How To…
Perform Root Cause Analysis for MDM with Solution Manager Diagnostics
Version 1.00 – February 2008

Applicable Releases:
SAP NetWeaver MDM 5.5 SP06 Patch01
SAP Solution Manager 4.0 SP15
Data Unification
Master Data Management
Table of Contents

Change History .............................................................................................................. 1

1 Root Cause Analysis for MDM ................................................................................. 2
  1.1 Goal .......................................................................................................................... 2
  1.2 Basic Concepts of Root Cause Analysis ................................................................. 2
    1.2.1 End-to-End Root Cause Analysis ...................................................................... 2
    1.2.2 Component Root Cause Analysis ....................................................................... 2
  1.3 Introduction to SAP Solution Manager .................................................................... 3
  1.4 Architecture .............................................................................................................. 3
  1.5 Tools ......................................................................................................................... 5

2 OS and DB Analysis for MDM ................................................................................. 6
  2.1 MDM Command Console ....................................................................................... 6
    2.1.1 MDM Server Information .................................................................................. 8
      2.1.1.1 MDS Activity Overview .............................................................................. 8
      2.1.1.2 GetMDSVersion .......................................................................................... 10
      2.1.1.3 GetMDSStatus .......................................................................................... 11
      2.1.1.4 GetReportFiles .......................................................................................... 11
      2.1.1.5 GetReportFile .......................................................................................... 12
      2.1.1.6 GetMDSParameters .................................................................................. 12
      2.1.1.7 DisplayLog ................................................................................................. 12
      2.1.1.8 DisplayDiskSpaceFree .............................................................................. 12
    2.1.2 MDM Repository Information ........................................................................... 12
    2.1.3 MDM Database Information ............................................................................. 13
      2.1.3.1 GetDBRepositoryList .................................................................................. 13
      2.1.3.2 GetDBMSSetting ....................................................................................... 13
    2.1.4 MDM Server Logging ....................................................................................... 13
      2.1.4.1 SetProtocolLogLevel0 .............................................................................. 13
      2.1.4.2 SetProtocolLogLevel1 .............................................................................. 13
      2.1.4.3 SetDatabaseLogLevel0 ............................................................................. 13
      2.1.4.4 SetDatabaseLogLevel1 ............................................................................. 13
    2.1.5 MDM Server Performance Statistics ............................................................... 13
      2.1.5.1 SetWilyInstrumentationLevel10 .................................................................. 13
      2.1.5.2 SetWilyInstrumentationLevel13 ................................................................. 14
      2.1.5.3 SetWilyInstrumentationLevel15 ................................................................. 14
3 Workload Analysis for MDM ................................................................. 25

3.1 E2E Workload Analysis ................................................................. 26

3.1.1 Architecture .............................................................................. 26

3.1.2 Overview .................................................................................... 27

3.1.3 MDM-Specific Analysis ............................................................. 29

3.1.3.1 Workload Summary ............................................................... 30

3.1.3.2 Time Profile .......................................................................... 34

3.1.3.3 Files in Distribution Folder ..................................................... 35

3.1.3.4 Repositories .......................................................................... 36

3.1.3.5 Memory Allocation ............................................................... 37

3.1.3.6 CPU & Memory Util. .............................................................. 37

3.2 Component Workload Analysis with Wily Introscope ................... 38

3.2.1 MDM Workload and Performance Metrics ............................... 38

3.2.2 SAP – MDM Overview ............................................................... 39

3.2.2.1 MDM Server Performance and Workload Overview ............... 45

3.2.2.1.1 Client Requests – Avg. Response Time (ms) ......................... 45

3.2.2.1.2 Client Requests – Number of Executions per Time Measurement Interval 48

3.2.2.1.3 Details for Master Data Server Performance ........................ 49

3.2.2.1.3.1 Performance and Workload for Client Requests ............... 50

3.2.2.1.3.2 Performance and Workload for DB ................................. 51

3.2.2.1.3.3 Performance and Workload for Server/Repository Locks .... 52

3.2.2.1.3.4 Recommendation/KPI ...................................................... 54

3.2.2.1.3.5 Examples ....................................................................... 55

3.2.2.1.4 Details for Master Data Server Activity Overview ................. 57

3.2.2.1.4.1 Examples ....................................................................... 57

3.2.2.2 MDIS and MDSS Performance and Workload Overview ............ 63

3.2.2.2.1 Details for Imports and Syndications .................................... 65

3.2.2.2.1.1 Performance and Workload for Import Server .................... 66

3.2.2.2.1.1.1 Import Server – Avg. Response Time per Import Step (ms) .. 66

3.2.2.2.1.2 Import Server – Number of Import Steps per Measurement Time Interval 67

3.2.2.2.1.2 Examples ....................................................................... 68

3.2.2.2.1.3 Performance and Workload for Syndication Server .......... 69
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.2.1.3.1</td>
<td>Syndication Server – Avg. Response Time per Syndication Step (ms)</td>
<td>69</td>
</tr>
<tr>
<td>3.2.2.1.3.2</td>
<td>Syndication Server – Number of Syndication Steps</td>
<td>70</td>
</tr>
<tr>
<td>3.2.2.2</td>
<td>Details for Import and Syndication Activities</td>
<td>71</td>
</tr>
<tr>
<td>3.2.2.2.1.1</td>
<td>Import and Syndication Activity Overview</td>
<td>72</td>
</tr>
<tr>
<td>3.2.2.2.1.2</td>
<td>Active Import Steps</td>
<td>72</td>
</tr>
<tr>
<td>3.2.2.2.1.3</td>
<td>Active Syndication Steps</td>
<td>73</td>
</tr>
<tr>
<td>3.2.2.2.2.1</td>
<td>Files in Inbound Ports</td>
<td>74</td>
</tr>
<tr>
<td>3.2.2.2.2.2</td>
<td>Files in Outbound Ports</td>
<td>75</td>
</tr>
<tr>
<td>3.2.2.3.1</td>
<td>Number of Loaded Repositories</td>
<td>76</td>
</tr>
<tr>
<td>3.2.2.3.1.1</td>
<td>Repository Details</td>
<td>77</td>
</tr>
<tr>
<td>3.2.2.3.2</td>
<td>MDM Processes Memory Allocation</td>
<td>78</td>
</tr>
<tr>
<td>3.2.2.3.2.1</td>
<td>Resource Allocation Details</td>
<td>79</td>
</tr>
<tr>
<td>3.2.2.3.2.1.1</td>
<td>Number of Logged-on Users</td>
<td>79</td>
</tr>
<tr>
<td>3.2.2.3.2.1.2</td>
<td>Number of Workflows in Memory</td>
<td>80</td>
</tr>
<tr>
<td>3.2.2.3.3</td>
<td>MDM Processes CPU Utilization</td>
<td>81</td>
</tr>
<tr>
<td>3.2.2.3.3.1</td>
<td>OS Overview</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>Configuration Reporting for MDM</td>
<td>83</td>
</tr>
<tr>
<td>4.1</td>
<td>E2E Change Analysis</td>
<td>83</td>
</tr>
<tr>
<td>4.2</td>
<td>Configuration and File Reporting</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>Root Cause Analysis Roadmaps</td>
<td>88</td>
</tr>
<tr>
<td>5.1</td>
<td>End-to-end Workload Analysis</td>
<td>89</td>
</tr>
<tr>
<td>5.2</td>
<td>Component Workload Analysis</td>
<td>90</td>
</tr>
<tr>
<td>5.3</td>
<td>MDM Workload Analysis with Wily Introscope</td>
<td>91</td>
</tr>
<tr>
<td>6</td>
<td>Appendix</td>
<td>92</td>
</tr>
<tr>
<td>6.1</td>
<td>Links</td>
<td>92</td>
</tr>
</tbody>
</table>
## Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Chapter</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.02.2008</td>
<td>V1.0</td>
<td></td>
<td>Creation</td>
</tr>
</tbody>
</table>
1 Root Cause Analysis for MDM

1.1 Goal

For today's complex IT solutions, analyzing the root cause of an incident requires a systematic top-down approach to finally pinpoint the root cause of an incident. The standard root cause analysis offers systematic analysis of incidents.

Operationally, root cause analysis tools are designed to work towards reducing the number of resources in each step of the resolution process. A small team of support consultants with technical core competence in root cause analysis and a specific component expert who can dive deep into the component is enough to drive the issue and nail it down to a component.

This guide introduces the root cause analysis tools provided for MDM solutions and enables you to perform root cause analysis for SAP NetWeaver Master Data Management (MDM) using SAP Solution Manager.

1.2 Basic Concepts of Root Cause Analysis

Root cause analysis is either triggered by incident management or solution monitoring (system monitoring or business process monitoring).

1.2.1 End-to-End Root Cause Analysis

When the help desk within the business unit cannot solve an incident, the application management organization performs E2E root cause analysis in order to isolate the component responsible for a functional defect or a performance problem: client, network, various SAP systems (such as SAP NetWeaver Portal, SAP CRM and SAP ECC), database, or storage.

SAP Solution Manager supports E2E Root Cause Analysis to identify the component that is responsible for the error. This top-down approach systematically helps you target the cause of the error.

1.2.2 Component Root Cause Analysis

Once the application management team has performed E2E root cause analysis and has identified the problematic component, incident management allows the issue to be forwarded to the right expert in custom development, business process operations, SAP technical operations or IT infrastructure.

Various tools are needed to complete the different tasks in component root cause analysis. In a distributed customer solution, these tools may be located on different systems. SAP Solution Manager provides a central access point to the system landscapes and easy access to the required tools.
1.3 Introduction to SAP Solution Manager

SAP Solution Manager provides several scenarios to implement, administer, support, operate, and monitor customer’s SAP solutions:

- **Implementation of SAP solutions**
  - SAP methods and tools
  - Global rollout
  - Customizing synchronization
  - E-learning management
  - Test management

- **Solution Monitoring**
  - System monitoring
  - Business process monitoring
  - Central system administration
  - Solution reporting
  - Service level reporting
  - SAP EarlyWatch Alert

- **Upgrade of SAP solutions**
  - SAP methods and tools
  - E-learning management
  - Test management

- **Change Request Management**
  - Follows ITIL standards
  - Maintenance processes

- **Service Desk**
  - SAP Best Practices for incident management
  - Integration of third-party help desks

- **Solution Monitoring**
  - System monitoring
  - Business process monitoring
  - Central system administration
  - Solution reporting
  - Service level reporting
  - SAP EarlyWatch Alert

- **Delivery of SAP Services**
  - On-site/remote delivery
  - Issue management

The **Root Cause Analysis** scenario provides tools to perform root cause analysis for SAP solutions for ABAP and Java based SAP Systems as of SP10 of SAP Solution Manager.

**MDM systems can be integrated into the Root Cause Analysis scenario in SAP Solution Manager as of Solution Manager 4.0 SP15.**

The required version of MDM is MDM 5.5 SP06 Patch1 or higher. With previous versions of MDM 5.5, no integration into the Solution Manager Root Cause Analysis scenario is possible.

1.4 Architecture

E2E root cause analysis in SAP Solution Manager is based on a central diagnostics database that is populated with data by diagnostic agents running on each satellite system. These agents are preconfigured and delivered by SAP, wherein the data required to isolate the problematic component is obtained from the component systems. They continuously collect exceptions (such as critical log entries, dumps, and queue errors), configuration snapshots, workload data including operating system and database statistics from each satellite system. The information is kept uniform across all stacks and is available from one central console in SAP Solution Manager.

E2E root cause analysis can support root cause analysis for components implemented in ABAP, Java, C(++) like MDM, or components that run on the Microsoft .NET framework.
Here, E2E root cause analysis in SAP Solution Manager standardizes and systematically aggregates the following:

- performance and resource metrics
- access to technical configuration
- exceptions such as logs and dumps (program terminations)
- traces (recording the activity of a single user or process)
- tracking changes to software (code), configuration, or content

Documentation on installation and configuration of SAP Solution Manager Root Cause Analysis can be found at [http://service.sap.com/diagnostics](http://service.sap.com/diagnostics).
1.5 Tools

The Root Cause Analysis application in Solution Manager can be started using the following URL:

http://<host>:<port>/Solution Manager

Concrete example: http://lu0099.wdf.sap.corp:50000/Solution Manager.

It is a Web-enabled console which comprises the sections **Exceptions, Configuration, Workload, OS and DB, and Traces**.

The sections *Exceptions, Configuration, Workload, Availability*, and *Traces* include E2E root cause analysis capabilities:

- **Exceptions** provides unified access to exceptions reflecting in high severity log entries and dumps. Component specific log and dump viewers can be accessed directly from the E2E exception analysis. MDM 5.5 is not integrated into the exception analysis and does not report data here. However, the section *Exceptions* can be used to analyze exceptions from Java or ABAP based systems connected to MDM.

- **Workload** provides unified access to solution wide workload statistics for MDM and other SAP Systems in the E2E workload analysis as well as access to detailed non-aggregated MDM-specific performance metrics in Wily Introscope.

- **Traces** provides the E2E trace to record performance and functional defects in a single activity of a single user or process from browser to disk. The measurement is started at the user interface of the end user. MDM 5.5 is not yet integrated into the E2E trace. However the E2E trace can be used to analyze performance or functional defects in SAP NetWeaver Portal or Web Application Server systems connected to MDM.

- **Configuration** comprises the E2E change analysis that provides transparency on changes (code, business configuration, content) that were lately applied to production. This analysis is particularly instrumental if a few ad-hoc changes were applied to a production that results in disruptions after applying these changes. Changes to technical parameters in the MDM .ini files are reported here.

- **OS and DB** provides remote database monitoring for MDM databases, a file system browser to browse the MDM installation directories on OS level and to display MDM logs, and the MDM command console that allows you to access several MDM specific analysis functions.
2 OS and DB Analysis for MDM

2.1 MDM Command Console

The MDM Command Console provides functions for analyzing several aspects of the MDM server, repositories and database. It sends commands to the CLIX command line tool on the MDM server and displays the result in Solution Manager. The commands that can be executed from Solution Manager are non-manipulative commands that do not change the MDM server. The only exceptions are commands to change log levels and the granularity of performance data to be collected.

After selecting the relevant solution and starting the system, you can select the commands in the Command field. They are divided into five groups:

- **MDM Server Information** provides commands for analyzing various aspects of the MDM server such as activity overview or obtain MDM version information.
- **MDM Repository Information** provides commands for analyzing various aspects of MDM repositories such as list available archives or master/slave information.
- **MDM Database Information** provides commands for obtaining MDM Database settings
- **MDM Server Logging** allows you to change the detail level of information that is written to MDM logs
- **MDM Server Performance Statistics** allows you to change the granularity of the performance data to be reported to Wily Introscope
Example: Getting the activity overview for the attached MDM host – no parameters needed.

If a command needs further parameters, the values have to be defined in the Parameters line. The required parameters can be seen using the Help button.

After you send the command, the same result will be shown as using the CLIX commands at operating system level.
Using the *Recurrent* option, the command will be sent continuously to the MDM server depending on the definition in an interval in seconds.

2.1.1 MDM Server Information

The following chapters describe commands that return MDM server-related information:

- MDS Activity Overview
- GetMDSVersion
- GetMDSStatus
- GetReportFiles
- GetReportFile
- GetMDSParameters
- DisplayLog
- DisplayDiskSpaceFree

2.1.1.1 MDS Activity Overview

The *MDSActivity* command provides a snapshot of all client requests that are currently active or waiting in the MDM server. It provides an overview of MDS activity and can help to analyze performance bottlenecks. The following information is displayed:

<table>
<thead>
<tr>
<th>Thread</th>
<th>State</th>
<th>User</th>
<th>Protocol</th>
<th>Command</th>
<th>Locks</th>
</tr>
</thead>
</table>

- **Thread**: Number of the MDS thread that executes the request.
- **State**: Shows whether the thread is active or waiting for a resource.
  - W = Wait // waiting for some resource, currently a lock, such as a server lock or repository lock or synchronization locks
  - R = Run // running
- **User**: User who triggered the action.
- **Protocol**: Reports the MDM internal protocol that is used. The different clients can access the MDM server using different MDM specific protocols. The following lists some of the most important protocols and their meaning:
  - *CatMgrServer* Data Manager
  - *CMSrvServer* Java API
  - *MDSAdmin* Console
  - *Itsam* CCMS Agent
  - See also chapter 4.2.2.1.1 Client Requests – Avg.Response Time (ms), figure Overview for MDM 5.5 SP06
- **Command**: Shows the command or MDS method that is currently being processed. Examples of methods are: *Login*, *StartMatching*, *GetAgencyKeys*

- **Locks**: Currently the following types of locks exist that will be acquired and gained by the client requests.
  - *Server Lock*
  - *Repository Lock*
  - *Synchronization Server Lock*
  - *Synchronization Repository Lock*
The first four characters in the lock column report the locks required; the latter four report the locks that have been gained or acquired so far by the activity.

<table>
<thead>
<tr>
<th>Locks</th>
</tr>
</thead>
<tbody>
<tr>
<td>------</td>
</tr>
<tr>
<td>-W-----</td>
</tr>
<tr>
<td>-W-----</td>
</tr>
<tr>
<td>-R-R--</td>
</tr>
</tbody>
</table>

There are three possible states per lock:
- R Read
- W Write
- - Not active.

**Example:** RW-- R---

The activity requests a **Server Read** and **Repository Write** lock and no synchronization locks. Furthermore, it already gained the server lock. Obviously this activity is acquiring the **Repository Write** lock. Its state is **Wait**.

- **Elapsed time**: Elapsed time in msecs since the activity started
- **Connection**: Application that connected to a repository
- **Repository**: Connected repositories

**Example: Matching Run**

```
<table>
<thead>
<tr>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
</tr>
<tr>
<td>Group:</td>
</tr>
<tr>
<td>Command:</td>
</tr>
<tr>
<td>Parameters:</td>
</tr>
<tr>
<td>Option:</td>
</tr>
<tr>
<td>Interval:</td>
</tr>
<tr>
<td>Stop Command</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread State</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>476 R Admin</td>
</tr>
<tr>
<td>3468 R</td>
</tr>
</tbody>
</table>
```

- **Matching** is running (state R), required a **Read Repository** lock and already gained this lock.
Example: Import and parallel user action

In total, three client requests are active. One activity processes a request for user Admin and one activity processes a request for user Expert. The third activity (Itsam) is triggered by the MDM Command Console itself and collects this monitoring data.

- Thread 5640 is running (State R) since 7440 msecs for user Admin. It performs an Import Records action on the business partner repository. In order to perform this action, it required a Write Repository lock and already gained this lock.

- Thread 1020 is waiting (State W) since 7393 msecs for user Expert. It performs a read on the repository (GetRecordIds). This action requires a Read Repository lock on the business partner repository. The lock is not gained so far because another process is holding a repository write lock.

2.1.1.2 GetMDSVersion

The GetMDSVersion command displays the version number and interface information of the MDM server installed on the selected host.
2.1.1.3 GetMDSStatus
The GetMDSStatus command shows information about repositories mounted on the MDM server.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository</td>
<td>The name of the repository</td>
</tr>
<tr>
<td>DBMS Server*</td>
<td>The DBMS Server on which the repository is stored</td>
</tr>
<tr>
<td>DBMS Type*</td>
<td>The DBMS type in which the MDM is stored</td>
</tr>
</tbody>
</table>

2.1.1.4 GetReportFiles
The GetReportFiles command returns a list of the files in the MDM Server's Log Dir.

The log information is an MDM system table with a predefined set of fields and records. Each record corresponds to an MDM system log that MDM maintains for each MDM Server.

The properties for each log are as follows:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date/Time</td>
<td>The date and time the system log was last modified</td>
</tr>
</tbody>
</table>
| MDM Log Type | • Server  
|              | • Audit  
|              | • Assertion  
|              | • Protocol  
|              | • Transaction  |

The Assertion logs for example contain internal runtime information. In the event of a failed operation, these logs can be used to help identify the cause of a failure; in normal operation, the logged information can be treated as non-error operational information.
2.1.1.5  GetReportFile
The GetReportFile command displays the contents of a file from the MDM Server's Log Dir.

2.1.1.6  GetMDSParameters
The GetMDSParameters command displays the contents of the MDM Server's mds.ini file.

2.1.1.7  DisplayLog
The DisplayLog command displays the contents of the MDM Server's logs (Server@etc.xml) file.

2.1.1.8  DisplayDiskSpaceFree
The DisplayDiskSpaceFree command displays the free disk space in MDM Server's report, archive, and accelerator directories.

2.1.2  MDM Repository Information
The RepositoryVerify command could be used in case of MDM server crash or restart scenarios. The RepositoryVerify command verifies a repository but makes no repairs. Keep in mind that this command generates additional load on the MDM Server. In case the repository cannot be loaded, the administrator should go on with check/verify/repair activities on the specific repositories. It is not necessary to perform these activities automatically.

The following further commands return repository-related information:
- GetRepositoryChangeCounter
- GetRepositoryMasterInfo
- GetRepositoryMasterChangeCounter
- DisplayAnalysisReport
- DisplayArchives

For details, refer to the MDM 5.5 Console Reference Guide.
2.1.3 MDM Database Information
The following commands return DB-related information.

2.1.3.1 GetDBRepositoryList
The GetDBRepositoryList command displays a list of repositories that are stored on the
DBMS instance.

2.1.3.2 GetDBMSSetting
The GetDBMSSetting command lists all the DBMS settings for that DBMS instance.

2.1.4 MDM Server Logging
The commands in this section allow you to increase the log level of information written to
the MDM server logs (Server…..xml). The log levels can be increased separately for the
MDM Server protocol methods and for the database interface. For a description of the
MDM server logs, refer also to section 3.2.3.

2.1.4.1 SetProtocolLogLevel0
This command sets the MDM server protocol log level to the default setting. For
productive use, the MDM server should run with this protocol level. It should only be
temporary increased for analysis purposes.

2.1.4.2 SetProtocolLogLevel1
This command increases the MDM server protocol log level. In addition to the normal log
content, MDM Server Protocol method executions and their elapsed time are being
written to the log. This log level should only be kept temporarily for analysis purposes.
Remember to decrease the log level back to 0 after your analysis is finished.

2.1.4.3 SetDatabaseLogLevel0
This command sets the MDM Server database interface log level to the default setting.
For productive use, the MDM Server should run with this protocol level. It should only be
temporarily increased for analysis purposes.

2.1.4.4 SetDatabaseLogLevel1
This command increases the MDM database protocol log level. In addition to the normal
log content, database statements and their elapsed time are written to the log. This log
level should only be kept temporarily for analysis purposes. Remember to decrease the
log level back to 0 after your analysis is finished.

2.1.5 MDM Server Performance Statistics
The commands in this section allow you to increase the detail level of performance
metrics reported to Wily Introscope MDM Server logs (Server…..xml). For a description
of the performance metrics, refer also to chapter 4.2.

2.1.5.1 SetWilyInstrumentationLevel10
This is the default Wily instrumentation level. For productive use, the MDM server should
run with this Wily instrumentation level. It should only be temporarily increased for
analysis purposes.
2.1.5.2 SetWilyInstrumentationLevel13

This command increases the Wily instrumentation level to 13. In addition to the default performance metrics, the number of lock requests and their elapsed time are reported to Wily Introscope. This log level should only be kept temporarily for analysis purposes. Remember to decrease the Wily instrumentation level back to 10 after your analysis is finished.

2.1.5.3 SetWilyInstrumentationLevel15

This command increases the Wily Instrumentation Level to 15. In addition to the default performance metrics, the number of database statements and their elapsed time are written to the log. This instrumentation level should only be kept temporarily for analysis purposes. Remember to decrease the Wily instrumentation level back to 10 after your analysis is finished.

Note: All settings to increase the protocol or log levels of the MDM server can be performed without having to stop and restart the MDM server on the remote host. On the other hand, the settings are also made in the mds.ini file. So with the next manual restart the last settings are used.
2.2 File System Browser

The file system browser allows central and safe read-only access to the MDM installation directories. With the file system browser it is possible to browse through the MDM installation directories, to display MDM logs and to display files in the distribution folders.

2.2.1 File System Browser

After you select the relevant solution and start the system, the MDM folder structure is displayed where all the MDM relevant log and trace files are physically stored.
The file system browser gives you the opportunity to search for different kinds of files without logging on to the operating system itself.

2.2.2 Log Browser

The Solution Manager provides the MDM log XSLT transformation to display the logs in MDM browser format.

When you select a log file, the following unformatted structure is displayed.
The format can be adjusted by using the *MDM Log Formatter*.
The log file will then be shown in the MDM browser format.

Currently this is only supported for Windows platforms. On Unix, open XML log files in the MDS file system do not include the closing XML tags. This is because: (1) Unix administrators generally use tail to follow progress in the logs; and (2) writing new data before the end of the log and then appending the closing tags would confuse tail, and more importantly crash MDS due to the way Unix deals with file manipulation. To view a snapshot of an open log in an XML viewer, first make a copy of the log and then append the following closing tags to the copy: </Open></MDM_Log>

### Note:
New features coming with SP06 increase readability of logs/traces. It distinguishes between informational and error messages and provides additional detail information for each message. It is possible to filter for error messages and complete time.
At the moment, the additional detail information is only offered in the following cases:

- if you view the log via the MDM Console which will use the corresponding stylesheet
- if you download the log file using Solution Manager and place both the log file and the `MDM_Log.xsl` file in one folder.

In Windows, the `MDM_Log.xsl` is located in the `MDM_DIR\Server\etc` folder.

---

**MDM Server Log v1.0**

<table>
<thead>
<tr>
<th>Thread</th>
<th>Timestamp</th>
<th>Operation</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>652</td>
<td>2007/12/17 14:07:12.782</td>
<td></td>
<td>MDS Configured for CPU Count of 2</td>
</tr>
<tr>
<td>652</td>
<td>2007/12/17 14:07:13.032</td>
<td></td>
<td>NCS Library 'C:\PROGRA~1\SAPMDM-2.9\Server\nes.dll' loaded successfully</td>
</tr>
<tr>
<td>652</td>
<td>2007/12/17 14:07:14.970</td>
<td></td>
<td>Init Database Tag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Business Functions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pl17611MRSQ_15.7.4.3</td>
</tr>
<tr>
<td>652</td>
<td>2007/12/17 14:07:15.676</td>
<td></td>
<td>Init Database Tag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ProductPl17611MRSQ_7.7.4.3</td>
</tr>
<tr>
<td>652</td>
<td>2007/12/17 14:07:17.110</td>
<td></td>
<td>Init Database Tag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MaterialPl17611MRSQ_7.7.4.3</td>
</tr>
<tr>
<td>652</td>
<td>2007/12/17 14:07:19.673</td>
<td></td>
<td>Error: Unable to register server in System landscape Directory: Call to SLDREG Utility in \usr\sap\sidreg\ridreg failed, Error 0x1. Set proper value for parameter 'SLDReg Utility Path' in server configuration file. If you are not using SLD, set parameter 'SLD Registration' to false in server configuration file so that the server does not try to register with SLD. This will avoid this error in the future.</td>
</tr>
</tbody>
</table>

To analyze an action, the following logs can be used:

- **Audit** – login/logout information
- **Server** – system level (startup/shutdown), major operations (update, create slave), SQL errors (if enabled), performance timers
- **Assertion** – assertions which are tests for invalid state, and code level errors (if enabled).
- **Transaction and Protocol** – incoming request tracking including timeouts, network send failures and critical failures
**MDM Server 5.5**
The MDM Server 5.5 writes the following logs that can be found in listed folders (for Windows):

- **MDM_DIR\Server\Logs\Assertion*.xml** Errors
- **MDM_DIR\Server\Logs\Server*.xml** Major server operations
- **MDM_DIR\Server\Logs\Audit*.xml** Login, logout tracking
- **MDM_DIR\Server\Logs\Protocol*.xml** Server level requests
- **MDM_DIR\Server\Logs\Transaction*.xml** Repository level requests
- **MDM_DIR\Server\Reports\*.xml** Long-running operations
- **MDM_DIR\Server\Distribution\etc\*.xml** Status of import or export on distribution ports

On Unix platforms there are the following log locations for the MDS:
- `/home/mdm/mdm/mds/Logs/…`
- `/home/mdm/mdm/mds/Reports/…`
- `/home/mdm/mdm/mds/Distributions/…`

**MDM Syndication Server 5.5**

- **MDM_DIR\Syndication Server\*.xml** Server logs/logs

**MDM Import Server 5.5**

- **MDM_DIR\Import Server\Temp\*.** Temp files during the import process
  - these files are either moved [to MDS] or removed after the import process.

While MDIS is processing a port, the logs for the import task are written to a temp directory where MDIS is installed (default) or specified by `Logs` parameter in *MDIS.ini*. Once an import task is finished, the related logs are moved to the corresponding port/log directory on the MDS side.

- **MDM_DIR\Import Server\Logs\*.xml** Server logs/assertion logs

Server Log – The server log is created under the `Logs` directory where MDIS is installed (default) or specified by the `Logs` parameter in *MDIS.ini*. The server log is for general issues, for example could not log in to the repositories, could not connect to MDM server, etc.

**MDM GUI Clients 5.5**

**Console:**
- **MDM_DIR\Console\AdminConsole.log** Jobs, notifications and errors

**Data Manager:**
- **MDM_DIR\Data Manager\ClientAssertLog*.xml** Errors
- **MDM_DIR\Data Manager\Connection*.xml** Requests to MDS

**Workflow:**
- **MDM_DIR\Workflow\*.xml**
2.2.3 Performance Log

In addition to the normal informational and error log messages, the MDM server can also write performance data to the MDM Server log. This is useful if you want to investigate a performance problem. It allows you to determine the runtime of MDM protocol methods or SQL Statements.

The extent of the information that is written to the MDM server log can be controlled with the following mds.ini parameters:

- **Protocol Log Level**
- **Database Log Level**

Possible values are 0 and 1.

The settings can be changed on the fly using the MDM Console commands described in section 3.1.4.

The default settings are as follows:

```plaintext
Protocol Log Level=0
Database Log Level=0
```

With these settings the default content (informational and error messages) are written to the MDM server log, no performance data will be logged.

With the setting **Database Log Level=1**, additional SQL commands issued by the MDM server and their execution time measured on MDM server side are written to the log.

<table>
<thead>
<tr>
<th>Thread</th>
<th>Timestamp</th>
<th>Operation</th>
<th>Action</th>
<th>Details</th>
<th>Segment Time (ms)</th>
<th>Complete Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5280</td>
<td>2008/01/20 16:00:41.075</td>
<td>Database</td>
<td>Exec</td>
<td>SELECT COUNT(*) FROM All_xCat_Dbs..All_ServerStatus WITH (TABLOCK)</td>
<td>0.736</td>
<td></td>
</tr>
<tr>
<td>5280</td>
<td>2008/01/20 16:00:41.091</td>
<td>Database</td>
<td>Exec</td>
<td>select GetDate();</td>
<td>0.712</td>
<td></td>
</tr>
<tr>
<td>5280</td>
<td>2008/01/20 16:00:41.091</td>
<td>Database</td>
<td>Exec</td>
<td>SELECT SettingString FROM SPGetArticleSettings.m000..All_CatalogSettings GEDE SettingId=66</td>
<td>0.330</td>
<td></td>
</tr>
<tr>
<td>5280</td>
<td>2008/01/20 16:00:41.091</td>
<td>Database</td>
<td>Exec</td>
<td>SELECT Catalog_Name, ServerName, LastCheckIn, ServerLockMask FROM All_xCat_Dbs..All_ServerStatus WHERE Catalog_Name=SPA_F3_ARTICLE_ENFTY</td>
<td>0.502</td>
<td></td>
</tr>
<tr>
<td>5280</td>
<td>2008/01/20 16:00:41.091</td>
<td>Database</td>
<td>Exec</td>
<td>SELECT ServerName, LastCheckIn, GetDate(), ServerLockMask FROM All_xCat_Dbs..All_ServerStatus WHERE Catalog_Name=SPA_F3_ARTICLE_ENFTY</td>
<td>0.559</td>
<td></td>
</tr>
</tbody>
</table>

The example above shows log entries written with **Database Log Level=1**. The columns provide the following information:

- **Thread** is the MDS internal thread ID that sent the SQL call to the database.
- **Timestamp** when the action was executed.
- **Operation** is always **Database** for database statements.
- **Action** is always **Exec** for database statements.
- **Details** shows the executed database statement.
- **Segment Time** shows the execution time of the SQL call measured on MDM server DB interface.
- **Complete Time** is not used for logging database statements.
With the setting `Protocol Log Level=1`, in addition to the default content, the execution of MDS internal protocol methods is also written to the MDM log.

<table>
<thead>
<tr>
<th>Thread</th>
<th>Timestamp</th>
<th>Operation</th>
<th>Action</th>
<th>Details</th>
<th>Segment Time (ms)</th>
<th>Complete Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5644</td>
<td>2008/01/17 14:25:57.338</td>
<td>Protocol Enter</td>
<td>Enter</td>
<td>MDSInternalServerCommand_ValidateProtocol</td>
<td>0.021</td>
<td>0.021</td>
</tr>
<tr>
<td>5644</td>
<td>2008/01/17 14:25:57.338</td>
<td>Protocol Exit</td>
<td>Exit</td>
<td>MDSInternalServerCommand_ValidateProtocol</td>
<td>0.311</td>
<td>0.332</td>
</tr>
<tr>
<td>5644</td>
<td>2008/01/17 14:25:57.448</td>
<td>Protocol Enter</td>
<td>Enter</td>
<td>MDSInternalServerCommand_CloseSession</td>
<td>0.701</td>
<td>0.701</td>
</tr>
<tr>
<td>5644</td>
<td>2008/01/17 14:25:57.448</td>
<td>Protocol Exit</td>
<td>Exit</td>
<td>MDSInternalServerCommand_CloseSession</td>
<td>0.328</td>
<td>0.328</td>
</tr>
<tr>
<td>5644</td>
<td>2008/01/17 14:26:30.288</td>
<td>Protocol Enter</td>
<td>Enter</td>
<td>MDSAdminServerCommand_GetFile()</td>
<td>0.081</td>
<td>0.081</td>
</tr>
<tr>
<td>5644</td>
<td>2008/01/17 14:26:30.288</td>
<td>Protocol Exit</td>
<td>Exit</td>
<td>MDSAdminServerCommand_GetFile()</td>
<td>0.081</td>
<td>0.081</td>
</tr>
</tbody>
</table>

The example above shows log entries written with `Protocol Log Level=1`. The columns provide the following information:

- **Thread** is the MDS internal thread ID that executed the Protocol method.
- **Timestamp** when the action was executed.
- **Operation** is always **Protocol** for executed MDM methods.
- **Action** can be **Enter**, **Acquire ..Lock**, **Execute**, **Returned ... Lock**, **Exit**.

One execution of a MDM Protocol method writes several steps in the MDM server log. The beginning first step is always **Enter**, the last step is always **Exit**.

- **Details** shows the name of the executed protocol method.
- **Segment Time** shows the execution time of the specific action.
- **Complete Time** shows the total execution time of the protocol method since **Enter**.

In this example, three protocol methods were executed by the same MDS internal thread 5644:

1. Method MDSPublicServerCommand_ValidateProtocol took 332 msecs in total. It did not require any locks.
2. Method MDSPublicServerCommand_CloseSession took 332 msecs in total. It did not require any locks.
3. Method MDSAdminServerCommand_GetFile() took 1208932 msecs in total. It acquired a read lock. Most of the time was spent in the method execution itself with 1196777 msecs.

The buttons on top of the log allow you to sort the log by different columns. It is possible to filter by a specific thread and to search within the logs, for example.

**Note:** The settings `Protocol Log Level=1` and `Database Log Level=1` effect all actions on the MDM server. It is not possible to restrict them to certain users or activities. With these settings a higher amount of data is written to the logs, therefore these settings should be kept only temporary for the investigation of performance problems.
The following provides a guideline for the investigation of performance problems using the performance log:

1. Ideally, perform your investigation during a time period with little or no other activity on MDS.
2. Use the MDM command console to increase database and protocol log levels.
3. Reproduce the performance problem.
4. Use the MDM command console to decrease database and protocol log levels.
5. Use the file system browser to download the log to your local PC and open it. Alternatively, open the log in the MDM Console.
6. Sort by descending segment time to identify long-running actions or DB statements.
7. Filter the thread ID of the long-running actions to get a complete overview.
2.3 Remote Database Monitoring

The DBA Cockpit in SAP Solution Manager provides database monitoring similar to the Database Performance Monitor (transaction ST04) in the SAP NetWeaver ABAP stack. It allows you to check the performance and configuration of the MDM database and to check the database log for critical DB errors. You can check the performance overview to determine whether DB performance is sufficient, to check for expensive SQL statements and to check the database logs for critical error messages.

The DBA Cockpit functions are DBMS-specific. Refer to the following SAP Notes for DBMS-specific documentation:

<table>
<thead>
<tr>
<th>DBMS</th>
<th>SAP Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>1028624</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>1027512</td>
</tr>
<tr>
<td>DB2 UDB</td>
<td>1027336</td>
</tr>
<tr>
<td>MaxDB</td>
<td>1028751 *</td>
</tr>
</tbody>
</table>

* Online documentation for remote database monitoring of MaxDB is available at: http://help.sap.com/saphelp_nw2004s/helpdata/en/fb/3e963c98bdc358e10000000a11405a/frameset.htm
3 Workload Analysis for MDM

Solution Manager Diagnostics provides two applications for performing workload analysis:
End-to End (E2E) Workload Analysis and Wily Introscope.

The **E2E Workload Analysis** application provides hourly-aggregated performance and resource consumption data. It allows you to compare the performance data of multiple systems and has a very flexible data display. It provides solution-wide performance data and enables you to isolate a system or software component that caused performance problems. The E2E Workload Analysis is the starting point for the root cause analysis of performance problems.

**Wily Introscope** provides non-aggregated performance and resource data for MDM and Java systems. It can be used to perform a detailed analysis of the MDM or Java system once the E2E Workload Analysis isolates a system or software component that caused performance problems.

The figure above shows the entry screen of the workload analysis. It provides links to the E2E Workload Analysis and to Wily Introscope on the left side.
3.1 E2E Workload Analysis

3.1.1 Architecture

The following figure shows the infrastructure that is required for the E2E Workload Analysis.

Diagnostics agents are installed on all hosts where SAP system software components are installed. They collect, aggregate and transfer online diagnostics data to the Wily Introscope Enterprise Manager (EM) every 15 seconds.

The Introscope Enterprise Manager is part of the CA Wily Introscope system management application. It is installed as part of SAP Solution Manager. The Introscope Enterprise Manager acts as repository of all performance metrics. It receives performance metrics from one or more Diagnostics Agents.

An hourly scheduled extractor batch job reads performance data from the Introscope Enterprise Manager, aggregates them on an hourly basis and writes the aggregates to the Business Intelligence part of the Solution Manager.

The E2E workload accesses the BI data and displays them flexibly.
3.1.2 Overview
The E2E workload analysis for MDM provides the following information:

### Overview: Avg. Client Request Response Time

<table>
<thead>
<tr>
<th>Workload Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDS Workload overview (Avg. Resp.Time (ms) &amp; Executions)</td>
</tr>
<tr>
<td>MDIS Workload overview (Avg. Resp.Time (ms) &amp; Executions)</td>
</tr>
<tr>
<td>MDSS Workload overview (Avg. Resp.Time (ms) &amp; Executions)</td>
</tr>
</tbody>
</table>

### Time Profile

<table>
<thead>
<tr>
<th>MDM Server Time Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Response Time [ms]</td>
</tr>
<tr>
<td>Average Number of Executions / h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MDM Import Server Time Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Response Time [ms]</td>
</tr>
<tr>
<td>Average Number of Executions / h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MDM Syndication Server Time Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Response Time [ms]</td>
</tr>
<tr>
<td>Average Number of Executions / h</td>
</tr>
</tbody>
</table>

### Files in Distribution Folder

<table>
<thead>
<tr>
<th>Total Number of Files in Distribution Folders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files in Distribution Folder per Repository</td>
</tr>
</tbody>
</table>

### Repositories

<table>
<thead>
<tr>
<th>Repository Growth [Number of main table records]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository Metrics [Avg. &amp; Max. Number of main, lookup, qualified table records]</td>
</tr>
</tbody>
</table>

### Memory Allocation

<table>
<thead>
<tr>
<th>Memory Allocation of Server Processes on OS Level</th>
</tr>
</thead>
</table>

### CPU & Memory Util.

<table>
<thead>
<tr>
<th>Metric Data Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Data</td>
</tr>
</tbody>
</table>
To start the workload and performance analysis, choose Workload → E2E Workload Analysis from the menu. Choose your solution and select all components. Click Start to begin your investigation.

The figure above shows the entry screen of the E2E Workload Analysis. Here you can get a good overview of the workload in your solution for the chosen timeframe. In the example above, the solution consists of a MDM system, an SAP NetWeaver Portal and an ECC system.

The Time Profile shows the average response time for the different software components and systems. This allows you to quickly identify software components with bad response times or time intervals where the performance degrades.

The workload metrics overview shows important KPIs for all systems and software components of the solution.

The KPIs displayed for MDM are:

- **Average Client Request Response Time**: Average time spent in the MDM Server to process client requests. Client requests can be a request to search for specific records, update records, perform matching or import new records, for example. They originate from the Portal (via Java API), ABAP back-end systems (via ABAP API), MDM Data Manager, Syndication Server, Import Server, and so on.

- **Number of Imports**: Displays the number of imported files or chunks of files (if the number of records in a file is larger than the chunk size in the Import Server) during the time interval selected. This metric shows only the automatic imports from Import Server. Manual Imports via Import Manager are not counted here.
- Number of Syndications: Displays the number of syndications during the time interval selected. This metric shows only the automatic syndications from Syndication Server. Manual syndications by the Syndicator are not counted here.

- Number of Records in Largest Repository: Displays the number of main table entries in the largest repository of the MDM Server.

3.1.3 MDM-Specific Analysis

Switch to the Master Data Server tab for more MDM-specific performance analysis views.

This tab provides the following views for MDM:

- Workload Summary
- Time Profile
- Files in Distribution Folder
- Repositories
- Memory Allocation
- CPU & Memory Util.
3.1.3.1 Workload Summary

The workload summary provides a workload overview for the MDM Server (MDS), the Import Server (MDIS) and the Syndication Server (MDSS) within the selected timeframe.

The MDM Server Workload Overview shows performance data reported by the MDM server. It provides the average response time and the number of executed requests per client type (Java API, ABAP API, Data Manager, and so on) in the MDM Server (MDS). For some administrative requests, such as login, the client type is not transferred to the MDM server. In these cases the MDM internal communication protocol is displayed instead (CMSrvServer, CatMgrServer).

As a rule of thumb, focus your investigation on the client types which indicate user interactions such as ABAP API, Java API, and Data Manager. **High average response times for user interactions might indicate general performance issues.**

The MDM Import Server Workload Overview shows performance data reported by the Import Server. It provides the average response time and the number of executed import steps for the Import Server (MDIS).
The following Import steps are displayed:
1. **Scanning Ports**: MDIS regularly triggers a port scan. During the port scan it checks if files exist in the port folder structure that need to be imported. If the port scan finds files, it triggers an import; otherwise it waits for the time interval defined in `mdis.ini` until it repeats the port scan.

2. The actual import starts by separating a file into several chunks if its row count exceeds the chunk size defined in `mdis.ini`. This step takes usually only a few ms and is therefore not displayed in the workload analysis.

3. **xStructuralTransformer**: The import file or import chunk undergo a structure and value mapping where the source structure is mapped to the target structure and source values are replaced by target values. The time consumed for these steps is reported here. The number of executions corresponds to the number of chunks or files transformed. The time spent for the transformation depends strongly on the size of the import files. Therefore, no general threshold can be given here.

4. **xImporter**: The transformed file or chunk is being imported to the MDS. The time consumed for the import is reported here. The number of executions corresponds to the number of chunks or files imported. The time spent for the transformation depends strongly on the size of the import files. Therefore, no general threshold can be given here. However, increasing import times for similar file or chunk size could indicate performance issues in the MDS.

The figure above depicts the import steps graphically. Grey boxes depict the steps taking place in the Import Server; blue boxes indicate the steps where the MDM Server is triggered from the Import Server. The yellow bubbles show the steps that are reported in the workload analysis:
1) Scanning port
2) Structural transformation and value transformation
3) Import

The **MDM Syndication Server Workload Overview** shows performance data reported by the Syndication Server. It provides the average response time and the number of executed syndication steps for the Syndication Server (MDSS).
The following Syndication steps are displayed:

1. **Query Execution**: MDSS regularly triggers a query execution in MDS and waits for the answer. The query returns repository main table records that need to be syndicated. As a rule of thumb, the query execution time should not exceed a few seconds.

2. **Export**: MDSS retrieves from MDS the **records to be syndicated and writes them for** Windows platforms to a temporary file in the user’s TEMP folder (the user under which account MDSS is running). This time is reported here.

3. **Copy to Memory**: The syndication results are compiled in a big object in memory.

4. **Copy to Port**: MDSS thread sends the blob (Binary Large Objects) to MDS and waits for completion. MDS unwraps the blob and writes its content to a file in the distribution folder. The MDSS thread is waiting for the answer. Therefore the *Copy to Port* metric reports mainly the time that MDSS waits for MDS.
The figure above depicts the syndication steps. The yellow bubbles mark the syndication steps that are reported in the workload analysis.
3.1.3.2 Time Profile

The Time Profile allows you to identify peak hours or hours with bad response times for MDS, MDIS and MDSS. You can check the influence of imports and syndications on the MDS performance by comparing the MDSS/MDIS time profiles with the MDS time profile.
3.1.3.3 Files in Distribution Folder

The Files in Distribution Folder view allows you to identify import or syndication backlogs. It displays the maximum number of files in the inbound or outbound ports for all repositories during a specific hour of the day. The number of files in the inbound/outbound Ready folders can shortly increase but should show a tendency to decrease over the course of time.

An increase in the number of files in the inbound Ready folders over a long time can indicate problems with the Import Server. Either the Import Server is importing too slowly, or it is not active at all.

The metric Inbound Exceptions shows the number of files in the inbound Exceptions folder. Normally the number of files here should be 0. Files in the inbound Exception folder were rejected by the MDIS as it was not able to perform mapping for them. They need to be reprocessed manually using the Import Manager.

An increase in the number of files in the outbound Ready folders over a long time can indicate issues with the connected PI (XI) systems. Either the PI (XI) system fetches and distributes the files too slowly, or not at all.

Note: Files in the outbound Ready folder will only be reported as of MDM 5.5 SP06 Patch2.
The *Files in Distribution Folder per Repositories* table allows you to identify the repositories with the highest number of files if multiple repositories are active on the MDM Server.

### Files in Distribution Folder per Repository

<table>
<thead>
<tr>
<th>Repository</th>
<th>INBOUND EXCEPTIONS</th>
<th>INBOUND FILES READY</th>
<th>OUTBOUND EXCEPTIONS</th>
<th>OUTBOUND FILES READY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA_TEST_UU1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ADMIN_META_UNIT_TESTS_010</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ADMIN_META_UNIT_TESTS_SPO5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ARTICLE_Risk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Approve_Roles</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Article</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Article_22_10_07</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Automation_Synchronization</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BP_IMPORTMANAGER</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BUSINESS_PARTNER</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### 3.1.3.4 Repositories

The *Repository Growth* chart gives an overview of the record change in the main table entry over the course of time. It allows you to identify fast-growing repositories.

The *Repository Metrics* show the average and maximum number of records in the main table, records in lookup tables and records in qualified lookup tables within the selected time interval.

### Repository Growth

![Repository Growth Chart]

### Repository Metrics

<table>
<thead>
<tr>
<th>Repository Metric</th>
<th>LOOKUP TABLE ENTRIES</th>
<th>MAIN TABLE ENTRIES</th>
<th>QUALIFIED LINKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository</td>
<td>Avg. Number of Entries</td>
<td>Maximum Number of Entries</td>
<td>Avg. Number of Entries</td>
</tr>
<tr>
<td>BUSINESS_PARTNER</td>
<td>2,643</td>
<td>2,643</td>
<td>22</td>
</tr>
<tr>
<td>InQuere</td>
<td>2,692</td>
<td>3,090</td>
<td>6,691</td>
</tr>
<tr>
<td>InQuere_AAD</td>
<td>2,692</td>
<td>2,962</td>
<td>0,357</td>
</tr>
</tbody>
</table>

- 36 -
3.1.3.5 Memory Allocation

This view provides the memory allocation of the MDM Server processes measured at operating system level. The memory allocation of the MDS Server depends mainly on the size of the repository, but also on other factors such as workflows in memory or logged on users.

![Memory Allocation Graph](image)

3.1.3.6 CPU & Memory Util.

This view provides important OS resource consumption KPIs for the physical hosts running the MDM server components. As a rule of thumb, exceeding the following threshold might indicate hardware bottlenecks and should be investigated in more detail:

- Average value for CPU:IDLE < 30%.
- For Windows OS:
  MEMORY:KBPAGEDIN (KB/h) > 10% MEMORY:PHYSICALMEMORY (KB)
- For Unix OS:
  MEMORY:KBPAGEDOUT (KB/h) > 10% MEMORY:PHYSICALMEMORY (KB)

![CPU & Memory Utilization Table](image)
3.2 Component Workload Analysis with Wily Introscope

Compared to the E2E Workload Analysis, Wily Introscope provides non-aggregated performance and resource data in a finer granularity. The Wily Introscope can be used for the ad-hoc analysis of resource consumption and system behavior and the analysis of the health status of the system. It provides live and historical data in graphical form.

Performing the component specific workload analysis with Wily Introscope is applicable if the E2E Workload Analysis indicates that the root cause of performance issues is in the MDM Server and you would like to nail down the issue, or if you want to analyze the last minutes load on the MDM server as this information is not yet available in the E2E Workload Analysis.

Best Practices for investigation/rules of thumb:
- Get a system status overview via the dashboards
- Check the component specific dashboards

The Wily dashboards combine singular metrics display in complex synchronized charts. For example, SAP MDM – Overview combines client requests and resource allocation on one screen.

3.2.1 MDM Workload and Performance Metrics

MDM provides workload and performance metrics for:
- MDM servers (MDM Server, Import Server, Syndication Server)
- MDM client activities (Console, Data Manager, API calls)
- MDM imports and exports
- MDM inbound/outbound port tracking
- Resource utilization
- Repository metrics

Data is collected over time, depending on the log level settings (Wily Instrumentation Level Threshold) in the corresponding .ini files. For more information refer to chapter 4.1.5 MDM Server Performance Statistics.
3.2.2 SAP – MDM Overview

To start Wily Introscope:

• Choose *Introscope Workstation*

  ![Introscope Workstation - Detailed Selection](image)

• Start the *MDM Data Server Analysis*

  ![Introscope Workstation - Detailed Selection](image)
This starts the Wily Introscope Workstation. Select dashboard 1 SAP MDM Overview to start the analysis.

The SAP MDM Overview screen provides charts for analyzing the performance and workload of MDS, MDIS and MDSS. The charts are explained in detail in the following chapters.
The following overview shows schematically the different MDM dashboards and their content.

### MDM server Performance and Workload Overview

- **Client Requests – Avg. Response Time (ms)**
- **Client Requests – Number of Executions per time measurement interval**
- **Performance and Workload for Client Requests**
  - Client requests (ms) – Avg. Response Time (ms)
  - Client Requests – Number of Executions
- **Performance and Workload for DB**
  - DB – Avg. Response Time (ms) per DB Request
  - DB – Number of DB requests
- **Performance and Workload for Server/Repository Locks**
  - Server Repository Locks – Avg. Response Time (ms)
  - Server Repository Locks – Number of Lock Requests

### MDS and MDSS Performance and Workload Overview

- **Import/Export Steps – Avg. Response Time (ms)**
- **Number of Imports/Exports**
  - Details for Imports and Syndications
    - Import Server – Avg. Response Time per Import Step (ms)
    - Import Server – Number of Import Steps
  - Performance and Workload for Syndication server
    - Syndication Server – Avg. Resp. Time per Syndication Step (ms)
    - Syndication Server – Number of Syndication Steps

### Additional Dashboards for Analysis

- **Number of loaded Repositories**
  - Repository Details
    - Number of Main Table Entries per Repository
    - Number of Lookup Table Entries per Repository
    - Number of Qualified Keys per Repository

- **MDM Processes Memory Allocation**
  - Resource Allocation Details
    - Number of logged on Users
    - Running Repositories
    - Number of Sessions
    - Total Number of Main Table Entries
    - Number of Database Connections
    - Number of Workflows in Memory

- **MDM Processes CPU Utilization**
  - OS Overview
    - Host CPU utilization (%)
    - Memory Paging (kilo/sec)
    - LAN utilization in packages in/out per network device
    - Disk utilization in %

To adjust the charts or to obtain further information, be aware of the following settings:
**Help Function**

If you move the pointer near a graph, help information is provided:

Every help information will show the following four values \((\text{Value, Min, Max and Count})\). The relevant information is given with the \(\text{Value}\) field.

**Value** shows the average value within the measurement time interval.

The value 17.8K should be interpreted as follows:

\[
17.8 \text{ k} = \frac{17800 \text{ms}}{1024} = 17.38 \text{ sec}
\]
**Scale Options**

As the average response time or total number is too small to be shown in the chart, you can change the scale options.

Right-click on the chart and choose *Scale Options.*
**Show Minimum and Maximum**

Most of the charts show average values. It is also possible to show minimum and maximum values within the time measurement interval.

Right-click on the chart and choose *Show Minimum and Maximum*.

**Live and Historical Data**

Wily Introscope provides live and historical data in graphical form. This can be changed by adjusting the time range.
3.2.2.1 MDM Server Performance and Workload Overview

The first two charts on the SAP MDM Overview dashboard provide performance and workload overview for the MDM Server.

3.2.2.1.1 Client Requests – Avg. Response Time (ms)

The performance chart shows the average response time in ms per client request within a measurement time interval. The time interval usually is 15 seconds.
**Displaying Top 3**

Only the top three client requests with regard to the average response time will be shown. The chart will be adjusted dynamically if a new client request shows up with higher average response time.

Client requests are request response cycles:
- A client sends a request to a TCP port of the MDS. Each client request corresponds to a top-level protocol method, such as *modify field*.
- A listener thread scans all ports and assigns the request to a worker thread.
- Now the measurement starts.
- Requests are processed (for example *modify field, retrieve main table row,...*) and the response is sent back to client.
- At that point, the measurement ends.

Possible client requests are: Console, Data Manager, Java API, ABAP API, Import Manager/ Server, Syndication Manager/ Server.
Overview for MDM 5.5 SP06:

The following table displays how requests from the different clients will be displayed in Wily Introscope Workstation.

<table>
<thead>
<tr>
<th>Requestor</th>
<th>Will be reported as:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console</td>
<td>SAP MDM Console</td>
</tr>
<tr>
<td>Data Manager</td>
<td>SAP MDM Data Manager</td>
</tr>
<tr>
<td>Import Manager</td>
<td>SAP MDM Import Manager</td>
</tr>
<tr>
<td>Syndicator</td>
<td>SAP MDM Syndicator</td>
</tr>
<tr>
<td>Publisher</td>
<td>MDM Publisher</td>
</tr>
<tr>
<td>Workflow</td>
<td>SAP MDM Workflow</td>
</tr>
<tr>
<td>MDSS</td>
<td>SAP MDM Import Server</td>
</tr>
<tr>
<td></td>
<td>CatMgrServer</td>
</tr>
<tr>
<td></td>
<td>CMSrvServer</td>
</tr>
<tr>
<td>MDSS</td>
<td>SAP MDM Syndication Server</td>
</tr>
<tr>
<td></td>
<td>CatMgrServer</td>
</tr>
<tr>
<td></td>
<td>CMSrvServer</td>
</tr>
<tr>
<td></td>
<td>ImageSrv</td>
</tr>
<tr>
<td>MDLS</td>
<td>SAP MDM Layout Server</td>
</tr>
<tr>
<td></td>
<td>CatMgrServer</td>
</tr>
<tr>
<td></td>
<td>CMSrvServer</td>
</tr>
<tr>
<td>JAVA API</td>
<td>CMSrvServer</td>
</tr>
<tr>
<td></td>
<td>SAP MDM JAVA API</td>
</tr>
<tr>
<td>ABAP API</td>
<td>CL_MDM_PROVIDER_55_SP06_PL01</td>
</tr>
<tr>
<td></td>
<td>WEB</td>
</tr>
<tr>
<td>Portal</td>
<td>SAP MDM JAVA API</td>
</tr>
<tr>
<td>Trex</td>
<td>Trex</td>
</tr>
<tr>
<td>CCM3 Agent</td>
<td>Itsaim</td>
</tr>
<tr>
<td>CUX</td>
<td>MDSAdmin</td>
</tr>
<tr>
<td></td>
<td>team</td>
</tr>
<tr>
<td></td>
<td>&gt; clix commands like get version are reported under Networking MDSAdmin</td>
</tr>
<tr>
<td></td>
<td>&gt; clix commands like monitor activity are reported under Networking Itsaim</td>
</tr>
</tbody>
</table>

Example: MDSS

Most of the activities of the MDSS requests are reported via **SAP MDM Syndication Server**. But there are some further administrative tasks such as Log-In that are reported via the respective protocol. Therefore, further metrics like **CatMgrServer** or **CMSrvServer** come up when the syndication server is running.

**Remark:** All synchronous client requests will be shown. Asynchronous processes such as load repository or archive repository are planned to be reported as of MDM 7.1.
3.2.2.1.2 Client Requests – Number of Executions per Time Measurement Interval

The workload chart shows the number of executions for each measurement time interval for each client request. An execution is one execution of a client request. For example, if the number is 7 then seven client requests were executed within the measurement time interval.
3.2.2.1.3 Details for Master Data Server Performance

Pressing the *MDS Performance Details* button displays a dashboard with more detailed metrics for MDM performance.

The *SAP MDM - Master Data Server Performance* dashboard provides performance details for:

- Client requests
- Database
- Locks
3.2.2.1.3.1 Performance and Workload for Client Requests

Client Requests – Avg. Response Time (ms)
This chart shows the same metrics as the chart on the overview page as explained in chapter 4.2.2.1.1. The only difference is that the number of displayed metrics is not limited to three.

Client Requests – Number of Executions per measurement time interval
This chart shows the same metrics as the chart on the overview page as explained in chapter 4.2.2.1.3. The only difference is that the number of displayed metrics is not limited to three.
3.2.2.1.3.2 Performance and Workload for DB

The DB charts provide an overview of the DB activity triggered by MDM. It allows you to quickly identify whether performance issues in the MDS might be related to insufficient DB performance.

A DB request is an execution of an SQL statement from any request processed on the MDS that requires DB requests, such as load repository, change field, display attachment.

It will be shown how many DB requests (executions) were processed and how long they took (response times). It is not possible to see which statement was executed, how many records were returned, or which user or process triggered the statement.

With the default Wily instrumentation level 10, no DB requests will be reported. If you encounter periods of bad response times, temporarily increase the Wily instrumentation level to 15 as described in section 3.1.5. to get DB requests reported. As reporting the database requests is performance intensive for the MDS, it is advisable to keep Wily instrumentation level 15 only for a short time and to reduce it to 10 after the investigation is finished.

**DB – Avg. Response Time (ms) per DB Request**
The chart shows the average response in ms per DB request.

**DB – Number of DB requests per measurement time interval**
The chart shows the number of DB requests per measurement interval.
3.2.2.1.3.3 Performance and Workload for Server/Repository Locks

The lock requests charts provide an overview of the locks that were requested and gained during the MDM Server activity triggered by MDM. It allows you to quickly identify whether possible performance problems in the MDS might be related to locking problems.

Lock requests are locks to the repository or the server. Locks are going to be acquired and gained.

There are four kinds of locks:
- Server lock
- Repository lock
- Synchronization server lock
- Synchronization repository lock

and two states per lock:
- Read
- Write

Server and repository locks (Read/Write locks) should be acquired:
- in write mode for any operation modifying a repository or the entire server,
- in read mode only for any operation reading a repository or the entire server.

Synchronization server and synchronization repository lock are acquired:
- in read mode, by synchronization operations only
- in write mode, when synchronization should be locked out either for a specific repository or for the entire server:
  - Stopping the server (server-wide safeguard)
  - Unmounting, deleting, normalizing, truncating master change log for the affected repository
  - Unarchiving over, duplicating over and creating over - for the repository being overwritten
The locking mechanism in MDM works as follows:
When a client request is processed at first the lock is tried to be acquired. If a lock cannot be acquired within two seconds, the client is informed about the locking situation with an out-of-band message. The client can display an appropriate message to the user. For example:

![Wait while the MDM Server completes an extended operation.]

The server goes on with the request waiting for the lock again. A timeout can be configured in `mds.ini` separately for the read lock and the write lock. The default is infinite time.
The two settings in the `mds.ini` file are: `Socket Read Timeout` and `Socket Write Timeout`. They are set per server, not per repository.

Once the timeout occurs, the original client request is terminated with an error.

**Server/Repository Locks – Avg. Response Time (ms) per Lock request**

The chart shows the average response in ms per lock request. The time it takes to acquire a lock is measured.

**The chart will not show the request acquiring the lock, but the point of time when the request is finished.** This means in the example above the process of acquiring the lock started 40.4K ms before.
Server/Repository Locks – Number of Lock Requests

The chart shows the number of lock requests per measurement interval and how long it took to set the lock.

3.2.2.1.3.4 Recommendation/KPI

If you see that the average time to acquire a lock is greater than one second over a longer period of time, this indicates a locking issue.
3.2.2.1.3.5 Examples

**Mass transaction in MDM Data Manager and parallel user action**

A user performs a mass change of records via the MDM Data Manager. This will require read locks (1) for selecting the records and write locks (2) for changing the records on the repository.

Now a second user tries to acquire a read lock (3) on the same repository. Due to the write locks of the first operation, the average response time for the second user will increase until some threshold is reached or the request is cancelled.

Keep in mind that this chart *Number of Locks* shows when the request *Acquire Lock* is finished, and not the duration of acquiring the lock.
Import server running, is scanning ports but no import takes place

The Import Server is running, it performs a port scan every minute. In this example each port scan acquires 51 repository read locks and 10 server read locks.

Why are there so many repository read locks?

There are several client requests while importing data, such as GetRegions, GetAgencies and so on. Each of these requests requires a repository read lock.

Why are there server read locks?

There are a couple of client requests per protocol CmSrv, like Login, GetListofRepositories, GetCatalogStatus. Such requests require the server read lock.
### 3.2.2.1.4 Details for Master Data Server Activity Overview

Pressing the *MDS Activity Overview* button displays the MDM server activity overview.

The **SAP MDM - Master Data Activity Overview** dashboard provides an overview of the waiting and active requests as well as the number of write and read locks in the selected time interval.

The **SAP MDM - Master Data Activity Overview** dashboard provides an overview of the waiting and active requests as well as the number of write and read locks in the selected time interval.
The difference between active and waiting requests is explained by the following chart:

All MDM requests are placed in a single queue. Whereas read requests can be executed in parallel, a write request will require exclusive access to the MDM repository. If a write operation is the next task in the request queue, the execution will wait until previous read requests have been worked off. Then the write operation will be initiated and subsequent operations have to wait until it is finished.
Number of Waiting and Active Requests in MDS

The chart shows the number of active and waiting Requests in MDS in a detailed level. Keep in mind that the chart shows the requests active at the selected time frame and not the finished requests as in the charts before.

Waiting requests: Requests currently waiting for a resource for example to acquire a lock
- Waiting for Repository Read Lock
- Waiting for Repository Write Lock
- Waiting for Server Read Lock
- Waiting for Server Write Lock
- Waiting for Synchronization Repository Read Lock
- Waiting for Synchronization Repository Write Lock
- Waiting for Synchronization Server Read Lock
- Waiting for Synchronization Server Write Lock

Active requests: The number of requests currently running in the MDS for example modify field, import,…

Number of Locks in MDS
The chart shows the number of locks in MDS. A lock is either
- a shared read lock:
  - Repository Read Lock Gained
  - Server Read Lock Gained
  - Synchronization Repository Read Lock Gained
  - Synchronization Server Read Lock Gained
- or an exclusive write lock:
  - Repository Write Lock Gained
  - Server Write Lock Gained
  - Synchronization Repository Write Lock Gained
  - Synchronization Server Write Lock Gained

### 3.2.2.1.4.1 Examples

**Import**

The example shows an import of a file containing 5,000 records using the Import Server. This requires a repository read lock (1) first and then, for importing the data to the repository, write locks (2) are required. Here the write lock lasted for about 2.5 minutes. During this time, no other access to the repository is possible.
Mass transaction in MDM Data Manager

A user performs a mass change of records using the Data Manager (1). This requires repository write locks (2) for changing the records in the repository.
Mass transaction in MDM Data Manager and parallel user action:

A user performs a mass change of records using the Data Manager (1). This requires repository write locks (2) for changing the records in the repository.

Now a second user tries to acquire a repository read lock (3) on the same repository. Due to the write locks of the first operation, the average response to gain the lock increases.
3.2.2.2 MDIS and MDSS Performance and Workload Overview

The charts in the second section of the SAP MDM Overview dashboard provide a performance and workload overview for the Import Server and Syndication Server.

In the overview screen, not all MDIS and MDSS specific steps are displayed, only the import (`ximporter`) and export steps (`Export`):

- **Import**
After the import or export is finished, no activity is shown.
3.2.2.2.1 Details for Imports and Syndications
Pressing the MDIS/MDSS Performance Details button displays a dashboard with more detailed metrics for import and syndication performance.
3.2.2.1.1 Performance and Workload for Import Server

3.2.2.1.1.1 Import Server – Avg. Response Time per Import Step (ms)

The charts show the average response in ms per import step.

- **Scanning ports**: the time it takes to scan the ports to check if any file is ready to be imported
- **xStructural transformation**: the time it takes to perform the structure and value transformation for a chunk
- **xImporter**: the time it takes to import a chunk into the repository

**Note**: The charts report Import Server port scans also for unloaded repositories. The Import Server must retrieve the repository status at least to decide whether to go on with the ports. Therefore there is at least one call to the server. The time spent is added to the scan ports metrics.
3.2.2.1.1.2 Import Server – Number of Import Steps per Measurement Time Interval

The charts show the number of import steps when the step is finished.

- **Scanning ports**: Executions: number of port scans
- **xStructural transformations**: Number of transformed chunks
- **xImporter Number**: Number of imported chunks
3.2.2.1.2 Examples
Importing a file with 7,000 records based on a chunk size definition of 3,000 results in three chunks to be imported.

```
[GLOBAL]
String Resource Dir=C:\PROGRA~1\SAPMDM-2.5\IMPORT-2\LangStrings\
Log Dir=C:\PROGRA~1\SAPMDM-2.5\IMPORT-2\Logs
Server=P127015
BLD Registration=0
SAP Req Utility Path=\
Wily Instrumentation=False
Wily Instrumentation Level=Threshold=15
Interval=60
MapScanTopToBottom=0
Verbose=0

[Business_Partner_P127015\MSQL_1_6_7_4_3]
Chunk Size=3000
No. Of Chunks Processed In Parallel=5
Login=Admin
Password=DESK&SAPOLOG09K?
Log protocol transactions=False
```
3.2.2.1.3 Performance and Workload for Syndication Server

3.2.2.1.3.1 Syndication Server – Avg. Response Time per Syndication Step (ms)

The charts show the average response time in ms per export step.
3.2.2.1.3.2 Syndication Server – Number of Syndication Steps

The charts show the number of export steps when the step is finished.
3.2.2.2 Details for Import and Syndication Activities
Pressing the **MDIS/ MDSS Activity Overview** button displays a dashboard with more detailed metrics for Import Server and Syndication activities.
3.2.2.2.1 Import and Syndication Activity Overview

The following charts of the activity overview display currently running (Active Counter) import and syndication steps.

3.2.2.2.1.1 Active Import Steps

The chart shows the number of active import steps:

- **Scanning Ports**
- **Structural Transformation**
- **Import**
3.2.2.2.1.2 Active Syndication Steps

The chart shows the number of active export steps:

- Query Execution
- Export
- Copy to Memory
- Copy to Port
3.2.2.2.2 Import/ Syndication Backlog

3.2.2.2.2.1 Files in Inbound Ports

The chart shows the number of files to be imported in the Ready folder of the correspondent inbound port.

If the import fails, the charts will show the number of files in the exception folder.
### 3.2.2.2.2.2.2 Files in Outbound Ports

The chart shows the number of files to be exported in the *Ready* folder of the correspondent outbound port.

**Remark:** This functionality will be supported with MDM 5.5 SP06 Patch 02.

### 3.2.2.2.2.3 Recommendation/ KPI

- The number of files in the *Ready* folder should not be constantly greater than 0.
- If you detect a file in the exception folder, check the reason for the import issue.
3.2.2.3 Additional Dashboards for Analysis

The charts in the third section of the **SAP MDM Overview** dashboard provide information about repositories and operating system data.

### 3.2.2.3.1 Number of Loaded Repositories

The chart can be used to check the status of the repositories.
3.2.2.3.1.1 Repository Details

The *Repository Details* button displays three further charts providing detailed information per repository.

<table>
<thead>
<tr>
<th>Number of Main Table Entries per Repository</th>
<th>Number of Lookup Table Entries per Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="chart1.png" alt="Chart" /></td>
<td><img src="chart2.png" alt="Chart" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Qualified Links per Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="chart3.png" alt="Chart" /></td>
</tr>
</tbody>
</table>

The charts show the
- Number of main table entries per repository
- Number of lookup table entries per repository
- Number of qualified links per repository.

These metrics are collected only when the repository is loaded and after that once a day.
3.2.2.3.2 MDM Processes Memory Allocation

The chart shows the memory allocation of the
- Master Data Server
- Import Server
- Syndication Server
- Layout Server
measured at operating system level.
3.2.2.3.2.1 Resource Allocation Details

The Resource Allocation Details button displays six additional charts providing information about resource allocated within the MDM Server.

3.2.2.3.2.1.1 Number of Logged-on Users

This chart shows the number of the following groups of users:

- Repository administrator users (administrating the repository)
- Server administrators users (administrating the server)
- Repository users (regular users)

Note: If the user name is Admin, this does not mean that you are logged on as an administrator. It depends on the client application that is used. Admin logon with the MDM Console is an repository administrator logon and the same user Admin logged on with MDM Data Manager is a regular user logon.
3.2.3.2.1.2 Number of Workflows in Memory

The chart shows the number of pre-launched and launched workflow jobs in memory.

**Example:**

Pre-launched workflows can be the following:

- launched

- completed

- deleted

The workflow jobs are kept in memory even if the status is *completed*. Delete the workflow items on a regular basis.
3.2.2.3 MDM Processes CPU Utilization

The chart shows the server CPU utilizations of the following processes on the correspondent software component:

- mds.exe
- mdis.exe
- mdss.exe
- mdls.exe

Here you can easily identify whether one of the server components consumes a high proportion of the CPU resources available on the host.

3.2.2.3.1 OS Overview

The OS overview shows the CPU utilization, disk utilization and memory paging of the host in the selected time frame. It allows you to detect time frames during which hardware bottlenecks occur.
**Host CPU utilization (%)**

This chart shows the resource utilization of the physical hosts running the MDM server components. The total CPU utilization of the server in % should not exceed 70-80% over a long time.

**Memory Paging (kB/sec)**

This chart shows the OS paging. The paging should not exceed 10% of the server memory/hour.
For more information, double-click the chart.

**LAN utilization in packages in/out per network device**

This chart shows the network packages that were transferred.
For more information, double-click the chart.

**Disk utilization in %**

This chart shows the disk utilization. The disk utilization should not exceed 50% over a long time.
For more information, double-click the chart.
4 Configuration Reporting for MDM

The Configuration section provides the E2E Change Analysis and Configuration and File Reporting for MDM. Both are applicable if the system behaves differently after a certain date or if some changes were made and you want to find out which technical parameters were changed.

4.1 E2E Change Analysis

The Overview screen provides an overview of the number of changes to MDM technical configuration files. (mds.ini, mdss.ini, mdis.ini).

Due to limitations in the analysis of .ini files (SAP Note 1111449), the number of changes is usually reported too high. When inserting/deleting a line in an .ini file, all shifted lines are detected as modified. Nevertheless, the Overview screen allows you to check whether configuration changes were applied on a certain day or not.
The next tab provides a **Summary** and *Details View*. The **Summary** view provides an overview of the MDM Server component (MDS, MDIS, MDSS) where technical parameter changes were applied.
4.2 Configuration and File Reporting

Configuration and file reporting allows you to display the MDM configuration files, to quickly identify modified lines in the configuration files, and also to compare different versions of the configuration files.

Select a timestamp for which you want to analyze the configuration. Choose Load if you want to analyze one configuration. Choose Compare to compare the same configuration files at different point in time or if you want to compare the configuration of a production and a QA system.

To display the content of the MDM configuration, click on the respective row. For example, click on ConfigStore ./mds.ini if you want to display the configuration of the mds.ini file.
The screenshot below shows the current content in the *mds.ini* file. The symbol in line 3 indicates that this line was changed while the other lines still have their initial value.
Clicking the *Attribute History* displays the history for the current line.

<table>
<thead>
<tr>
<th>Status</th>
<th>Date</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>19/10/2007</td>
<td>Accelerator Dir=C:\Program Files (x86)\SAP MDM 5.5\Server\Accelerator</td>
</tr>
<tr>
<td></td>
<td>24/10/2007</td>
<td>Accelerator Dir=C:\Program Files (x86)\SAP MDM 5.5\Server\Accelerators</td>
</tr>
<tr>
<td></td>
<td>25/10/2007</td>
<td>Accelerator Dir=C:\Program Files (x86)\SAP MDM 5.5\Server\Accelerator</td>
</tr>
</tbody>
</table>

In this example, the parameter file was first created on Oct. 19, 2007. The line was changed on Oct. 24 and on Oct. 25, 2007.
5 Root Cause Analysis Roadmaps

The following roadmaps provide guidelines for analyzing performance and functional issues.

Root cause analysis (cross-component)

- Is it a performance problem?
  - End-to-end workload analysis

- Is it a functional problem?
  - Incorrect results or error messages in application
    - Log analysis
  - Unexpected termination of MDM Client components (Data Manager, Import Manager, ...)
    - Client component analysis with AppSight
5.1 End-to-end Workload Analysis

- System components with unusually high response time or throughput
  - Component workload analysis
- Single transaction or client or network not performing
  - ABAP API - MDM integration
    - ABAP: ABAP trace, SQL trace (→ Explained in course E2E100)
    - MDM Server Performance Log
  - Java API - MDM integration
    - Java: End-to-end trace (→ Explained in course E2E100)
    - MDM server Performance Log
  - MDM Data Manager usage
    - AppSight Analysis of Data Manager (→ Explained in course E2E100)
    - MDM Server Performance Log
5.2 Component Workload Analysis

Component workload analysis

- Is it a Java component?
  - Java component workload analysis (→ Explained in course E2E100)

- Is it an ABAP component?
  - ABAP component workload analysis (→ Explained in course E2E100)

- Is it a MDM component?
  - MDM workload analysis with Wily Introscope
5.3 MDM Workload Analysis with Wily Introscope

MDM workload analysis with Wily Introscope

Average response time per user click not acceptable

Operating system analysis

- CPU time of the MDM Server processes is unexpectedly high
  - Analyze CPU consumption
- High disk utilization
  - Analyze disk utilization

SAP MDM analysis

- High memory allocation growth of MDM Server process
  - MDM resource consumption analysis
- Does memory growth correspond to increase in repository table entries
  - Repository analysis
- Does memory growth correspond to increase in Workflows in memory
  - Workflow reorganization

Requests are waiting for locks

MDM locking analysis

- Long running imports or syndications running at the same time
  - Check import /syndication performance / eventually reschedule
- User requests holding locks
  - Determine lock holder with MDM activity monitor

Long running SQL statements

DB Performance analysis in DBA COCKPIT

Long method execution

Detailed analysis of executed action / customer message
6 Appendix

6.1 Links

For more information, refer to the following links:

- How To...setup SMD for MDM Root Cause Analysis – MDM 5.5 SP06 (planned to be released March 2008)
  service.sap.com/installMDM

- SAP Solution Manager Root Cause Analysis: http://service.sap.com/diagnostics

- How to guides: www.sdn.sap.com/irj/sdn/howtoguides

- SAP Solution Manager Installation:
  service.sap.com/solutionmanager → Installation Guides → Solution Manager 4.0.

- MDM 5.5 Documentation Center:
  service.sap.com/installMDM