add the INTO clause accordingly.

Dynamic Value Ranges

To filter the records affected by an OpenSQL command, the user can define Dynamic Value Ranges for each table used in the command. It is possible to set a filter on every field for any table with the SAP standard “select-options” feature.

A prerequisite for this feature is that the command must be parsed (simulate or execute), so the table names must be in the Table List.

To define a Value Range for a table, select the table, and press the “Use Value Ranges” button on the toolbar of the Table List. A popup window will appear. On the right side, the table name is displayed in a tree control. All the fields are under the node of the table selected. Doubleclick on the name of the field to define a select-option for it. It will appear on the right side of the popup window, with the SAP standard controls next to it. The usage of these controls is exactly the same as if they were on a standard selection screen. Press “Save” to finish the creation of Value Ranges. Upon a second doubleclick on the field name, or selecting the “Delete Selected Objects” button, a field may be removed from the right side, and no Value Range will be created for that field.

When the Value Ranges are defined, they appear in the Table List as a subnode of the table that was selected. The next step is to include this in the command text. Every Value Range has a name generated and displayed in the first column of the Table List. This is the reference that must be used in the command text to use the Value Range.

To do so, the user can either write it in the command text manually, or doubleclick on the Value Range in the Table List. In this case, the selected Value Range will be added to the actual cursor position.

To delete a Value Range, open up the same popup window with the same button that is used to define a Value Range. Remove the select-options for the given field by doubleclicking on it or choose the “Delete Selected Objects” button for it, and press “Save”. The Value Range that was

removed will no longer appear in the Table List. If it was used in the command text, a syntax error will occur when executing the command, so it must be removed from the command text also.

Note: adding Value ranges to the command text does not automatically filter the results. The command must be re-executed.

Note: Value Ranges are not available for Clustered and Pooled tables.

Linked Query Assistant

The Linked Query Assistant is a tool to provide a user friendly way of constructing relations between tables within Select commands. Using this feature, table joins can be made without any typing.

The Linked Query Assistant can be called up with the context menu of the Table List or using the toolbar of the Table List. Select one table, and right-click on the selected table or press the button on the toolbar.

All table relations are read from the Data Dictionary, and are offered through the context menu popping up. The checktable relations are menu entries in the context menu, the foreign key relations can be displayed with a popup window (due to the possible large number of items to be displayed).

The first level of the popup menu is always fixed: the user has the option to create one of the following table relations:

- Inner join within the current command
- Left outer join within the current command
- Subquery within the current command

The second level contains the list of tables that are in a checktable relation to the selected table (as defined in the Data Dictionary) and a menu entry to read the foreign key relations also.

The checktable join conditions will be offered as submenu items for each table. Since multiple fields may refer to a table, there may be more than one choice. Due to the fact that composite foreign key relations may be defined in the Data Dictionary, composite menu entries may be offered. These are marked with an asterisk sign in the context menu.

Selecting a menu entry on this level will trigger the generation of the relation between the tables. If it is an inner/left outer join, the join condition will be generated and added to the current command's text. In case of a subquery, the subquery text will be inserted into the current command's text.

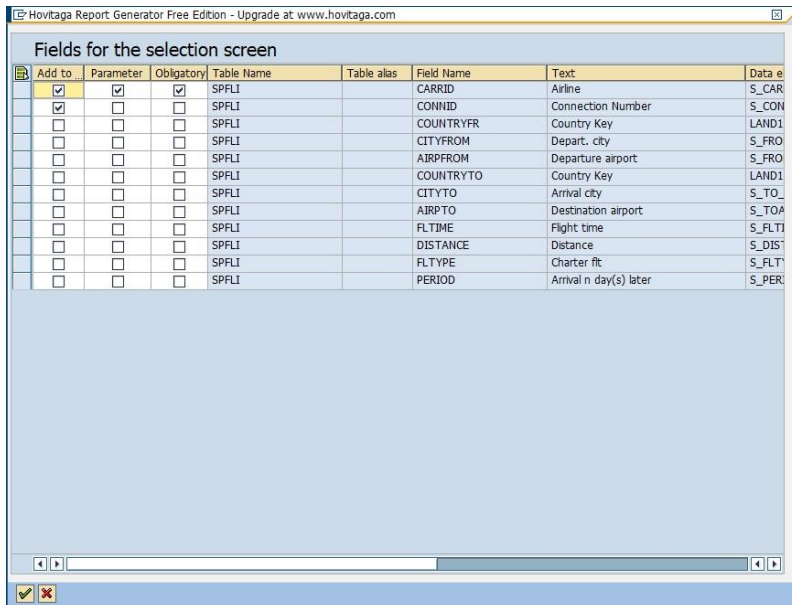
Note: the generated join condition or subquery will be inserted to the actual cursor position.

The foreign key join conditions will be offered in a popup window. Doubleclicking on a line will trigger the same actions as above (selecting a menu entry on the last level of the context menu).

Note: if the table relations are not maintained in the Data Dictionary, items will not appear in the context menu nor in the popup window.

Generating a report

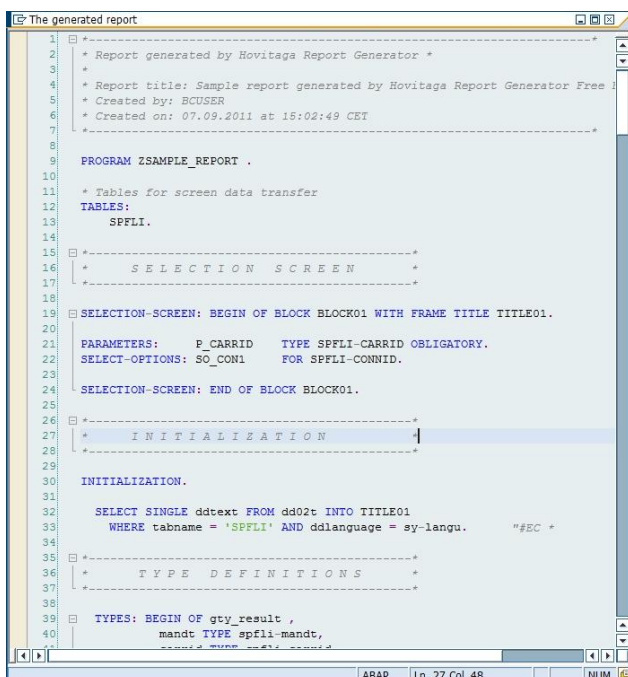
To create a report based on the select command, simply press “Generate report” or hit CTRL-F8.



Add to	Parameter	Obligatory	Table Name	Table alias	Field Name	Text	Data element
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SPFLI		CARRID	Airline	S_CARRID
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		CONNID	Connection Number	S_CONNID
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		COUNTRYFR	Country Key	LAND1
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		CITYFROM	Depart. city	S_CITYFROM
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		AIRPFROM	Departure airport	S_AIRPFROM
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		COUNTRYTO	Country Key	LAND1
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		CITYTO	Arrival city	S_CITYTO
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		AIRPTO	Destination airport	S_AIRPTO
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		FLTIME	Flight time	S_FLTIME
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		DISTANCE	Distance	S_DISTANCE
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		FLTYPE	Charter flight	S_FLTYPE
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SPFLI		PERIOD	Arrival n day(s) later	S_PERIOD

A popup window will appear all the fields from the tables that are used in the command will appear in an ALV grid. You can select the fields that will be included in the selection screen of the generated report. Checkboxes control whether the fields are obligatory or not and if they are simple parameters or complex select options.

The generated report will be displayed in a popup window. The report generation process is much more flexible in the full version (see limitations).



```

1  * Report generated by Hovitaga Report Generator *
2  *
3  * Report title: Sample report generated by Hovitaga Report Generator Free Edition *
4  * Created by: ECUSER *
5  * Created on: 07.09.2011 at 15:02:49 CET *
6  *-----*
7
8
9  PROGRAM ZSAMPLE_REPORT .
10
11  * Tables for screen data transfer
12  TABLES:
13    SPFLI.
14
15  *-----*
16  * SELECTION SCREEN *
17  *-----*
18
19  SELECTION-SCREEN: BEGIN OF BLOCK BLOCK01 WITH FRAME TITLE TITLE01.
20
21  PARAMETERS: P_CARRID TYPE SPFLI-CARRID OBLIGATORY.
22  SELECT-OPTIONS: SO_CON1 FOR SPFLI-CONNID.
23
24  SELECTION-SCREEN: END OF BLOCK BLOCK01.
25
26  *-----*
27  * INITIALIZATION *
28  *-----*
29
30  INITIALIZATION.
31
32  SELECT SINGLE ddtxt FROM dd02t INTO TITLE01
33    WHERE tabname = 'SPFLI' AND ddlanguage = sy-langu.    "SEC"
34
35  *-----*
36  * TYPE DEFINITIONS *
37  *-----*
38
39  TYPES: BEGIN OF gty_result ,
40    mandt TYPE spfli-mandt,
41    "mandt = Mandat des Daten"

```

Key mapping

Key	Function
F8	Execute
Shift-F8	Simulate
F9	Report generation

To learn more about Hovitaga Report Generator, visit www.hovitaga.com or send a mail to info@hovitaga.com. Detailed whitepapers and video demonstrations are available on our website.