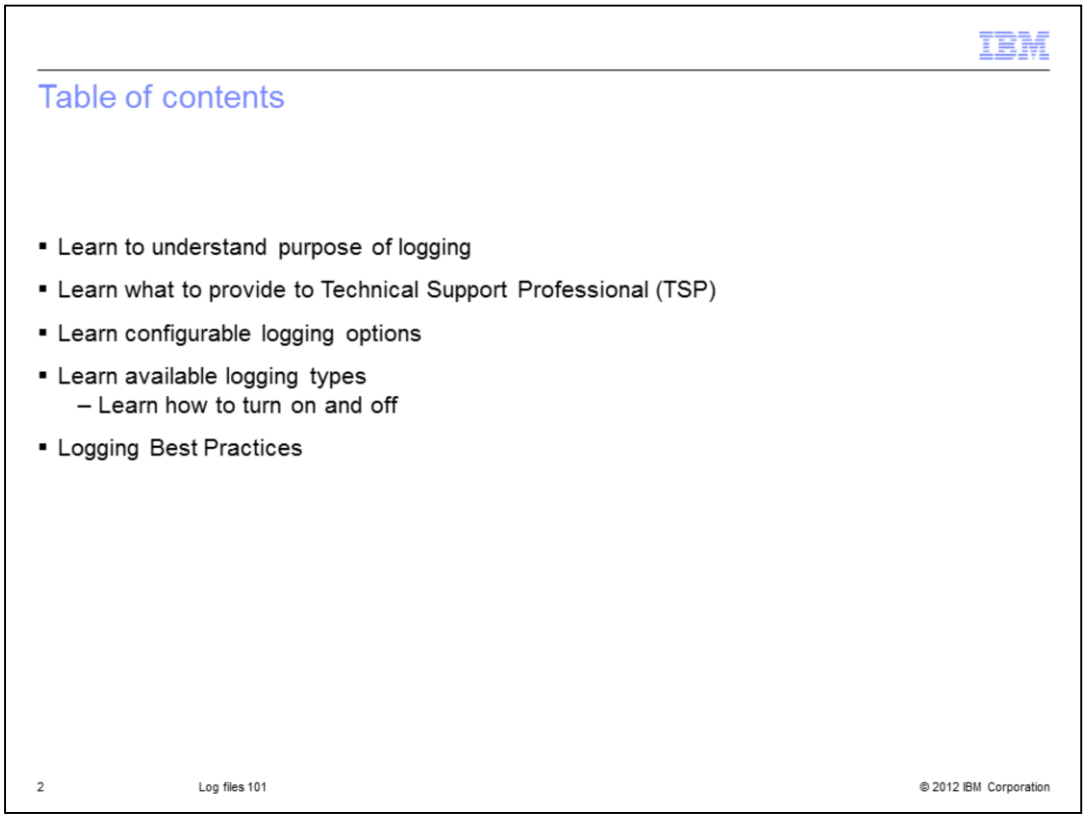
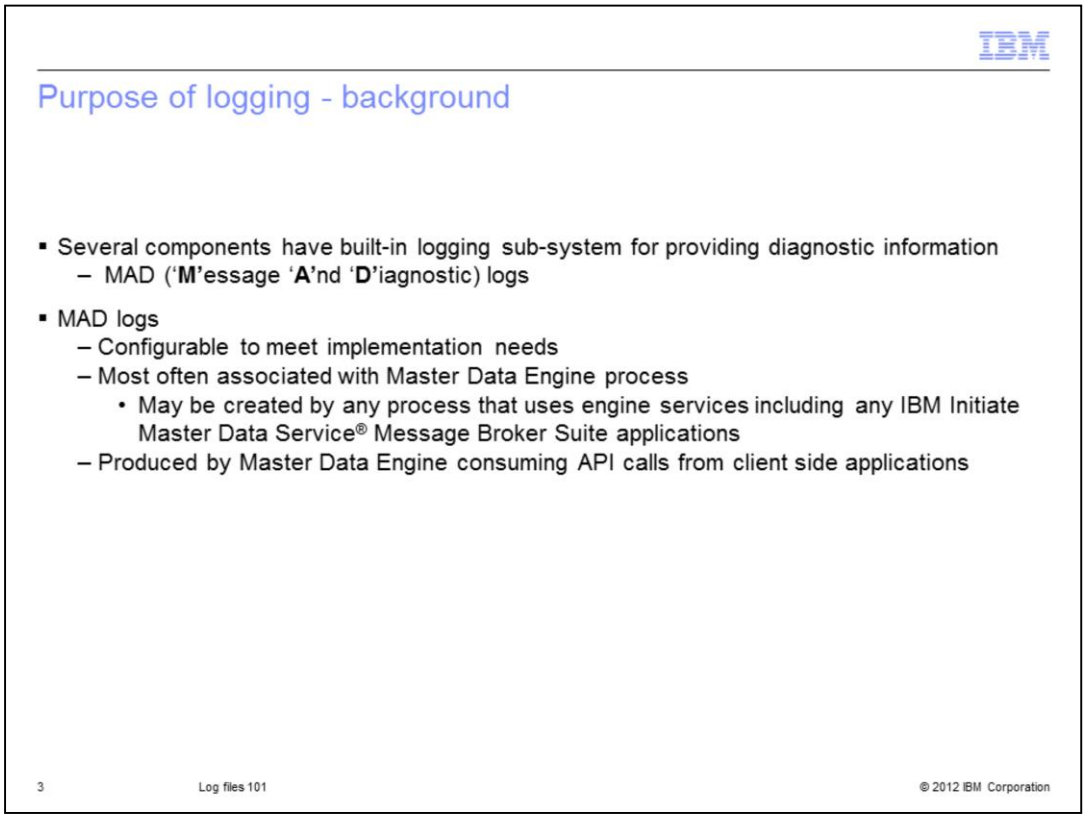
This is the IBM Initiate® “Log Files 101” training presentation.



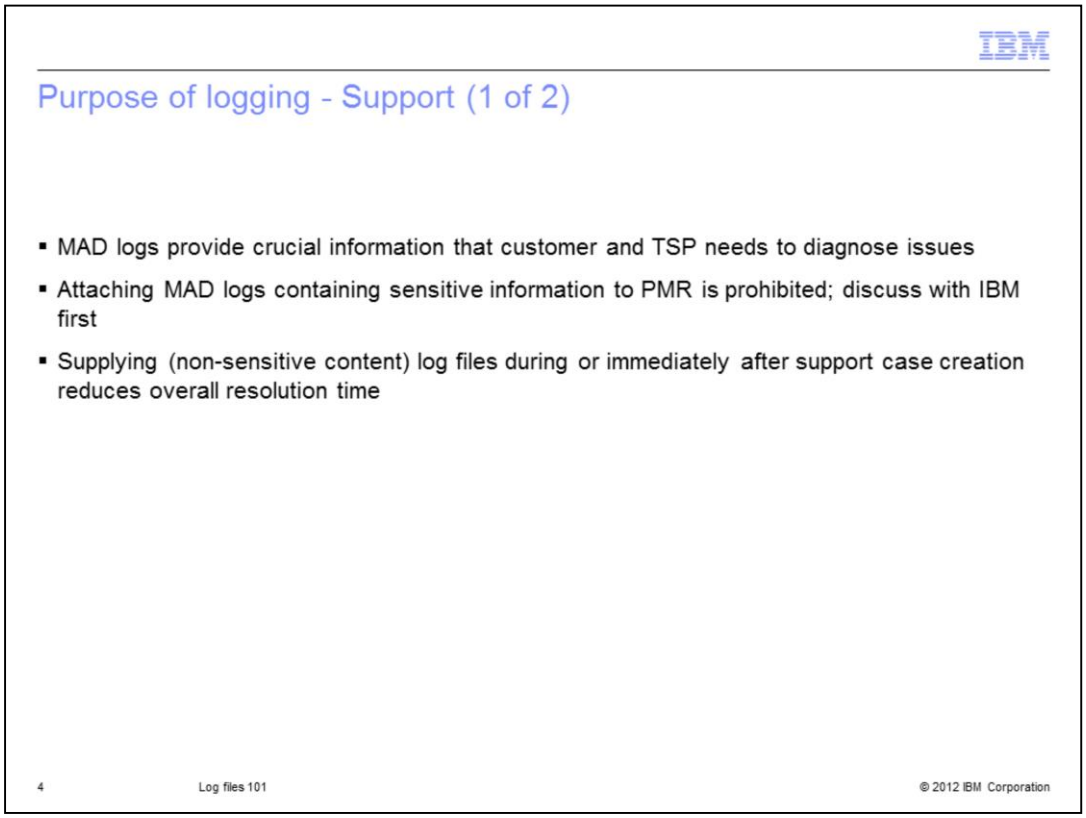
The objective of this presentation is to help you understand, in general, the purpose of logging. You will also learn where to find the log files and what logging information the IBM Technical Support Professionals, referred to as TSPs, need to know to troubleshoot an issue. You will see some of the options that are configurable for logging and then review some of the different logging types. Finally, the training will conclude with a review of IBM Initiate best practices and log setting recommendations for your Master Data Engine.



It is important to get a basic understanding on the purpose of logging. Several IBM Initiate components have a built-in logging sub-system that provide useful diagnostic information. They are called “MAD logs”, which stands for message and diagnostic logs.

These log files are configurable to meet any implementation need. They are most often associated with the Master Data Engine process, but can be created by any process that uses the Master Data Engine service in a C or C++ environment. This includes any of the Message Broker Suite applications, such as the Inbound Message Reader and Inbound Message Broker. MAD logs are also produced by the Master Data Engine when external API calls are made from client side applications.

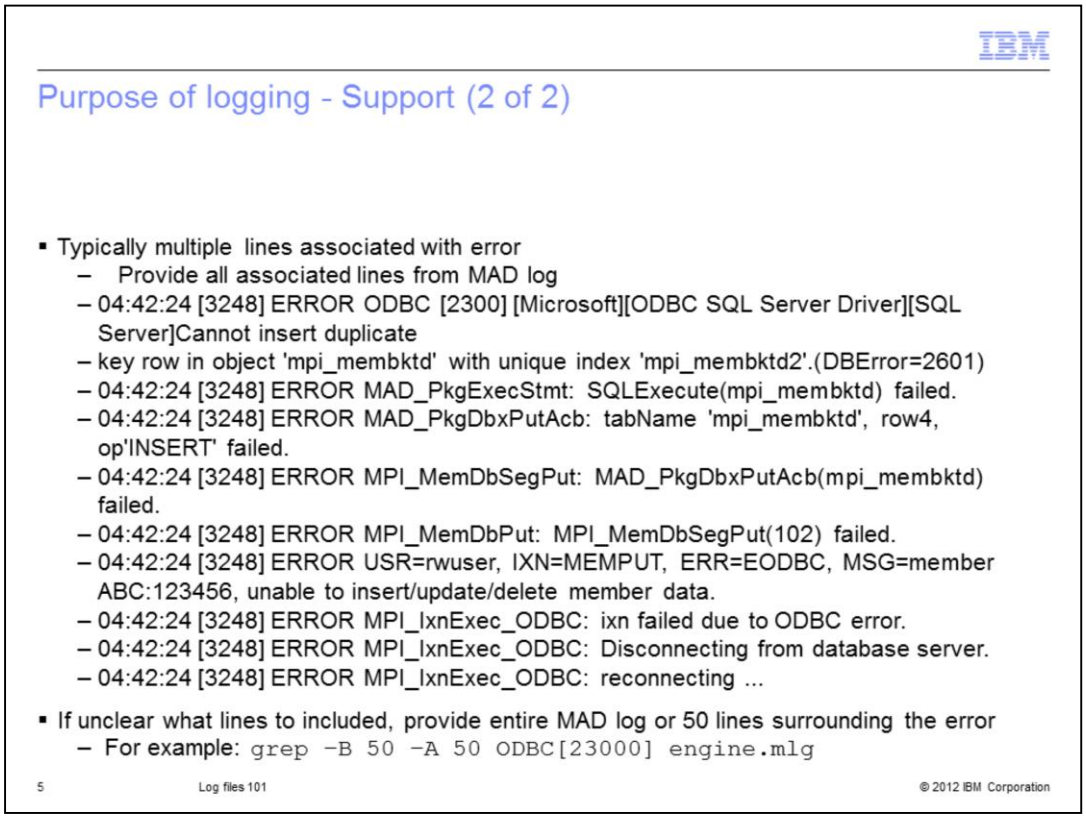
It is important to note that while client applications will not have MAD log entries themselves, the resulting interactions with the Master Data Engine indirectly produces MAD log entries.



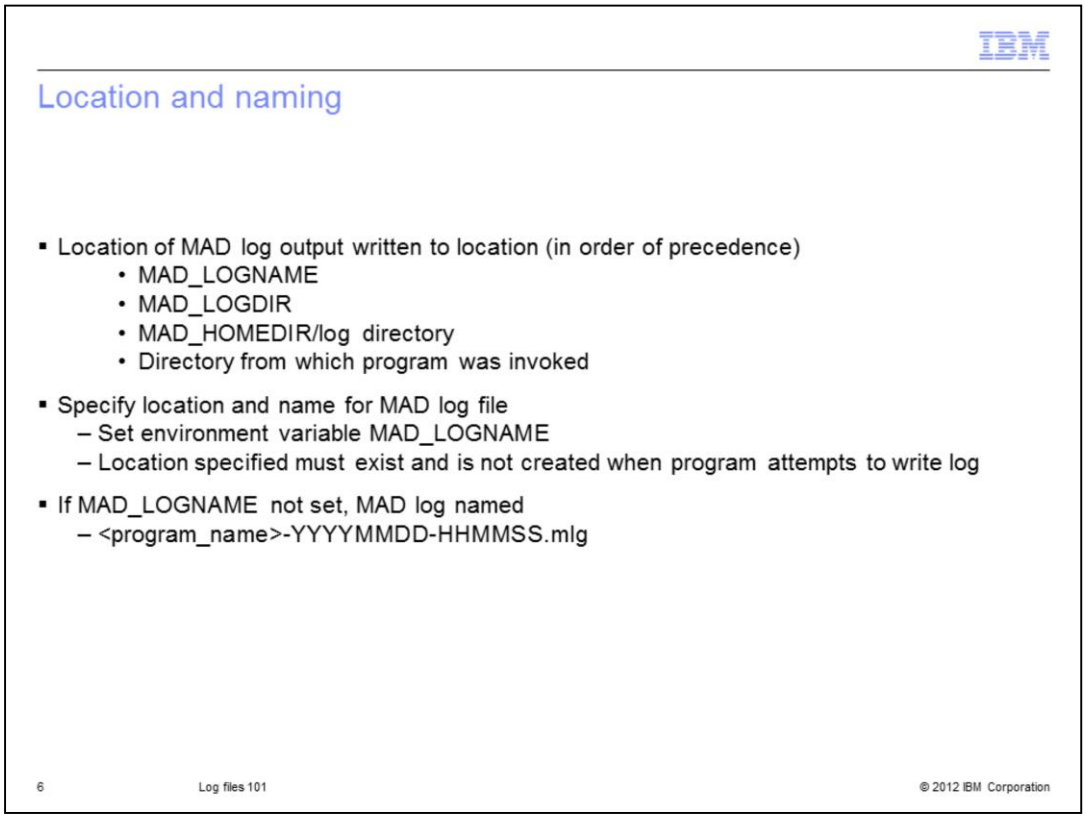
MAD logs provide crucial information that both the users and TSPs require in order to diagnose and correct issues or problems with the product.

Attaching MAD logs to the support case is prohibited since they may contain protected health information, referred to as PHI. Engage IBM Support to discuss the transmission of potentially sensitive information to IBM.

When TSPs receive a service request, the first thing they will want to see is your MAD logs. As long as the log file does not contain sensitive information, attach it to your service request at the time of submission in order to speed up the support process. If the MAD log is large, compress it in a zipped or gzipped format and then attach it. Compressing the log files can save a lot of space and time when uploading.



Next, you need to know what should be included from the MAD log. The TSPs like to see all the log entries associated with an error or other issue that you may have a question about. Often when an error occurs, there are multiple lines. This example depicts an ODBC error with approximately ten lines, all of which are reporting different information for that error. If at all possible, provide all the lines associated with an error. If it is unclear which lines should be included, either include the entire MAD log file, or else 50 lines before and 50 lines after the error. For example, you may use the grep command as displayed on this slide, to get the previous 50 and the trailing 50 lines after the error in your engine log.

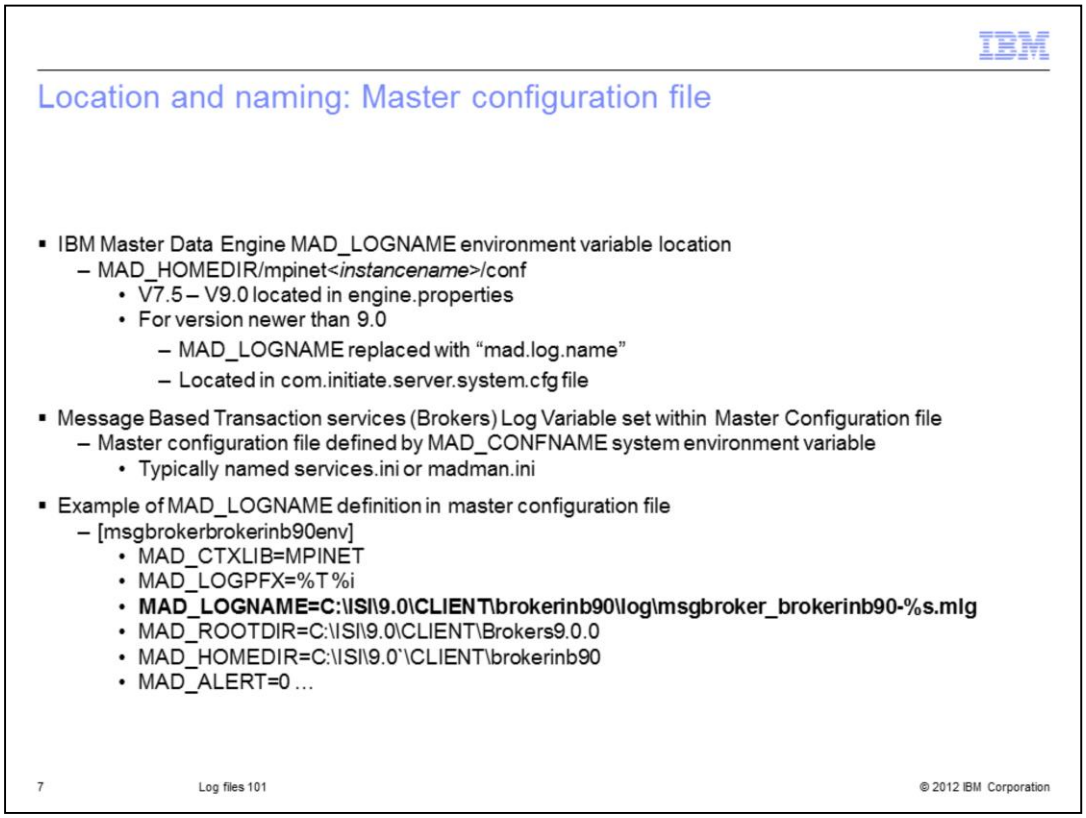


In addition to knowing what Support needs from the MAD logs, you also need to know where you can find these logs.

The location is indicated by one of a few different environment variables. The Master Data Engine looks for the presence of these variables and stops when it has successfully found one. They are listed on this slide in order of precedence.

First is the MAD_LOGNAME environment variable. If set, it will point to both the absolute path and name of the log file. The location of the path must already exist and is not created automatically. If MAD_LOGNAME is not set, it will look for the MAD_LOGDIR environment variable. This variable contains the directory where the log files reside and will use the default naming convention. If MAD_LOGDIR is not set, it will automatically go to the environment home directory and log subdirectory. Finally, if none of these environment variables are set, the MADLOG is written to the directory where the program is invoked from. These environment variables are used to help you organize where these log files are going to be stored and written to.

If the MAD_LOGNAME is not set, there is a default naming convention used. The convention is the program name followed by a date and time stamp with the extension of .mlg, which stands for mad log.

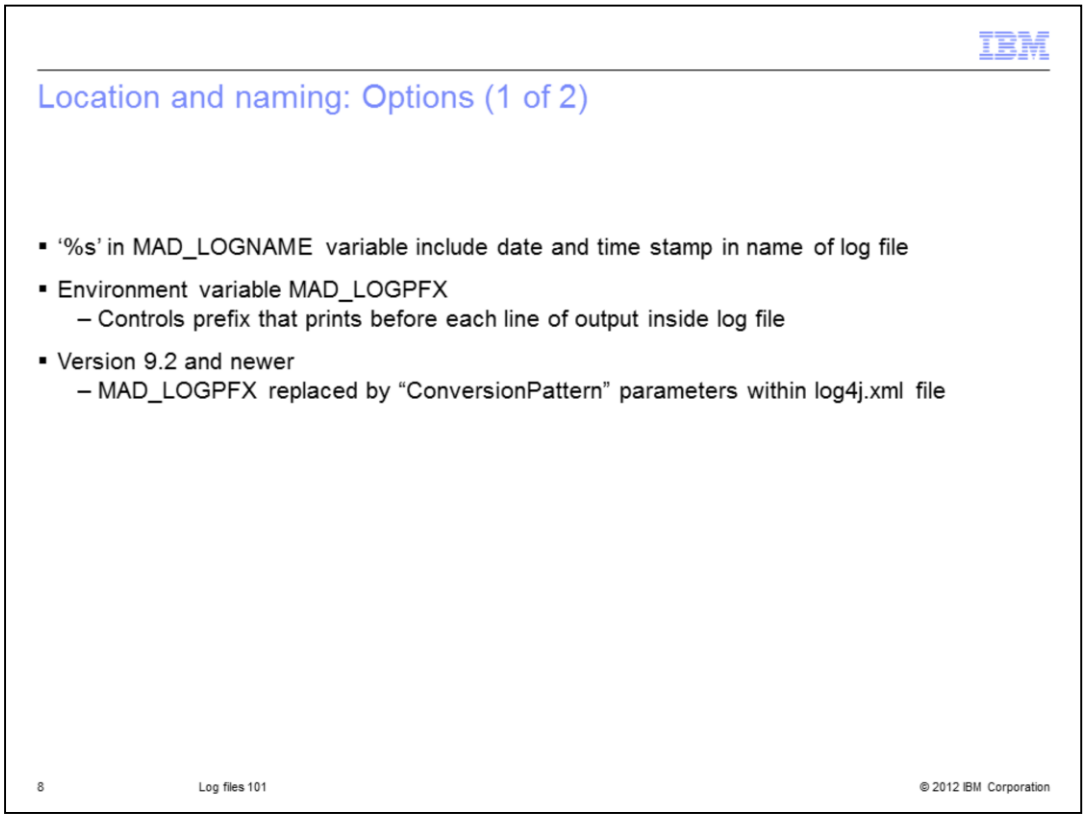


Another important aspect of the logging variables is how they are created and initialized. The environment variables are set in various places and the location depends on the application and version.

For Master Data Engine versions 7.5 through 9.0, the MAD_LOGNAME environment variable is set in the engine.properties. For versions newer than 9.0, the MAD_LOGNAME environment variable can be found within the com.initiate.server.system.cfg file.

For the message based transaction services, the log environment variables are set in the master configuration file. The master configuration is typically named "services.ini" or "madman.ini", but can have any name so long as it is defined or pointed to by the MAD_CONFNAME system environment variable.

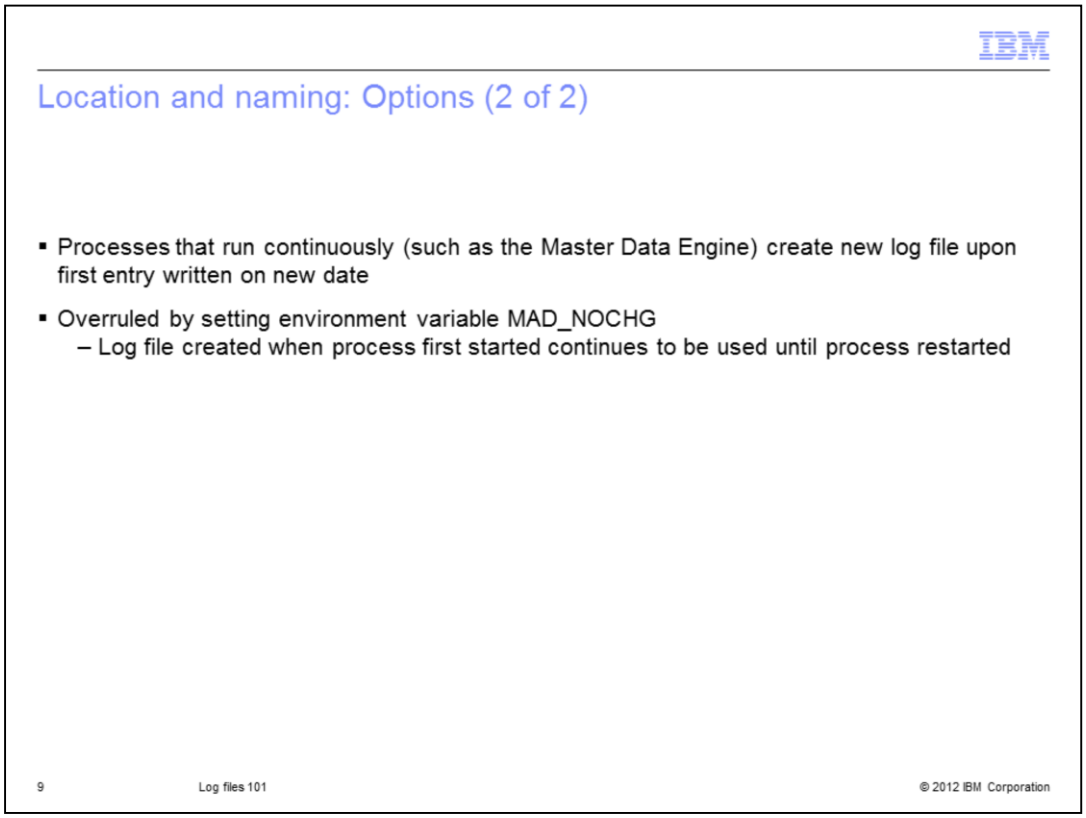
The example illustrated on this slide is for a version 9.0 IBM Initiate Inbound Broker running on a Windows® platform. The snippet is located within the master configuration file.



There are some options available for the log's location and naming convention. There is a percent sign that you will often see in the MAD_LOGNAME environment variable. This symbol tells the Master Data Engine to include a date and time stamp in the log file's nomenclature.

There is also an option on the prefix to be used. The prefix is going to be in the actual log file entries and is defined by the MAD_LOGPFX environment variable. The PFX in MAD_LOGPFX represents "prefix".

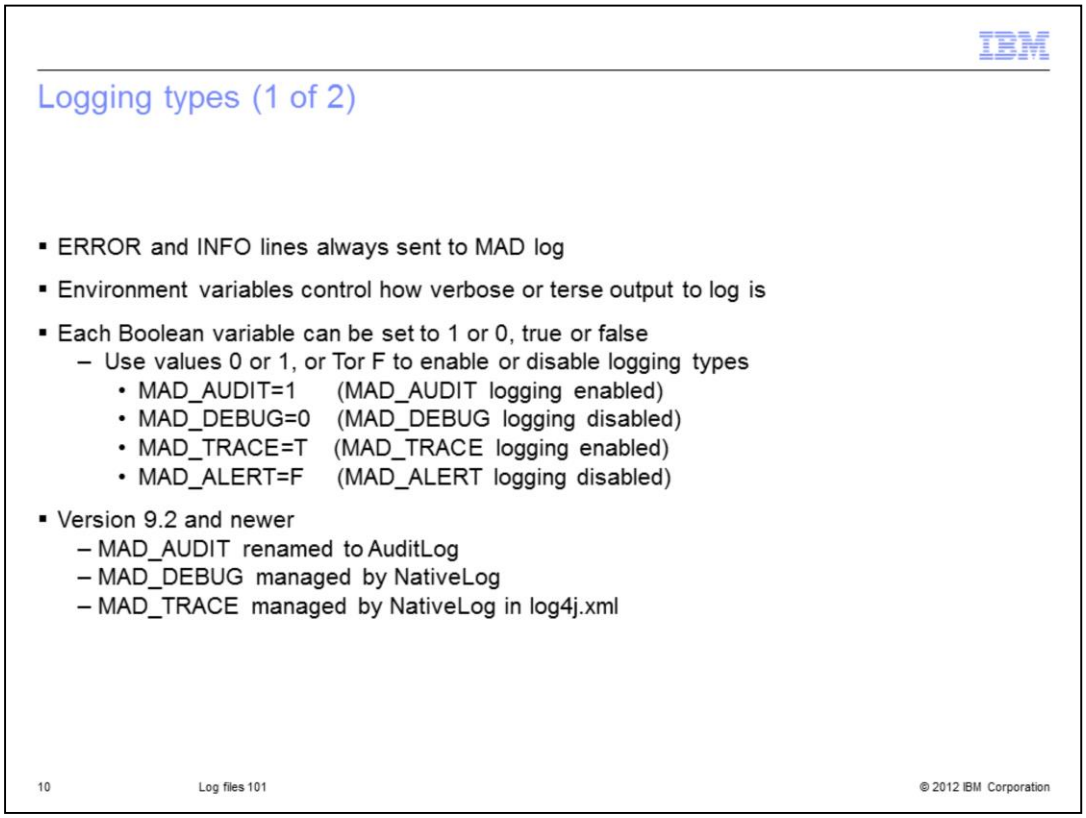
For version 9.2 and newer, MAD_LOGPFX was replaced by "ConversionPattern" parameters within the log4j.xml file.



Another important logging option has to do with processes that run on a continuous basis.

The Master Data Engine is a good example of a process that typically runs 24 hours a day, seven days a week. A new log file is created when triggered by the first log entry after a new day. For example, if there is a log event that triggers a log entry at 12:01AM, then a new log file is created at that point in time.

If, for any reason, this new day log creation is not the behavior you want, use the MAD_NOCHG (mad no change) environment variable. This indicates that the log file should continue to write to the same file until the process has been restarted.

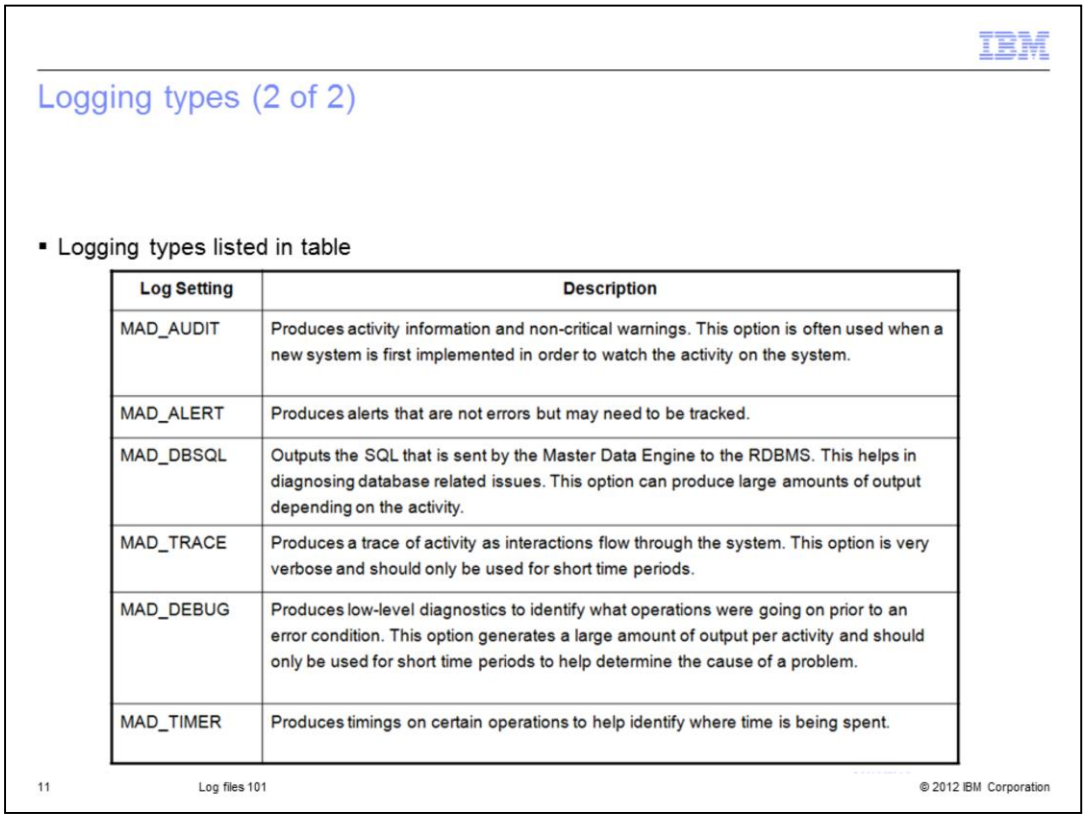


As for the content that is written in the log file, there are different logging types that can be enabled or disabled. Each of which control the type and the verbosity of the information being written in the log file. The two logging types that are always written, regardless of what the logging type is set to, are “ERROR” and “INFO” entries.

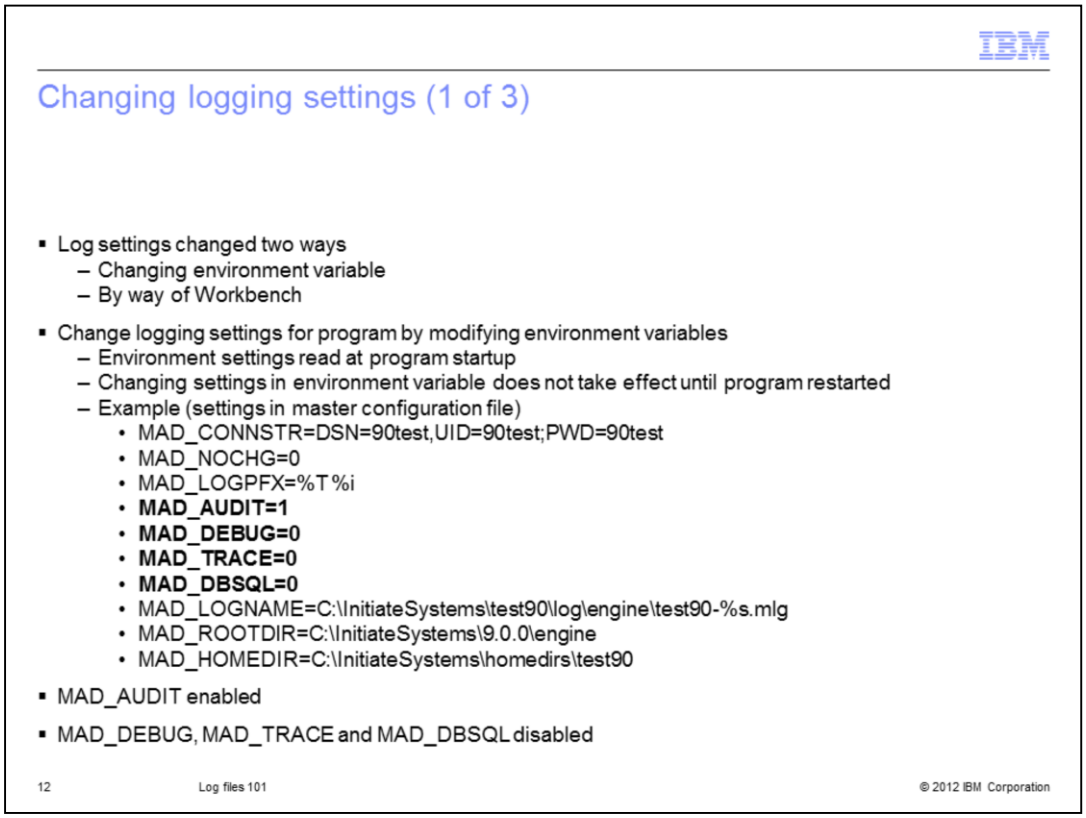
Other environment variables can be set to increase the amount of the logging written to the MAD log. Each Boolean variable can be set to 1 (for true) to enable, or 0 (for false) to disable the logging types.

An example of a logging type is if in the master configuration file you have MAD_AUDIT=1, then that means auditing or the audit logging type is enabled. Another example is if you have MAD_DEBUG=0. That means that the MAD debug logging type is disabled, and so forth.

For versions 9.2 and newer, log4j is used instead. MAD_AUDIT is renamed to AuditLog, and MAD_DEBUG and MAD_TRACE are managed by NativeLog in log4j.xml.



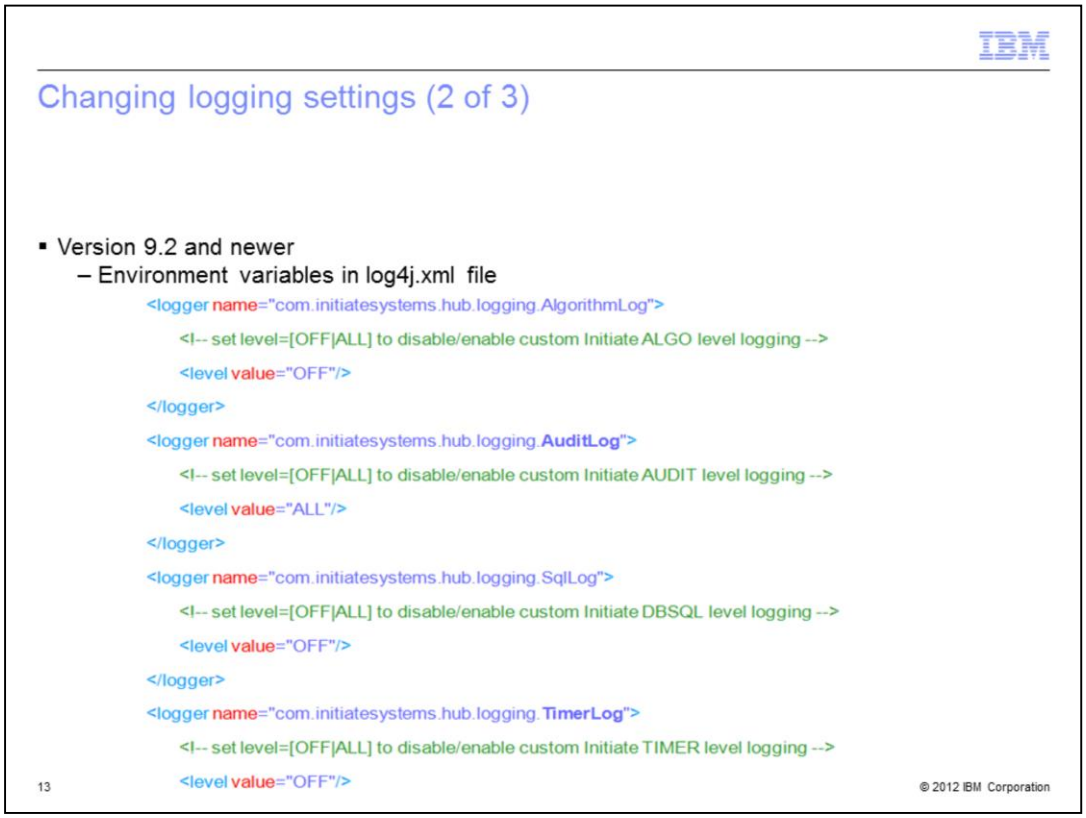
This slide displays a copy of the table found in the Master Data Engine Installation Guide documentation. It explains what you can expect to see in your log file if you enable any of these logging types.



Log settings can be changed two different ways – by way of the environment variable, or by way of the IBM Initiate Workbench client.

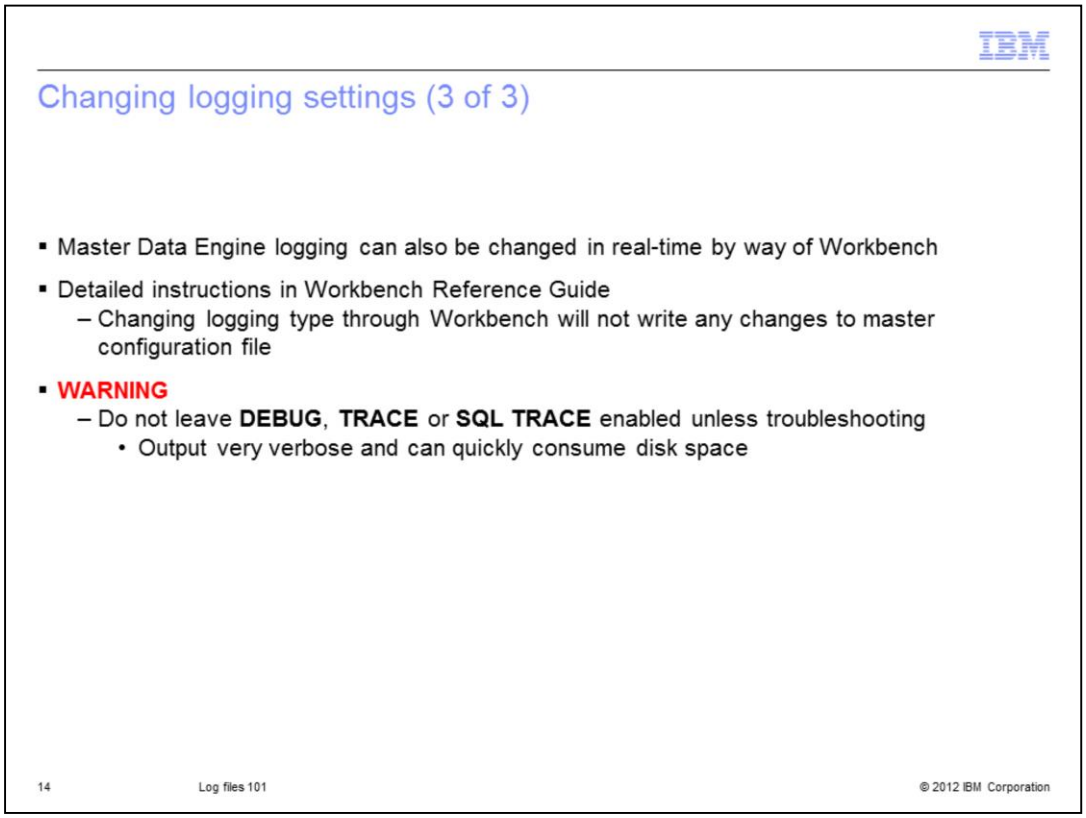
If you make changes to the master configuration file, those changes will not take effect until the next startup of that process. So, changing the environment variable after the process has already been started will not affect the logging type until the process has been restarted and the environment variables are read at startup.

In this example, MAD_AUDIT has been turned on (or enabled) and MAD_DEBUG, MAD_TRACE and MAD_DBSQL is disabled by being set to zero.



For version 9.2 and newer, log4j is employed. In order to change the log settings, the log4j.xml file needs to be changed.

This example displays a snippet from a log4j.xml file. Here you can see that in order to enable a log type you need to set the variable equal to “ALL”. To disable, you need to set it to “OFF”.

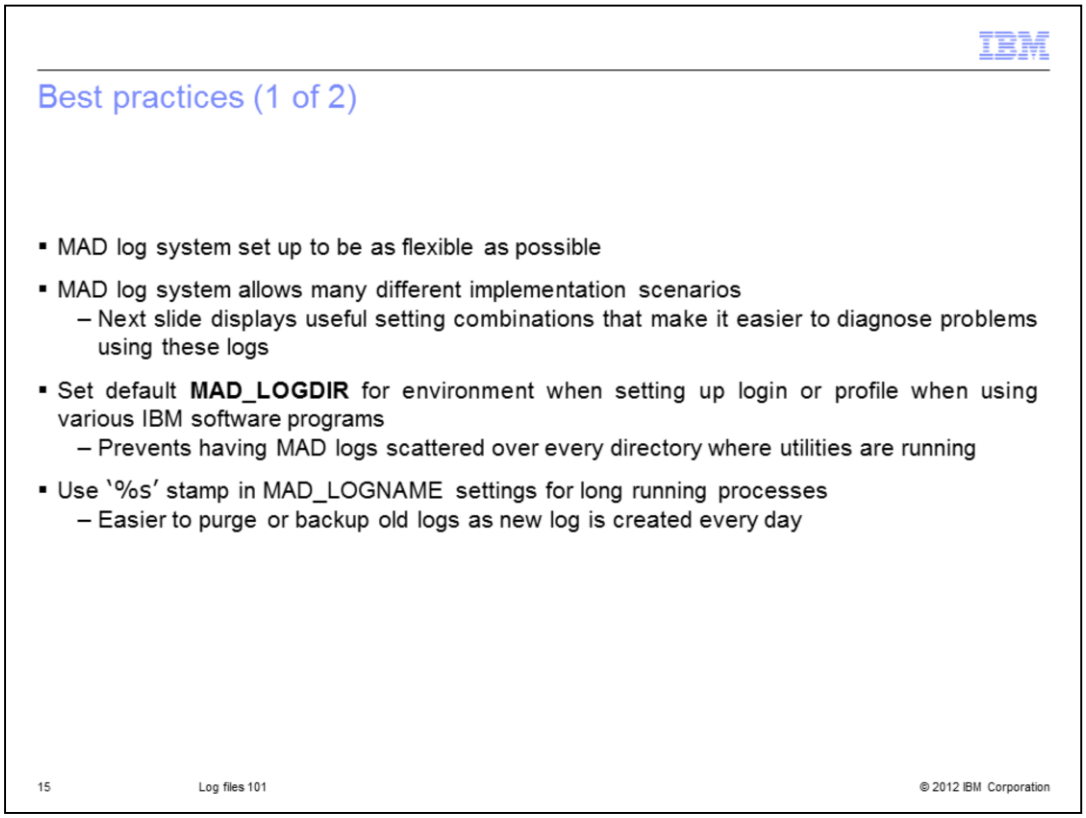


The second method to change log settings is through Workbench.

Changing the logging by way of this method can be done in real-time. For example, say that you are debugging an issue in your production environment. Rebooting the Master Data Engine is not an option, but you need to increase the amount of logging to troubleshoot an issue. You can turn on the DEBUG logging type through Workbench, trigger the error, and then turn this logging type off. All of which takes place in real-time. This has the benefit of not having to re-start your process and conserving disk space, as some logging types are extremely verbose. For details on this procedure, see the Workbench User Guide.

Note that changing the logging type through Workbench is not persistent. This means that changes will not update or write the master configuration file, engine.properties or log4j.xml and will only take effect until that process is restarted. At which time, the existing settings that are listed in the master configuration file, engine.properties or log4j.xml will be read in again and applied by the process.

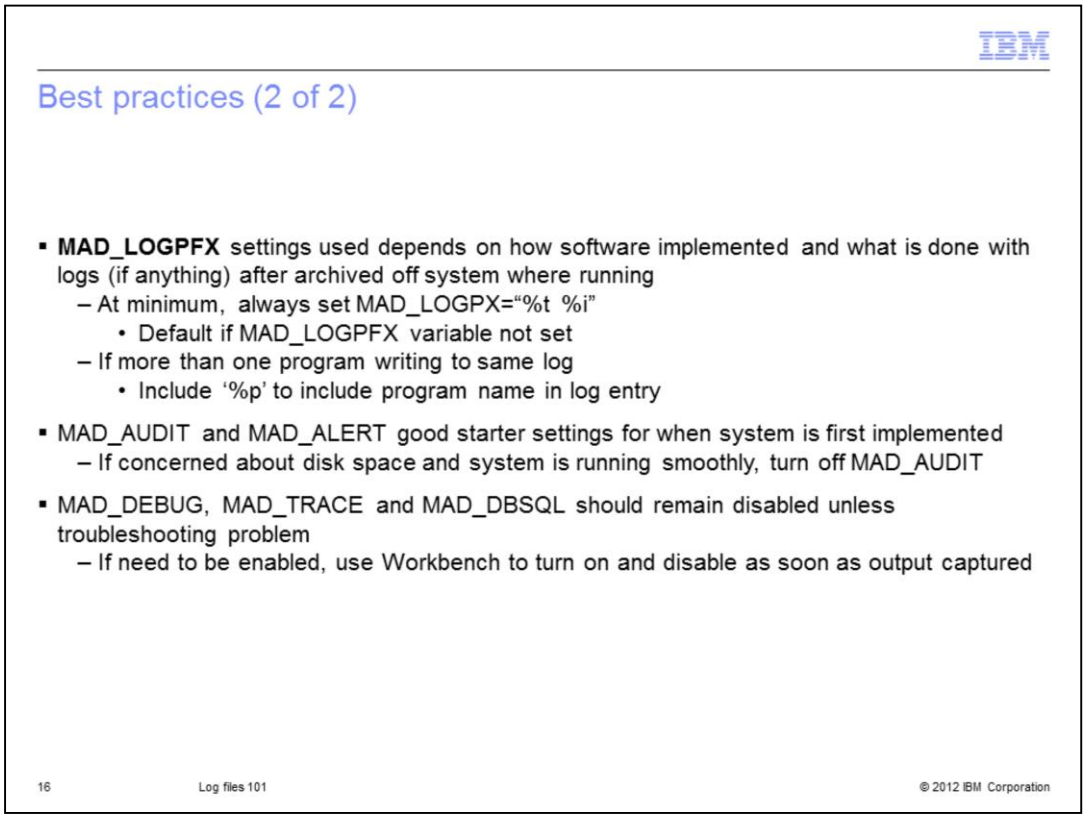
It is important to not enable debug trace or SQL trace settings for an extended amount of time. The amount of output that is written to the MAD logs is extremely verbose and can quickly consume disk space. Only enable these settings for short periods at a time.



So far, this training module has covered how to find log files and what information to make available to the TSPs from the log file. It also covered how to configure the different types of log settings and how to increase the amount of information you see in the log files. Now you will also be presented with logging best practices.

The MAD log system is set up to be as flexible as possible and to allow for many different implementation scenarios. One suggestion is to always set up a MAD_LOGDIR variable in your system. Some of the IBM Initiate utilities have to be run from the command line. Setting up a profile or environment that reference your MAD_LOGDIR will prevent MAD logs from being scattered over different directories when you are running a utility.

It is also recommended to use the percent signs option in the MAD_LOGNAME environment variable so that the date and time stamp is written in the file name. This makes it easier to purge or backup old log files as a new log file is created every day.



The option that you use with the MAD_LOGPFX depends on how the software is implemented and how you are planning on backing up the log files at a later time. At a minimum, use the %t and %i options. If not set, MAD_LOGPFX uses these by default.

If there is more than one program writing to the same log file, include %p. This writes the program name in each log file entry. This helps identify the program written in each entry.

At system startup or on initial implementation, MAD_AUDIT and MAD_ALERT are typically turned on. If you are concerned about disk usage and the system is running smoothly, you can turn off the MAD_AUDIT option.

MAD_DEBUG, MAD_TRACE and MAD_DBSQL should remain disabled unless you are troubleshooting a particular issue. If you are troubleshooting a reproducible issue, use Workbench to enable and disable those additional logging types in real time.

Frequently asked questions (1 of 2)

Q1. Where is the MAD log for our Master Data Engine located?

A1. The location of the log is defined in the master configuration file under your engine instance. The MAD_LOGNAME variable in your configuration indicates where this file is located.

Q2. An Inspector user reported an EODBC error after attempting to save a task. Which log file can I look at to see the details about the error?

A2. While Inspector itself does not write to a MAD log, the Master Data Engine host to which you are connected will log details about the error.

Q3.I am concerned about an ERROR I see in the log and I cannot tell what information from the log file is helpful. What can I provide to the TSP to help research the cause of this error?

A3. When in doubt, provide the entire log file or the 50 lines before and after the log entry in question.

The next two slides are some general questions and answers. Pause this presentation to review the questions and answers.

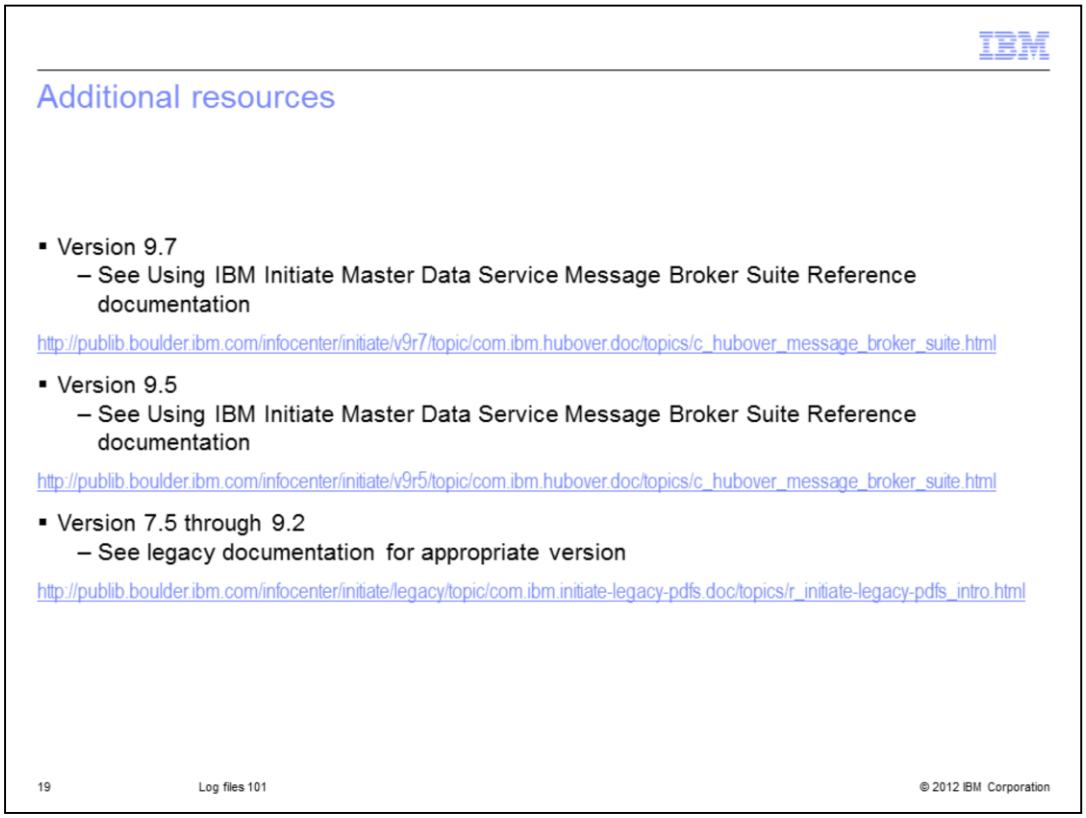
Frequently asked questions (2 of 2)

Q4: I have many AUDIT entries in the Inbound Broker log file and everything has been running smoothly for months. How can we disable the AUDIT entries?

A4: Locate the environment definition section for the Inbound Broker and change MAD_AUDIT=0. At the next startup of the Inbound Broker, AUDIT entries will no longer be written.

Q5: I am concerned about an error in the log file but there is not any other information in the log file to report. How can I increase the amount of logging without restarting the Master Data Engine?

A5: Changes to the Logging Type can be made in real-time through Workbench. Remember to turn additional logging off once you have captured the required output.



For additional information and details, visit the link that corresponds to the version of software you are using.

Trademarks, disclaimer, and copyright information