

# How to...

## Aggregate Data in the Update Rules for Performance Improvement

BUSINESS INFORMATION WAREHOUSE



**Applicable Releases: It was tested with SAP BW 3.0B but it should be applicable to all SAP BW releases.**

**January 2004**

SAP (SAP America, Inc. and SAP AG) assumes no responsibility for errors or omissions in these materials.

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.

mySAP BI "How-To" papers are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using mySAP BI. Should you wish to receive additional information, clarification or support, please refer to SAP Professional Services (Consulting/Remote Consulting).

## 1 Business Scenario

You have an InfoCube that is designed for highly aggregated reporting. On its way to this InfoCube the source data is aggregated in the update rules because the source data is on a lower level of detail (granularity) than required in the InfoCube. For instance, the InfoCube requires data on a monthly level whereas the source data is on daily level. When loading this data a significant part of the loading time is spent in the update rules. You would like to improve the performance of the update rules.

## 2 Introduction

Aggregation in the above-described scenario is usually automatically done in the update rules if some of the characteristics like e.g. "calendar day" are not used in the InfoCube. However, this aggregation is done after the update rules have been processed as defined for each record. The idea of the performance improvement described in this paper is that the aggregation is done already in the start routine. Thus, the update rules that are processed for each record have to be processed for a smaller amount of records.

This approach should only be considered if the time spent in the update rules is significant and if the aggregation ratio of the number of incoming source records and the number of records stored in the InfoCube is high.

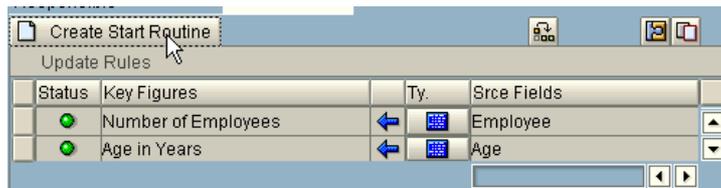
It is not possible to use the scenario if the order of aggregation and update rules can't be changed. For instance, the low level of detail is needed in the update rules themselves or calculations and transformations of data take place that are not commutative with aggregation. That's why this approach must be carefully analyzed and tested for each InfoCube before it is used.

### 3 The Step By Step Solution

A start routine for the update rules is created or changed and the coding for the aggregation is added.

#### 1. Create a Start Routine for the Update Rules

1. Create (or change an existing) start routine for the update rules of the InfoCube.



2. Do the following (or similar) declarations

```
*declarations
field-symbols: <DP_WA> like line of DATA_PACKAGE.
```

```
DATA: Z_DATA_PACKAGE type standard table of
DATA_PACKAGE_STRUCTURE with non-unique default key
initial size 0.
```

3. Add the following code. Replace the field names in the CLEAR statement with the fields that should be aggregated.

```
loop at DATA_PACKAGE assigning <DP_WA>.
*clear all the fields that are not contained in the
*InfoCube, in this example the InfoCube aggregates
*the data from the source over 0EMPLOYEE and
*OPERSON.
  clear: <DP_WA>-employee,
        <DP_WA>-person.

*Aggregate the data into the internal table
*Z_DATA_PACKAGE
  collect <DP_WA> into Z_DATA_PACKAGE.
endloop.

*Overwrite DATA_PACKAGE
DATA_PACKAGE[] = Z_DATA_PACKAGE[].

*empty the temporary table Z_DATA_PACKAGE
refresh Z_DATA_PACKAGE.
```

## 2. Testing and Monitoring

It is absolutely necessary to test whether query results are (still) correct after using these new update rules. In the following some "high level" checks are described that should be used in addition to query testing. For these tests it is assumed that the same data is loaded twice, with and without the enhancement in the start routine.

1. The request overview in the InfoCube management shows the number of transferred and added records. In our example request 176296 was loaded without aggregation in the start routine (in the default way), whereas request 177003 was loaded with the described enhancement in the start routine. Both requests have the same amount of transferred and added records, which is correct.

Requ...	R...	C...	C...	R...	QM...	Te...	Display ...	Transferre...	Added Records
177003								207652	96904
176296								207652	96904

2. In the detail view of the monitor you can compare the individual data packages of the two requests. In request 176296 data package 3 was reduced from 10000 (transferred) to 5033 (added) records.

3. The same view for request 177003 shows an additional line indicating that the same reduction has already been done in the start routine.

4. When you compare query results you can use the Request ID from the Data Package dimension as free characteristic and filter by a single request.

5. You can also display the Request ID in the query result and filter by both requests in one query navigation.

Structure			
Request ID	176296, 177003		
<b>Request ID</b>	<b>Age Range</b>	<b>Number of Actions</b>	<b>Headcount FTE</b>
176296	< 20	48	3,00
	20 - 29	808	191,00
	30 - 39	1.427	0,00
	40 - 49	527	0,00
	50 - 59	168	1,00
	60 - 69	238	0,00
	>= 70	3	5,00
	na	1.355	1,00
	<b>Result</b>	<b>4.574</b>	<b>2,00</b>
177003	< 20	48	3,00
	20 - 29	808	191,00
	30 - 39	1.427	0,00
	40 - 49	527	0,00
	50 - 59	168	1,00
	60 - 69	238	0,00
	>= 70	3	5,00
	na	1.355	1,00
	<b>Result</b>	<b>4.574</b>	<b>2,00</b>
<b>Overall Result</b>		<b>9.148</b>	<b>4,00</b>