

Integration of SAP TM and SAP EM – Optimizing the performance

How to optimize the performance of the data extraction for SAP EM in SAP TM (order tracking scenarios)

Author: Daniel Härder
Document Date: 25.02.2014

TABLE OF CONTENTS

SUMMARY	3
CHOOSING THE RIGHT DIRECT OUTPUT AGENT.....	3
OPTIMIZING THE DATA RETRIEVAL	3
Avoiding retrieval of data that is not needed.....	4
Retrieving additional data.....	4
OPTIMIZING THE RELEVANCE FUNCTIONS	4
Relevance Functions for Application Object Types	4
Relevance Functions for Event Types.....	5
BAdI method for central relevance check.....	5
OPTIMIZING THE FUNCTION MODULES FOR EXTRACTION.....	5
USING MULTIPLE QUEUES	6
IMPORTANT NOTES.....	6

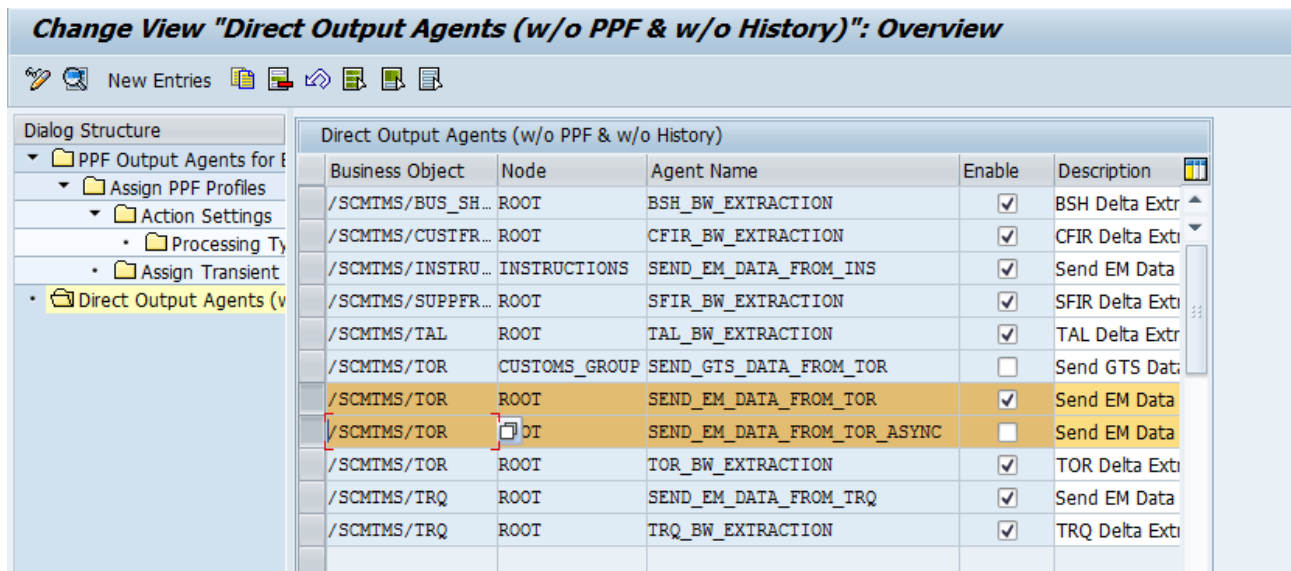
SUMMARY

Recently some new features have been introduced in the area of the data extraction for SAP EM in SAP TM. With these some new possibilities to optimize the performance are available (-> *Important Notes*). This document gives an overview of the several points which are relevant regarding the performance of the data extraction and gives some recommendations and best practices. It is highly recommended to follow the guidance given by this document.

CHOOSING THE RIGHT DIRECT OUTPUT AGENT

Two direct output agents are available that can be used to trigger the EM application interface. At all times, only one of the two shall be enabled. The direct output agents can be maintained in IMG activity *Maintain Output Management Adapter Settings* (path: SPRO -> Cross-Application Components -> Processes and Tools for Enterprise Applications -> Reusable Objects and Functions for BOPF Environment -> PPF Adapter for Output Management -> Maintain Output Management Adapter Settings). For more details regarding the usage of these two direct output agents you can check consulting note 1842397 and this document <http://scn.sap.com/docs/DOC-52226>.

If you experience long response times for dialog processes in SAP TM, in which the data extraction for SAP EM is involved, you should definitely check the note and the document. There you can find guidance on how either the direct output agent itself (including data retrieval and data extraction) or at least the data extraction can run in an asynchronous process to significantly reduce the response time.



OPTIMIZING THE DATA RETRIEVAL

You can check in IMG activity *Define Business Process Types* (path SPRO -> Integration with Other SAP Components -> Event Management Interface -> Define Application Interface), which data is already retrieved by default in the standard for business process type TMS_TOR (see also screenshot below). The “application tables” refer in this case to the associations from the TOR root node to related nodes (e.g. TOR_ITEM contains all item nodes that belong to a specific TOR instance, for which the extraction is triggered). This means that all this data is available by default and can be used in the different relevance and extraction function modules by accessing importing parameter *I_ALL_APPL_TABLES*.

Display View "Define Available Application Tables": Overview

Business Process Type: TMS_TOR

Structure/Table	DDIC Definition	DB Struct. Name	Bus. ...	Updte Fld Name	No C...	Inser...	Up
TOR_CAPA_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	<input type="checkbox"/>				
TOR_CAPA_ROOT_BEFORE	/SCMTMS/S_EM_BO_TOR_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	<input type="checkbox"/>				
TOR_CREATED_KEYS	/BOBF/S_FRW_KEY	/BOBF/S_FRW_KEY	<input type="checkbox"/>				
TOR_EXECUTION_INFO	/SCMTMS/S_EM_BO_TOR_EXEC_INF	/SCMTMS/S_EM_BO_TOR_EXEC_INF	<input type="checkbox"/>				
TOR_EXECUTION_INFO_BEF...	/SCMTMS/S_EM_BO_TOR_EXEC_INF	/SCMTMS/S_EM_BO_TOR_EXEC_INF	<input type="checkbox"/>				
TOR_ITEM	/SCMTMS/S_EM_BO_TOR_ITEM	/SCMTMS/S_EM_BO_TOR_ITEM	<input type="checkbox"/>	CHANGE_MODE		C	U
TOR_ITEM_BEFORE	/SCMTMS/S_EM_BO_TOR_ITEM	/SCMTMS/S_EM_BO_TOR_ITEM	<input type="checkbox"/>				
TOR_LOCATION_ADDR	/SCMTMS/S_EM_BO_LOC_ROOT	/SCMTMS/S_EM_BO_LOC_ROOT	<input type="checkbox"/>				
TOR_LOCATION_LOG	/SCMTMS/S_EM_BO_LOC_ROOT	/SCMTMS/S_EM_BO_LOC_ROOT	<input type="checkbox"/>				
TOR_PARTY	/SCMTMS/S_EM_BO_TOR_PARTY	/SCMTMS/S_EM_BO_TOR_PARTY	<input type="checkbox"/>				
TOR_REQ_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	<input type="checkbox"/>				
TOR_REQ_ROOT_BEFORE	/SCMTMS/S_EM_BO_TOR_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	<input type="checkbox"/>				
TOR_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	<input checked="" type="checkbox"/>	CHANGE_MODE			
TOR_ROOT_BEFORE	/SCMTMS/S_EM_BO_TOR_ROOT	/SCMTMS/S_EM_BO_TOR_ROOT	<input type="checkbox"/>				
TOR_STOP	/SCMTMS/S_EM_BO_TOR_STOP	/SCMTMS/S_EM_BO_TOR_STOP	<input type="checkbox"/>				
TOR_STOP_ADDR	/SCMTMS/S_EM_BO_LOC_ADDR	/SCMTMS/S_EM_BO_LOC_ADDR	<input type="checkbox"/>				
TOR_STOP_BEFORE	/SCMTMS/S_EM_BO_TOR_STOP	/SCMTMS/S_EM_BO_TOR_STOP	<input type="checkbox"/>				
TOR_STOP_SUCCESOR	/SCMTMS/S_EM_BO_TOR_STOP_SUCC	/SCMTMS/S_EM_BO_TOR_STOP_SUCC	<input type="checkbox"/>				
TOR_TENDERING	/SCMTMS/S_EM_BO_TOR_TEND	/SCMTMS/S_EM_BO_TOR_TEND	<input type="checkbox"/>				
TRQ_ROOT	/SCMTMS/S_EM_BO_TRQ_ROOT	/SCMTMS/S_EM_BO_TRQ_ROOT	<input type="checkbox"/>				
TRQ_ROOT_BEFORE	/SCMTMS/S_EM_BO_TRQ_ROOT	/SCMTMS/S_EM_BO_TRQ_ROOT	<input type="checkbox"/>				

Avoiding retrieval of data that is not needed

If not all this data is necessary for your scenario, method AVOID_RETRIEVAL_OF_APPL_TABLES of BAdI /SCMTMS/SEND_TOR_DATA can be implemented to avoid the retrieval of one or more of these "application tables". Example implementation /SCMTMS/AVOID_DATA_RETRIEVAL can be checked on how this works. By retrieving only the data that is needed and avoiding the unnecessary retrievals the overall performance can be enhanced.

Retrieving additional data

In the opposite case, if additional data is necessary for your scenario, method GET_ADDITIONAL_DATA of BAdI /SCMTMS/SEND_TOR_DATA can be implemented. By doing this, there is only one central retrieve needed for all the additional data. Multiple retrieves of the same data in the relevance or extraction function modules can be avoided. Example implementation /SCMTMS/GET_ADDITIONAL_DATA can be checked on how this works.

OPTIMIZING THE RELEVANCE FUNCTIONS

In general, relevance functions should be implemented carefully, since they are executed quite often. Therefore even little performance bugs can sum up to add significant processing time.

Relevance Functions for Application Object Types

In the standard relevance functions like ODT20_REL_TOUR (function module /SCMTMS/REL_AOT_ACT_TOR), it is only checked that the TOR object has one of the supported categories and that the application object type matches. This can lead to unnecessary updates of the corresponding event handlers, because the BAPI /SAPTRX/BAPI_EH_POST is called as soon as any data is changed in any of the nodes of a freight order/freight booking. This can be optimized by implementing custom relevance functions. The number of event handler updates can be reduced to the ones that are really necessary, if the relevance function checks more specifically, if an update of the corresponding event handler should be done.

These relevance functions can have a bad impact on the performance (especially if many application object types are active) since they are called for each application object type that is defined and active. Therefore no additional data shall be retrieved inside the relevance functions. BAdI method GET_ADDITIONAL_DATA

can be used to provide all necessary application data for the EM application interface (see chapter *Retrieving additional data* above).

Relevance Functions for Event Types

It is normally clearly defined under which circumstances an event shall be sent from TM to EM and this should be reflected in the corresponding relevance function. These relevance functions can have a bad impact on the performance since they are called for each TOR that has been changed. Therefore no additional data shall be retrieved inside the relevance functions. BAdI method GET_ADDITIONAL_DATA can be used to provide all necessary application data for the EM application interface (see chapter *Retrieving additional data* above).

BAdI method for central relevance check

Method PREVENT_EVENT_MSG_SENDING of BAdI /SCMTMS/SEND_TOR_DATA can be used to implement one central check for all available event types. Example implementation /SCMTMS/PREVENT_EVMSG_SENDING can be checked on how this can be done for the standard order tracking scenarios. If this central check fails, the single event type relevance functions will not be executed anymore. If you expect frequent changes to the TORs that do not result in sending an event message to SAP EM, this BAdI method may enhance the performance, especially in case of mass updates.

OPTIMIZING THE FUNCTION MODULES FOR EXTRACTION

Multiple retrieves of the same data should be avoided (as already described for the relevance functions) and additional data should be retrieved by implementing BAdI method GET_ADDITIONAL_DATA. All the “application tables” should be accessed with binary search or by using available table keys. The standard ones are sorted by the criteria listed in the following table (in TM 9.0 SP09 and TM 9.1 SP01 respectively with implemented note 1965595):

Application Table	Sorted by
TOR_CAPA_ROOT	TOR_ROOT_NODE and NODE_ID
TOR_CAPA_ROOT_BEFORE	TOR_ROOT_NODE and NODE_ID
TOR_CREATED_KEYS	---
TOR_EXECUTION_INFO	PARENT_NODE_ID and EVENT_CODE
TOR_EXECUTION_INFO_BEFORE	PARENT_NODE_ID and EVENT_CODE
TOR_ITEM	PARENT_NODE_ID and NODE_ID
TOR_ITEM_BEFORE	PARENT_NODE_ID and NODE_ID
TOR_LOCATION_ADDR	NODE_ID
TOR_LOCATION_LOG	NODE_ID
TOR_PARTY	NODE_ID
TOR_REQ_ROOT	TOR_ROOT_NODE and NODE_ID
TOR_REQ_ROOT_BEFORE	TOR_ROOT_NODE and NODE_ID
TOR_REQ_TU_ROOT	TOR_ROOT_NODE and NODE_ID
TOR_REQ_TU_ROOT_BEFORE	TOR_ROOT_NODE and NODE_ID
TOR_ROOT	TOR_ID
TOR_ROOT_BEFORE	TOR_ID
TOR_STOP	---
TOR_STOP_ADDR	NODE_ID
TOR_STOP_BEFORE	---
TOR_STOP_SUCCESSOR	---
TOR_TENDERING	NODE_ID
TRQ_ROOT	TOR_ROOT_NODE
TRQ_ROOT_BEFORE	TOR_ROOT_NODE

USING MULTIPLE QUEUES

The standard integration scenario uses a single queue (standard queue name EM_TMS_TOR) for all communications from TM to EM. By this it is guaranteed that data on EM side is always consistent, but using only one queue may also be a bottleneck, since all updates to EM are processed sequentially.

BAdI /SCMTMS/SEND_TOR_DATA offers the possibility to implement custom logic in method CALL_EVENT_MGR to establish the usage of multiple queues for the extraction of data to SAP EM. An example on how this can be done is available in example implementation /SCMTMS/SPLIT_TOR_DATA_BY_TYPE of the BAdI. For more details regarding this you can check this document: <http://scn.sap.com/docs/DOC-49085>.

IMPORTANT NOTES

- [1842397](#) - Different direct output agents available for data extraction
 - ➔ Consulting note regarding the available direct output agents
- [1935617](#) – Multiple Queue handling for TM-EM Integration
 - ➔ Note that introduces BAdI /SCMTMS/SEND_TOR_DATA
 - ➔ All follow-up notes shall be implemented as well (1937006, 1938440, 1956427, 1940106, 1942449)
- [1965595](#) – Enable data extraction for SAP EM with asynchronous direct output agent
 - ➔ Prerequisite note for asynchronous mode for direct output agent SEND_EM_DATA_FROM_TOR
 - ➔ Sorting of standard application tables as listed in chapter “Optimizing the function modules for extraction”
 - ➔ Refactoring of standard relevance and extraction function modules to enhance performance

© 2014 SAP AG. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

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.

