

# Master Data Governance for Material: Performance Tweaks for MDG6.1 and later releases



## Applies to:

Master Data Governance for Material Master Data (MDG-M) – MDG6.1 and higher

## Summary

This document deals with several performance tweaks you can use to accelerate the startup of MDG Create or Change Material scenario. As the measurements are taken from a dedicated and unique test environment please handle the advices in this document with the necessary care. The measurement figures and performance gains may not be transferable to 1 to 1 to your system.

**Author(s):** Jan Poeschel, Thomas Mayer

**Company:** SAP

**Created on:** March 2013

**Changed on:** December 2014

## Table of Contents

Motivation.....	3
Hardware Requirements and System Performance .....	4
Reference System.....	4
Native System or Virtual Machine .....	5
Application Configuration and UIBBs.....	6
State of Building Blocks and Lazy Load .....	7
Hierarchical UI .....	7
Governance Scope.....	8
Memory Consumption .....	9
Direct Call of OVP .....	10
Custom UIBBs .....	10
Implementation Hints.....	11
UI Price Tags.....	12
Performance .....	12
SAP Notes.....	12
Custom Code.....	12
Implementation Hints.....	13
Upgrade to MDG 7.0 or higher .....	13
Running an EhP6 UI in an MDG 6.1 System .....	13
Performance Tweaks Checklist.....	15
Related Content.....	16
Copyright.....	17

## Motivation

Master Data Governance for Material master data enables customer to create, change or delete master data records and put them under governance. For a controlled roll-out of material data in and across enterprises a governance process is needed to guarantee that the intended data content will be deposited in the right place. In this context, the governance of material usage in processes is as important for Master Data as governance targeted to avoid duplicate or incorrect material records.

Displaying major parts of the material master data requires more system resources than a simple report or transaction. Therefore we assume, in terms of performance, the major point of interest for customers would be the runtime consumption up to the point that the material master data is completely shown and editable on screen. In the case of material this means once the Material Overview Page (OVP) is fully loaded.

Taking this point into consideration, measurements were taken on an SAP performance system and analyses were performed on how the application scales and how much resources a single step consumes. Besides the official sizing guide for customer system landscapes, this guide focuses on the application itself and how it can be accelerated. The findings are based on the scenario "Create Material" whereas the findings can also be used for the Change Material scenario. Change Material consumes a similar but slightly reduced amount of system resources and should be less critical.

Due to the fact that Create/Change Material is not a sub-second application displaying the material master data, the development group identified some best practices for tweaking the application on the customer side. Please note that SAP delivers the complete data model and all possible UI building blocks. Customers adjust the standard application configurations tailoring them to their current business and needs.

**Note:** The description and key figures given in the following name the configuration BS\_MAT\_OVP\_03 delivered with MDG 6.1. Tips for subsequent releases are marked. Topics with no release dependency are valid for all releases.

All following remarks regarding performance improvement are also valid for the later releases of MDG. Nevertheless, the performance key figures of the new releases are better due to several performance improvements and a better support of lazy load.

## Hardware Requirements and System Performance

Before tweaking the Create or Change Material application in your system, please follow the Sizing Guide for MDG Material. It will guide you in regards to the hardware that should be used when running a Material governance scenario. Here an example for MDG 6.1: when you have less than 10 active users running 200 processes per hour, you would need approximately 5GB of memory. If you have more than 10 users creating 750 processes you would need about 9GB of memory. Summarized, the Sizing Guide outlines the hardware requirements of your MDG-M installation.

You can find the Sizing Guide here: [www.service.sap.com/sizing](http://www.service.sap.com/sizing)

## Reference System

Once you setup your system, you should have a look how the application scales on it. We recommend doing these tests first before you start to adapt and enhance MDG-M to your needs.

This guide provides you with some reference figures supporting you in the early stages of determining the correct hardware setup. When you absolutely don't meet the following reference figures, please adjust your system landscape before continuing. Otherwise, you won't reach the ideal and targeted performance.

The reference system for MDG 6.1 figures bases on a SUN operating system with 4 virtual Intel E7- 4807 CPUs (each 1.8GHZ) and a physical memory size of 32 GB. Ideally, the system should not be under load to prevent inaccuracy in measuring.

As mentioned before, the subsequent tweaks focus on starting the overview page (OVP) of the material master data. Usually, you start the initial page of Create/Change Material where you select a material ID and a change request type. We shall ignore this step as it is not performance critical and directly start the material master. Therefore, create a shortcut/favorite in your SAP Menu for the following application:

Web Dynpro Application: MDG\_BS\_MAT\_OVP  
 Application Configuration: BS\_MAT\_OVP\_03  
 Parameters: ACTION=CREATE, USMD\_CREQ\_TYPE=MAT01

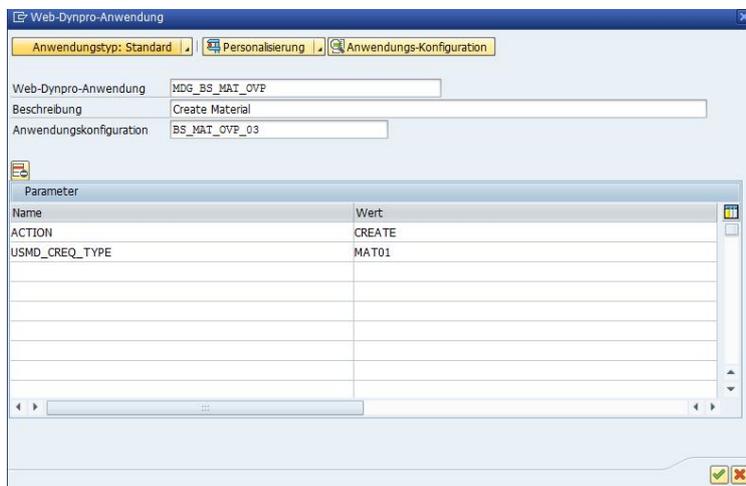


Figure: SAP Menu Favorite

We use the standard SAP application and configuration, which consists of the maximum of fields, covering the complete data model. There are no tweaks at all so far. This enables us to have a very close comparison to the standard runtimes.

**Note:** For this and all further comparisons, please follow the common guidelines for measuring performance. This means: Check the system status is ready for measurements: there should be no imports, not much user traffic, high system load, or any jobs scheduled. You can check common parts using transaction Z\_MEC. Please notice that this transaction may not be present in all releases. For preparation, execute runs in advance to fill buffers and avoid generating of programs or sub-routines (For example, after imports).

When single measurements of your complete measurement series differ in more than 15% your system is too unstable for further measurements. You should continue analyzing the reasons for this.

Taking measurements with transaction SAT does not consider the network traffic and browser rendering time. Be aware that the following reference figures take these two options into account. They were recorded using the SAP SUPA tool.

Please retrieve the statistical record for one of the measurements using transaction STAD. Here we have a look at the following figures:

- **Response Time:** Is the total response time without network and rendering efforts.
- **CPU Time:** Should be the biggest block in terms of response time.
- **DB Req. Time:** MDG-M has a very healthy relation to database runtime. The value here should be 10% or less in relation to the given CPU time.
- **Overhead:** The waiting time and roll processes are defined here as “overhead”. You can calculate them by subtracting CPU and DB time from Response Time. The relation should be 5-10% maximum.

If you not meet these figures, a first starting point for your analysis is the STAD result. Here you can easily see what direction your investigation should take: Is the DB time too high or too un-steady? Are your CPU capacities too low? Watch out for extreme deviations. One exception here: the lower response times you will get, the KPIs scale more differently – like e.g. CPU scales at 80% and the DB time at 17%.

You also gain higher response times when you have already enhanced the underlying data model MM with your own entities. The values here are based on the standard data model of MDG6.1 and therefore we recommend checking your application before any enhancement.

Please note that a STAD record often consists of more than one entry, as the application and/or backend triggers round trips. When you refresh your application using F5 you can find out the range of the statistical record sets. Don't forget to sum up the figures.

**Note:** MDG Material scales very well with available CPU power. This means that free processes positively influence the response time and also the single core power. In our reference measurements a slightly higher core frequency (1.8 to 2.1 GHZ) and a surplus of available cores decrease the response time.

## Native System or Virtual Machine

Master Data Governance is a highly customizable and very flexible application that puts material objects under governance. These functions are demanding on hardware and so we recommend identifying your needs (For example, the number of users and processes, duration of governance process etc.) and setting up your hardware accordingly at an early stage of your project.

Here we would like to share observations made from dealing with MDG performance over the last few years. MDG-M performs best when installed completely on native hardware. This would be our first recommendation. When using a virtual machine, it is of advantage when the application and database server run on the same instance. Try to avoid mixing up the approaches – this means a virtual machine application server connected to a native database server.

On OS level we observed the best performance on Linux or Power 8. If the latest IBM Power version is not available, we observed that later releases performed better.

As MDG is CPU speed demanding, we recommend reaching a high level of SAPS on your system.

## Application Configuration and UIBBs

Once your system is setup accordingly and the application scales similar to the reference measurement you can continue focusing on the application configuration. As mentioned before, SAP delivers the maximum configuration covering all model entities. However, not all scenarios require all material master data.

A first approach would be that the customer defines possible scenarios for involved roles and business users. One role is able to maintain other master data details than another business role (For example, different specialist levels). Therefore it would be an option to copy the standard configuration and divide it up into configurations tailored for the various roles. The advantage here would be removing unused UI building blocks (UIBBs) that may take up processing time unnecessarily.

See the available building blocks below and how they may scale. All UIBBs are collapsed and lazy load is introduced:

UI Building Blocks	Comments
(Initial Load) Change Request Material Basic Data	Nearly half of the total runtime is consumed by initial load, change request and basic data. This cannot be cut.
Plants	The second largest block is represented by the material plant level – 12 UIBBs in total.
Distribution Chain	The distribution chain is spread over 6 UIBBs and consumes the third largest block of total runtime.
Dimensions, Internal Comment, Basic Text	This level contains three building blocks consisting of 7 UIBBs in total and has medium impact on runtime.
Valuation	The valuation level is spread over 4 UIBBs and has medium impact on runtime.
GTIN, Descriptions	Those are represented by 2 UIBBs and have minimal impact on runtime.
Quality Inspection	Can be neglected in terms of runtime.
MARA Sales, MARA Purchasing	Those are represented by 2 UIBBs and have minimal impact on runtime.
Classification	Can be neglected in terms of runtime.

Table: UI Building Block Runtime Consumption

The initial load of the material master with the required UIBBs for change request and material basic data consumes approximately half of your total runtime. This part cannot be cut. You can try to accelerate this part by upgrading your hardware.

The following four blocks are of interest. Even if you decide not to remove complete blocks but only several UIBBs out of them you may gain a noticeable improvement in your total runtime. Analyze your requirement for of the mentioned topic blocks, or even of parts of the blocks, and remove unused blocks/UIBBs to save runtime.

**Note:** You can use the SAP standard configuration BS\_MAT\_OVP as a template for your own configurations when you use the deep-copy function. Don't include in this deep copy the section CR\_SECTION. Make sure you have the configurations copied for OVP, CBA and Layout. Afterwards, remove unused UIBBs and their wiring information, too. **Failing to remove the wiring information can result in unnecessary resource consumption.**

In MDG 7.0 and later use the configuration BS\_MAT\_OVP\_04 or the most recent one as a template instead. These configurations are adapted to support the FPM parameter FPM\_IGNORE\_WIRE\_SOURCE (See SAP Note 1943327 to improve the performance of older configurations in the same way).

## State of Building Blocks and Lazy Load

Our reference measurement refers to the maximum configuration with all UIBBs collapsed and lazy-load active. SAP delivers the standard configuration with this setup; please see the corresponding WD configuration MDG\_BS\_MAT\_OVP\_03 for. Even if the maximum configuration is available, the state and the lazy-load prevent from loading the complete UIBBs. They are fully loaded when expanded.

This state initially saves a lot of runtime. Therefore, decide which UIBBs you really need to load in advance, and which UIBBs can be loaded on user command. Every UIBB that is collapsed saves a certain amount of runtime. In general the following rule applies for UIBBs:

- When the UIBB is not assigned to the configuration, the corresponding feeder is not run.
- When the UIBB is assigned but collapsed, the feeder creates an instance for a later use.
- When the UIBB is expanded, the feeder runs and the UIBB take part at the phase model. Subroutines that consume a lot of CPU resources at runtime such as GET\_DATA or FLUSH are processed. The maximum possible runtime for an initial load is consumed.

**Note:** Changing the state of the UIBBs from “expanded” to “collapsed” saves end-users runtime but also up to one third (and more) of application server CPU time for a medium or full-blown standard UI.

## Hierarchical UI

Besides the standard process-oriented (flat) layout which SAP delivers there is a second configuration available containing the same UIBBs but ordered in a hierarchical way. The UI ID is MDG\_BS\_MAT\_OVP\_03H. The configuration is used by default in the Change Material scenario and shows only the minimal Basic Data and UIBBs for organizational data on the first screen. Further UIBBs and details are accessible using the “Details” button. Thus, the instantiation of UIBBs can be further postponed – similar to the collapsed state.

If this layout suits to your needs, you can also use it for the Create Material scenario. The hierarchical UI can save additional runtime and resources compared to the flat UI. It also saves up to 12 MB of memory consumption for each user process and startup. Database accesses and traffic and rendering time are not influenced in a noticeable way.

**Note:** You can deep-copy the standard hierarchical configuration BS\_MAT\_OVP\_03H and adjust it further to your needs. Don't include the section CR\_SECTION in this deep copy. Removing unused UIBBs and setting the state to “collapsed” with “lazy load” (setting on FPM application configuration level) may multiply the amount of saved runtime. This differs from case to case. By using the proposed layout you may already save additional runtime and 10+MB of memory for each process.

In MDG 7.0 and later releases use the configuration BS\_MAT\_OVP\_04H or the most recent one as a template instead. These configurations are adapted to support the FPM parameter FPM\_IGNORE\_WIRE\_SOURCE (See SAP Note 1943327 to improve the performance of older configurations in the same way.).

## Governance Scope

In MDG 6.1 and later there is the possibility to define the “Governance Scope”. This scope defines if entities and fields are put under governance or not. If entities are not governed their fields are disabled on the UI and in the staging area.

SAP delivers a special BC set which reduces the scope from MDG 6.1 to EhP6 for the data model MM. When doing please verify that the used UI configuration shows the EhP6 layout as well. Otherwise you may lose performance advantages.

**Note:** Using the Governance Scope for single entities does not have a huge impact on your performance. Reducing the scope in a dimension like the EhP6 scope may save 4 MB of memory.

## Memory Consumption

The question for memory consumption interacts highly with the configuration you have chosen and the states of the single UIBBs. Beginning at the smallest possible configuration, the minimum memory consumption in MDG 6.1 SP2 is 50 MB for one single process. Here only the UIBBs “Change Request” and “Basic Data” are present on the screen; it does not matter if they are expanded or collapsed.

If you use the full standard UI configuration, 128 MB of memory is consumed per start-up. When you now collapse all UIBBs (except the Change Request UIBB) you can easily save up to 40 MB for each start-up.

The following chart shows you how much memory consumption you may save at start-up

- ... when using collapsed UIBBs as default
- ... when removing unused UIBBs

The levels describe the existence of several logical blocks on the UI. The difference to the previous level describes the consumption of a logical block – like, for example, 8 MB for Valuation UIBBs in the expanded state.

- Basic: All
- Level 9: Removed Valuation UIBBs
- Level 8: Removed Plants UIBBs
- Level 7: Removed Distribution Chain UIBBs
- Level 6: Removed Quality Inspection
- Level 5: Removed MARA Sales and Purchasing
- Level 2: Change Request and Material Basic Data only

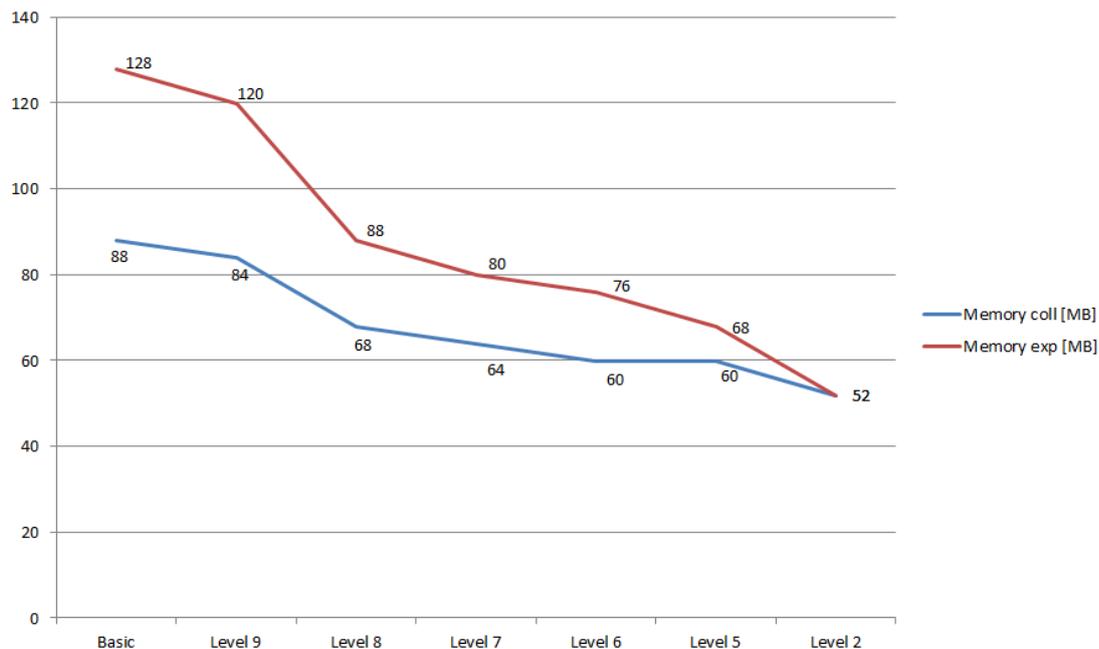


Figure: Memory Consumption expanded vs. collapsed in MDG 6.1

**Note:** Removing unused UIBBs saves you up to 32 MB of memory consumption like in the case for all Plant UIBBs. Using the UIBB state “collapsed” instead of “expanded” saves you additional memory. Multiply the saved amount of memory per process with your users and processes and you could adjust your original sizing plans achieving a lower TCO environment.

**These figures highly depend on the model you are using and the release you have installed. In this example the standard SAP delivered data model at MDG 6.1 is used without any enhancements.**

## Direct Call of OVP

In some business cases it may be useful to start the Material Governance from the Initial Screen – as it is proposed in the corresponding PFCG roles. In the Initial Screen it is possible to select material templates, determine the change request type or predefine organizational data values.

However, referring to certain configurations the material master governance always starts with a certain change request type. When there is no need for template data and the material number to be retrieved internally (can be changed in the master data screen anyway), then you can also start the material master overview page directly without processing the Initial Screen. This saves you time, getting directly to the master data.

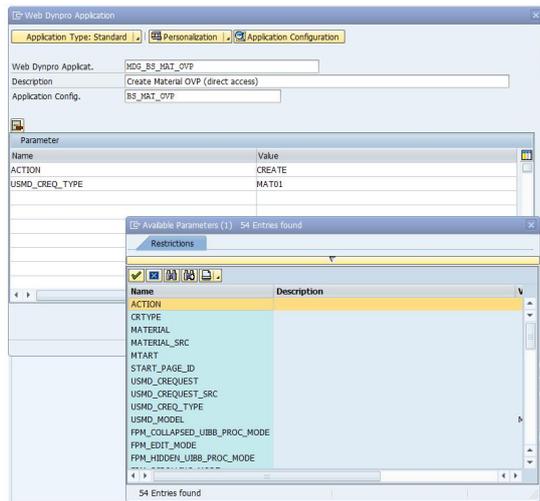


Figure: Parameters of Application MDG\_BS\_MAT\_OVP

To do this, create a new node in your user roles (Transaction PFCG) and add the material master data governance UI (see also our performance example):

**WD Application:** MDG\_BS\_MAT\_OVP

**WD Configuration:** BS\_MAT\_OVP  
(or custom specific)

**Parameters<sup>1</sup>:** ACTION  
USMD\_CREQ\_TYPE  
CRTYPE

## Custom UIBBs

Most often the standard material master data model is enhanced by customer-specific fields. Besides the model extension, UI configurations and programming are needed to bring the fields to the screen. In this case we would like to share some information that may be useful before setting up screen components. Different UI controls consume different amounts of system resources.

Please check that SAP Note 1756538 is applied to your system if you are below MDG6.1 SP03.

**Note:** For performance reasons it is not recommended to implement with MDG-M a copy with the same layout as known from the SAPGUI transactions MM01, MM02, and MM03. It is not the intention of MDG-M to replace these transactions.

Besides the quality of the hardware the performance of the governance UI is mainly influenced by the number of open UIBBs (see [State of Building Blocks and Lazy Load](#)). This also includes UIBBs used to organize the layout of the UI (Composite UIBBs).

Therefore, any enhancement of the UI creating a more complex UI will decrease the performance.

<sup>1</sup> The list of parameters can differ from this example. It depends on the scenario you want to support and if you are using e.g. Context Based Adaptation (CBA).

## Implementation Hints

- Execute a performance test with the newest standard UI configuration. Test and document several scenarios like creating a material, display and changing a material, and processing a change request.

These key figures give you the impression of the performance of the standard implementation of MDG in your system and allow comparisons with other systems. In addition you can check if the sizing of your system fits.

- Activate the buffering of the table FPM\_ADAPT\_SET (Transaction SE11 → Technical Settings). See also SAP Note 1860705.
- If you observe problems with the performance of the UI you can disable the highlighting of the changes in MDG.
- Keep the number of UIBBs on the governance UI as small as possible.

The UIBBs are independent building blocks of the UI. They retrieve data independently from each other. Each UIBB has its own feeder class instance, which retrieves the data from the Service Provider Interface (SPI). These SPI class instances access each the central MDG framework, which buffers the change request data and the governed business object (In this case the material BO). Reducing the number of UIBBs also reduces the number of class instances with decreases the memory and run time consumption.

As an alternative show the UIBB (the assignment blocks) collapsed first to take advantage of lazy load.

You can also move the UIBBs to subpages or tab strips to display details. All three solutions are equal from a performance point of view.

- Avoid a deep nesting of UIBBs!

Sometimes composite UIBBs are used as a kind of layout manager. Try to use the capacity of the composite UIBBs to the maximum. A single composite UIBB with two columns can contain up to six standard UIBBs.

Often the composite UIBBs are used as layout manager. Try to keep the number of them as small as possible and use their capacity (composite UIBBs with two columns can contain up to 6 UIBBs).

To monitor the complexity of your UI configuration you can use the FPM Hierarchy Browser of the FPM Workbench (transaction FPM\_WB).

- Display the data only once. Don't show the same data (For example, MARA-MEINS) in several UIBBs.

As mentioned before, any UIBB needs a separate retrieval of data from the central MDG framework. In case you need some values in the context of several UIBBs, try to display this data in a central UIBB on top of the UI.

- Don't mix configuration mode, customizing mode, and Context Based Adaptations (CBA) throughout the development of your enhancement.

If you need only smaller adaptations of the standard UI, you can use CBA to hide or show UIBBs or to change the layout of some UIBBs.

In case of larger changes you can use the customizing mode to adapt the UI configuration on layout level (For example, BS\_MAT\_OVP\_LAYOUT\_05). For smaller additional adaptations of the UIBBs (For example, dependent on the change request type or material type hide/show UIBBs or fields) you can use CBA.

To facilitate the later maintenance we don't recommend the combination of CBA on layout level and customizing on UIBB level.

If you want to implement your own overview page or use only a minimal number of UIBBs (~3) we recommend creating your own configuration (instead of hiding them all using CBA). To simplify the implementation you can copy a standard configuration and then remove all UIBBs you don't need. If you also want to adapt the UIBBs you can use the deep copy functionality to create also copies of the embedded UIBBs.

Don't create a deep copy of the section CR\_SECTION and its UIBBs. Here we recommend using CBA only. You also can add your own UIBBs with the above mentioned side effects to the performance.

## UI Price Tags

Here some “UI Pricing Trends” that may support your decision when designing own UIBBs. One UI element is measured here with reference to the consumed Browser CPU Time (Rendering). The following values are taken on a dedicated test environment – nevertheless you can use them to get a feeling what certain elements may cost and how they scale on client browser.

The runtime consumption is described here with very low, low (< 1 msec), and high (up to 35 msec).

UI Element	Runtime Consumption	Browser Used
Button	Very low	Inter Explorer 8 and 9
Input Field	Very low	Inter Explorer 9
Drop Down by Key	Low	Inter Explorer 9
Panel	Low	Inter Explorer 9
Group	Low	Inter Explorer 9
Tab per Tab strip	Low	Firefox 10
Calendar	Medium	Inter Explorer 9
ALV	High	Inter Explorer 8
Table	High	Inter Explorer 9

Table: UI Elements and their Price Tags (Source: SAP Central Performance Team)

The following browsers were in focus: Firefox 3.6, 4.0, and 10.0.1 – plus Internet Explorer 7, 8, and 9. Generally, you can say that the browser scales and performs better at the latest release. The lowest client CPU consumption (and the best performance in total) you will most probably get with the IE9 (state 2009).

## Performance

- Install the latest available and possible kernel or at least the latest available kernel patch.
- Install the latest available support packages for NetWeaver (including FPM) and MDG. If not possible implement at least all the performance SAP Notes mentioned below.

## SAP Notes

In order to assure having applied all released performance SAP notes, please update to the latest release and support package available.

If you cannot upgrade to the latest SP, please consider SAP Note 2105467. It lists several SAP Notes implementing performance improvements in FPM and SPI. It also contains a description how to find MDG relevant performance notes.

## Custom Code

There are several points for customer to extend the existing code base. Please follow the standard ABAP programming guidelines and have a strong focus on performance efficient programming. You can find more information here: [www.service.sap.com/performance](http://www.service.sap.com/performance).

Nevertheless, there are some MDG specific hints we can provide to you. When creating your own UIBB feeder or when using the cross-entity BAdI USMD\_RULE\_SERVICE\_CROSS\_ET in MDG assure to access any data model related data using the MDG AF provided Convenience API – direct accesses are not buffered.

In your feeder implementations you should raise an FPM event only one time for each round-trip (if possible) in order to avoid multiple runs of this event. Check the FPM event queue for this.

Within your Service Provider Interface (SPI) implementation buffer results from triggered FPM actions.

When you trigger a round-trip not intending to re-load UIBB data, you have the option to skip the data retrieval. Please use parameter SKIP\_RETRIEVE of method BEFORE\_GET\_DATA for the same.

In MDG 6.1 you have the option to read and analyze the FPM event queue using an MDG-M provided method. This is very useful as an FPM event triggers the Web Dynpro phase model. To save process/method runtime in the feeder, the helper class CL\_MDG\_BS\_MAT\_ASSIST\_UI provides the two methods IS\_LAST\_EVENT and IS\_EVENT\_UNPROCESSED analyzing the event queue.

An example for the usage could be the modification of the screen header line: processing the method should be enough at the end of the event queue – not at each and every event.

### Implementation Hints

- If you enhance a feeder class (package MDG\_BS\_MAT\_UI) or a service provider class (package MDG\_BS\_MAT\_SP) be aware, that it is called for each event and roundtrip.

Be aware, that any feeder is called for any event and roundtrip. Therefore, check at the beginning of your enhancement, if the execution of the enhancement is necessary. Add a similar check in enhancements of the service provider class.

Instead of enhancing a feeder class, you can implement our own feeder class. To simplify the implementation you can inherit from the standard feeder. Replace the feeder by CBA or customizing of the component configuration. In this case the feeder is only called if the correct UIBB is involved.

- If you need for example derivations or validations data of other entities, please use the APIs mentioned in the “Master Data Governance Application Programming Interface Guide” (<http://scn.sap.com/docs/DOC-45127>).

Don't use the methods of class USMD\_MODEL\_EXT anymore. They don't reuse the data buffered in the MDG framework.

As the MDG framework already buffers the data for the governance UI, it is not necessary to create a separate buffer. It is better for performance if the data is only read from the framework when really required rather than keeping a separate buffer at any roundtrip.

### Upgrade to MDG 7.0 or higher

The question may be asked if it is better to upgrade to the next higher release in terms of performance reasons, for example, MDG 7.0. The next release offers you a slightly bigger data model covering more material master data entities like Document Management System Integration. To some extent this feature pack requires slightly more system resources. Please see the corresponding information in the available sizing guide: [www.service.sap.com/sizing](http://www.service.sap.com/sizing)

Compared to MDG 7.0 and seen from a pure runtime perspective, the application runs slightly faster at MDG 6.1 level. The reason behind is the material master data model that is a little smaller at MDG 6.1 than in higher releases. You can easily overcome this gap by removing unused UIBBs and/or entities.

### Running an EhP6 UI in an MDG 6.1 System

Customers may also go the other way round and upgrade their MDG installation to MDG 6.1 still using the EhP6 model scope. Here it is of interest how the EhP6 configuration behaves in the environment of a massively extended MDG 6.1 system.

In general a comparison may be slightly fuzzy as the two underlying performance test systems showed different CPU specifications. The EhP6 system operates on 8 instead of 4 virtual CPUs. However the Intel CPUs show the same core performance of 1.8GHz. Both systems are virtual, equipped with 32GB of RAM and running a SUN OS. As the successor system is an upgraded copy of the previous one and assuring that there is no load when taking the measurements you can use the results for a possible trend.

The EhP6 configuration and the governance scope functionality were used to reduce the UI to an EhP6 scope. The measurement was taken for the step from Initial Screen to the Create Material overview page.

**Result:**

An upgrade to MDG 6.1 on this specific test system costs an EhP6 configuration user less than 0.3 seconds of additional total runtime. The memory consumption rises by 15MB per user and process. Even the CPU consumption remains stable or decreases slightly. The results are valid for the step from Initial Page to the overview page.

**Note:** Using the EhP6 configuration in an MDG 6.1 upgraded system costs, using this hardware, a small amount of total runtime and causes an increase of 15MB of used memory – for each process and user for the mentioned step. This increase on system resources may be absorbed by tweaking your application according to this guide.

**Please be aware that the mentioned figures are a result from a very specific and optimized test environment and cannot be used 1:1 for comparisons outside of it. They shall serve as trend or outlook what may happen by an upgrade on an optimized system.**

## Performance Tweaks Checklist

The tips and tweaks of this document are summarized in a quick checklist. Please provide this checklist when you communicate with SAP regarding performance questions to speed up the processing of your questions. Thank you!

- What is the specification of your reference system (if applicable: number of SAPS)?
- What type of system (development, test, productive)?
- Native or virtual system?
- Which MDGM release is used?
- Did you test your installation like proposed in chapter "Reference System"
  - What are the results in runtime for the given example?
- How long does the application start take for the overview screen (OVP)?
  - With standard SAP configuration (Change Request and Basic Data expanded, others are collapsed)
  - With custom configuration? Please name the UIBBs expanded as well.
- Is the application configuration modified in your example?
  - Are custom UIBBs added?
  - Which implementation hints did you consider for your custom UIBB?
  - Did you check the implementation hints for custom code?
- If you face performance issues, did you...
  - Update the kernel to the latest patch?
  - Import the latest NetWeaver/FPM support packages
  - Consider SAP Note 2105467

## **Related Content**

[Master Data Governance for Material - Overview](#)

[Master Data Governance in SDN](#)

[Master Data Governance Sizing Hints](#)

[ABAP Performance Guide](#)

[MDG-M UI Extension](#)

[MDG-M Model Extension](#)

[MDG-M New Entity Types](#)

## Copyright

© 2015 SAP SE or an SAP SE affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE and its affiliated companies ("SAP SE Group") for informational purposes only, without representation or warranty of any kind, and SAP SE Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE 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.

SAP SE and other SAP SE products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE in Germany and other countries.

Please see

<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>

for additional trademark information and notices.