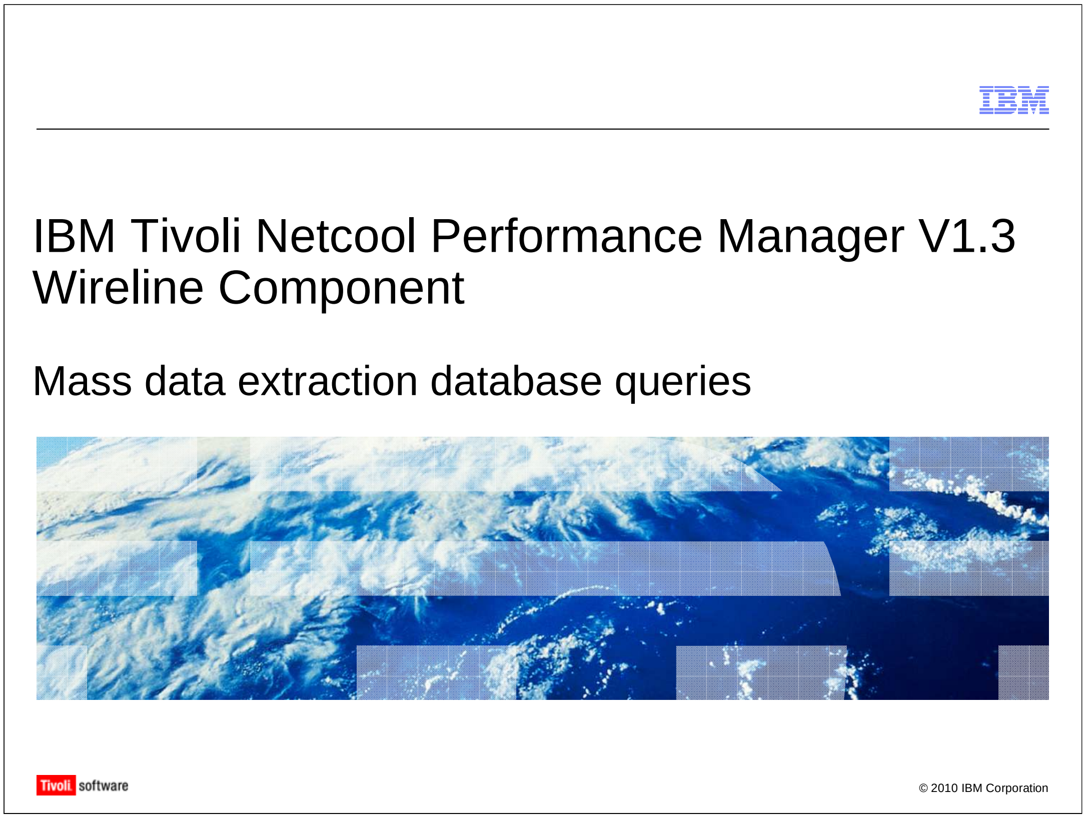
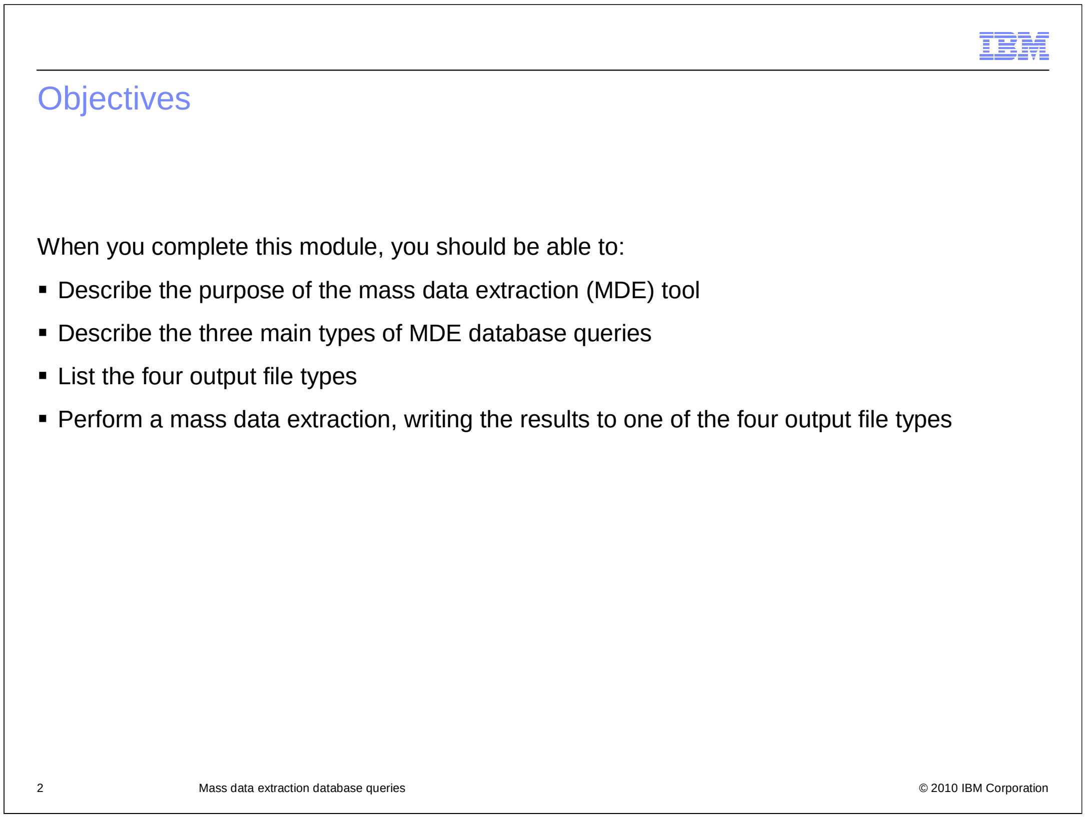


IBM Tivoli Netcool Performance Manager V1.3 Wireline Component

Mass data extraction database queries

Tivoli software

Welcome to the IBM Education Assistant module for Tivoli® Netcool® Performance Manager 1.3 Wireline Component entitled *Mass data extraction database queries*.

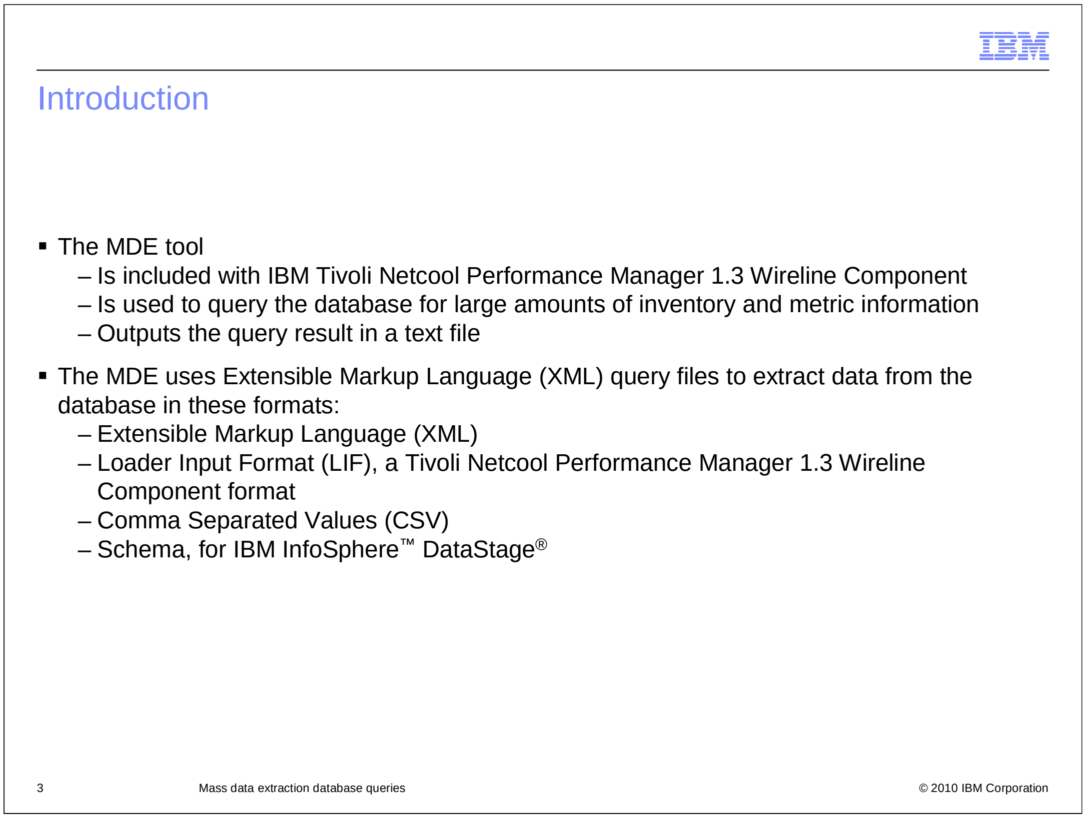


Objectives

When you complete this module, you should be able to:

- Describe the purpose of the mass data extraction (MDE) tool
- Describe the three main types of MDE database queries
- List the four output file types
- Perform a mass data extraction, writing the results to one of the four output file types

When you complete this IBM Education Assistant module, you should be able to describe the purpose of the mass data extraction tool, or MDE. You should also be able to describe the three main types of MDE database queries, list the four output file types, and perform a mass data extraction. When you perform a mass data extraction, the results are written to one of the four output file types.

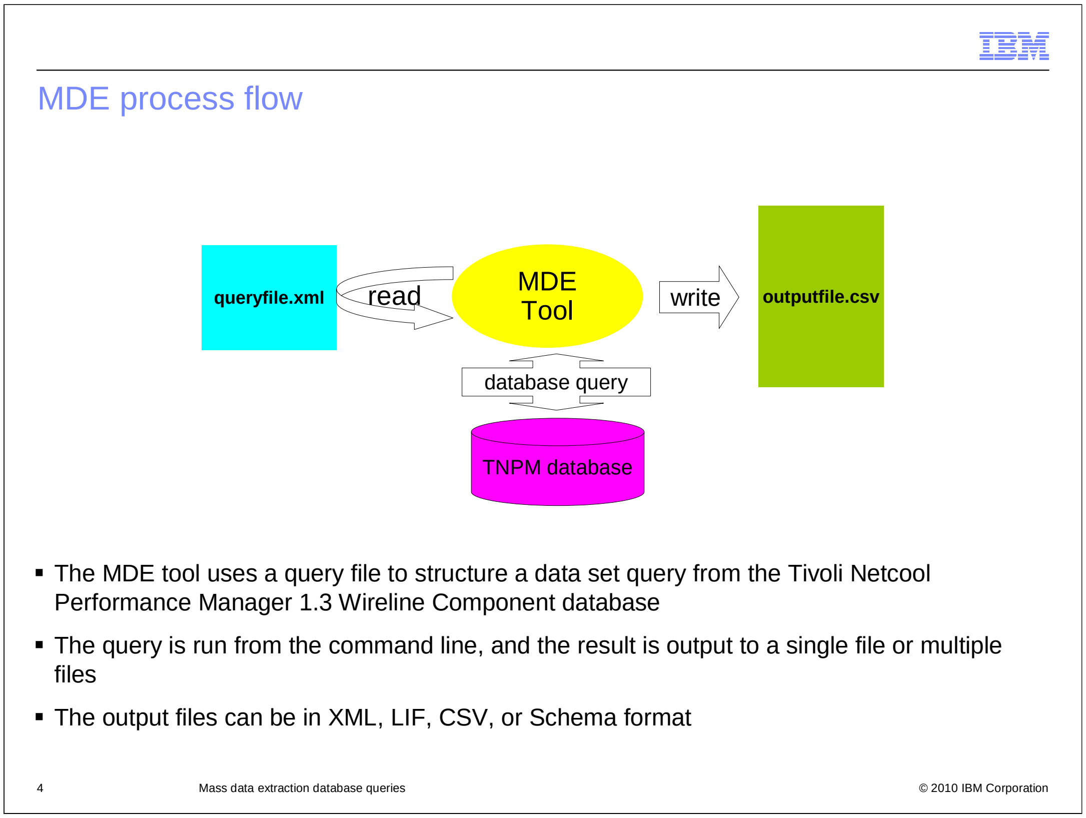


Introduction

- The MDE tool
  - Is included with IBM Tivoli Netcool Performance Manager 1.3 Wireline Component
  - Is used to query the database for large amounts of inventory and metric information
  - Outputs the query result in a text file
- The MDE uses Extensible Markup Language (XML) query files to extract data from the database in these formats:
  - Extensible Markup Language (XML)
  - Loader Input Format (LIF), a Tivoli Netcool Performance Manager 1.3 Wireline Component format
  - Comma Separated Values (CSV)
  - Schema, for IBM InfoSphere™ DataStage®

The MDE tool is included with IBM Tivoli Netcool Performance Manager 1.3 Wireline Component. This tool is used to query the database for large amounts of inventory and metric information and output the query result in a text file. This module describes how to use the MDE tool to perform various data extractions.

The MDE uses extensible markup language, or XML, query files to extract data from the database in the formats that are listed on the slide.

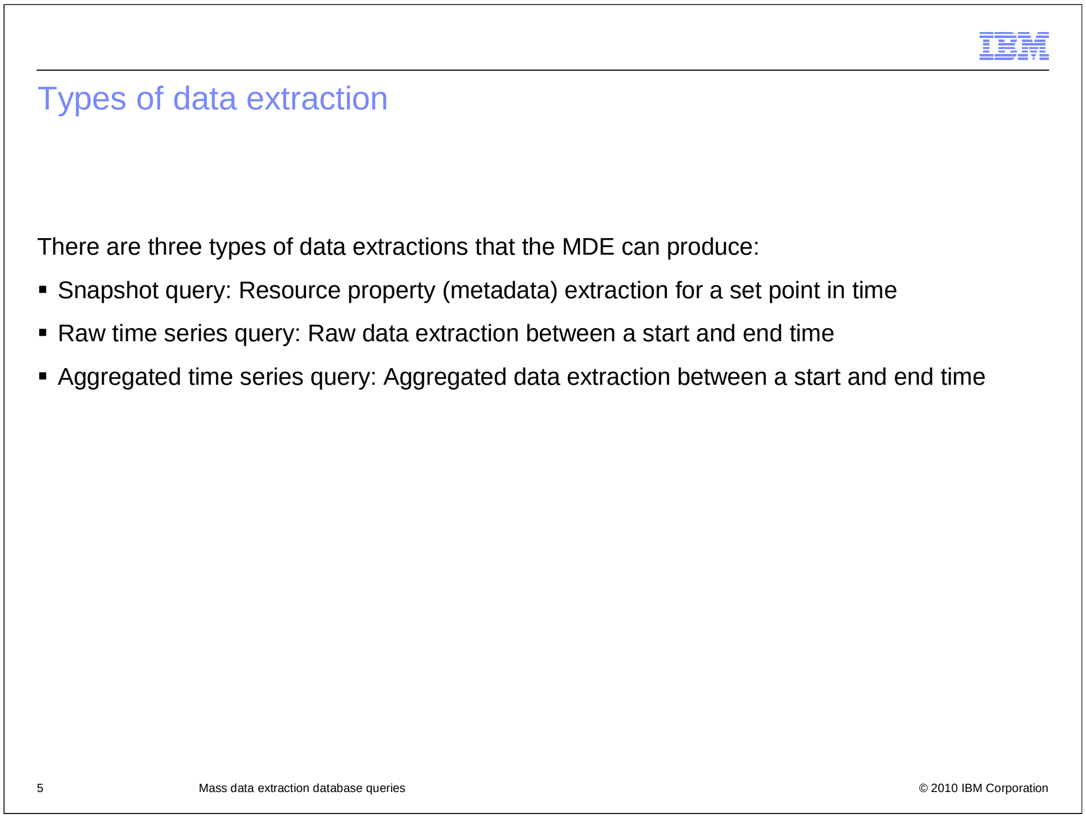


MDE process flow

- The MDE tool uses a query file to structure a data set query from the Tivoli Netcool Performance Manager 1.3 Wireline Component database
- The query is run from the command line, and the result is output to a single file or multiple files
- The output files can be in XML, LIF, CSV, or Schema format

The MDE tool uses a query file to structure a data set query from the Tivoli Netcool Performance Manager 1.3 Wireline Component database.

In this example, the MDE tool reads in the queryfile.xml file, which contains the database query command. It then executes the query in the database and writes the result to a single file or multiple files. The output files can be in XML, LIF, CSV, or Schema format. In this example, the result is an **outputfile.csv** file.



Types of data extraction

There are three types of data extractions that the MDE can produce:

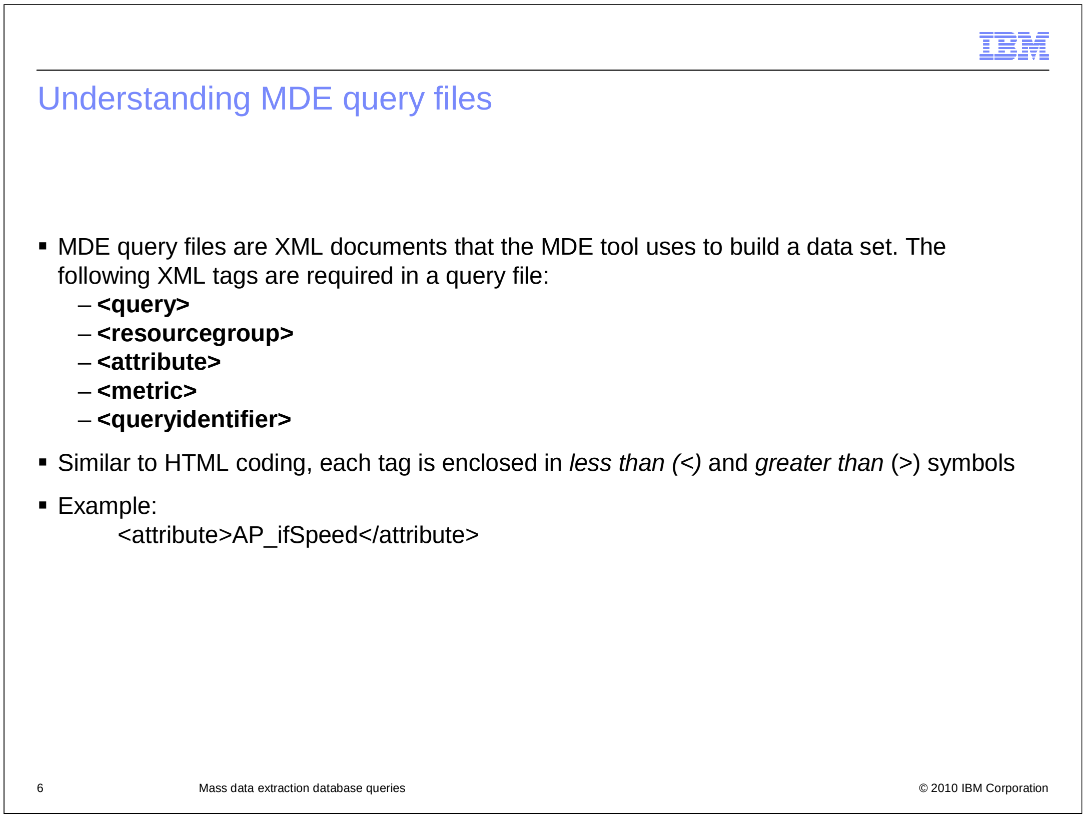
- Snapshot query: Resource property (metadata) extraction for a set point in time
- Raw time series query: Raw data extraction between a start and end time
- Aggregated time series query: Aggregated data extraction between a start and end time

There are three types of data extractions that the MDE can produce.

A snapshot query, which extracts resource property, or metadata, for a set point in time.

A raw time series query, which extracts raw data between a start and end time.

An aggregated time series query, which extracts aggregated data between a start and end time.



Understanding MDE query files

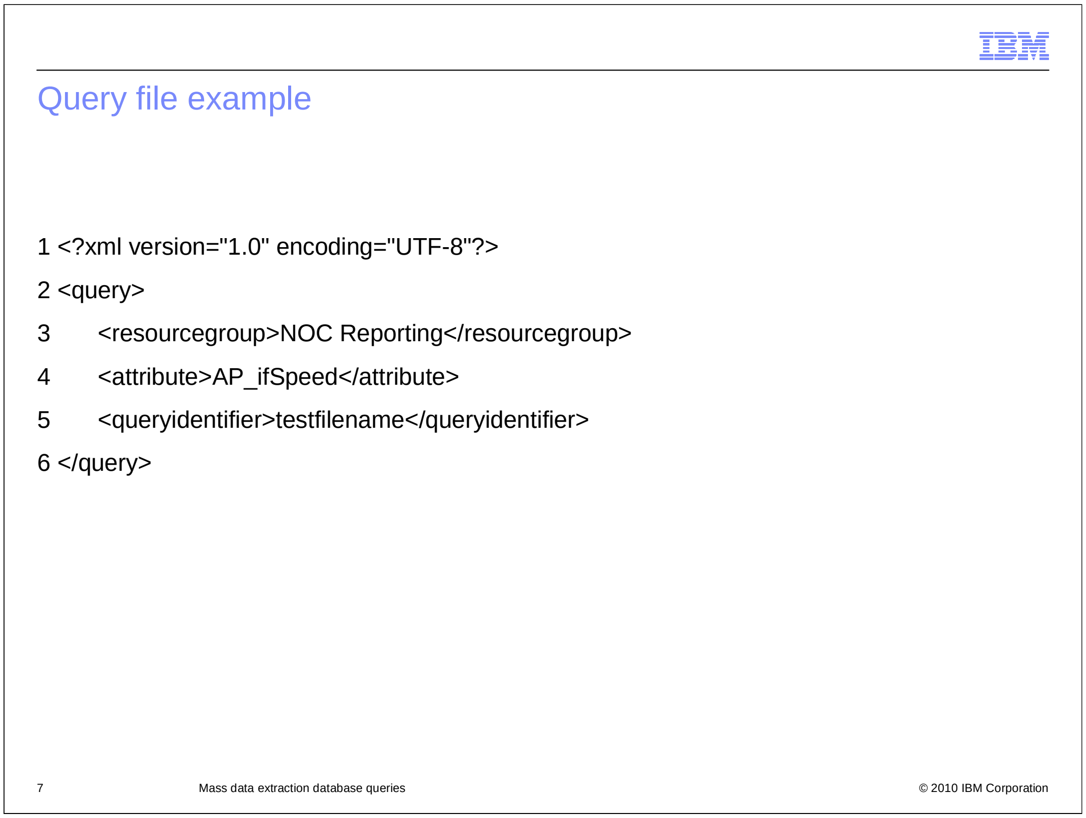
- MDE query files are XML documents that the MDE tool uses to build a data set. The following XML tags are required in a query file:
  - **<query>**
  - **<resourcegroup>**
  - **<attribute>**
  - **<metric>**
  - **<queryidentifier>**
- Similar to HTML coding, each tag is enclosed in *less than (<)* and *greater than* (>) symbols
- Example:

  <attribute>AP_ifSpeed</attribute>

MDE query files are XML documents that the MDE tool uses to build a data set to extract data from the database.

> Use the **query** tag to wrap all other tags and indicate that they form a query.
>
> Use the **resourcegroup** tag to indicate the reporting group where the resources to query are found. The value in the tag must include the full path to the reporting group. This group contains the resources that are the target of the query. Report group paths are delimited by the tilde (~) character.
>
> Use the **attribute** tag to identify a property to be extracted. Your query must contain a minimum of one of these tags.
>
> Use the **metric** tag to identify a metric to be extracted. This tag is used only in times series queries. The value in this tag must include the full path to the collection formula. The path is delimited by the tilde (~) character.
>
> Use the **queryidentifier** tag to define the naming convention of the output files, which can be CSV, LIF, XML, or Schema.

Similar to HTML coding, each tag is enclosed in *less than* and *greater than* symbols. An example of a correct **attribute** tag is shown on this slide.



Query file example

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <query>
3       <resourcegroup>NOC Reporting</resourcegroup>
4       <attribute>AP_ifSpeed</attribute>
5       <queryidentifier>testfilename</queryidentifier>
6 </query>
```

This slide shows an example of a simple query file. Line numbers have been added to aid in explanation.

Line 1 declares the XML version and encoding.

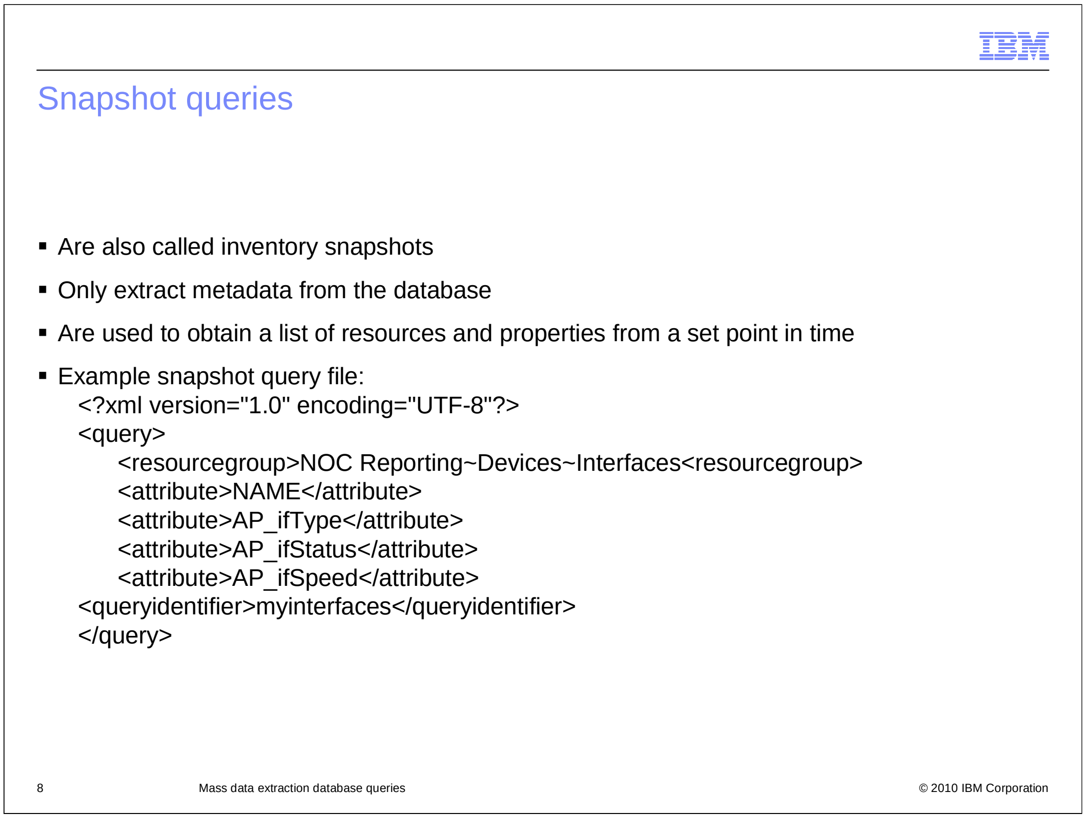
Line 2 starts the query section of the file with the **query** tag.

Line 3 selects all resources in the NOC Reporting groups, and all dependent groups. It begins and ends with the **resourcegroup** tag.

Line 4 selects the subelement property AP_ifSpeed. It begins and ends with the **attribute** tag.

Line 5 sets the name of the generated file name to testfilename. It begins and ends with the **queryidentifier** tag.

Line 6 ends the query section of the file with a terminal **query** tag.



Snapshot queries

- Are also called inventory snapshots
- Only extract metadata from the database
- Are used to obtain a list of resources and properties from a set point in time
- Example snapshot query file:

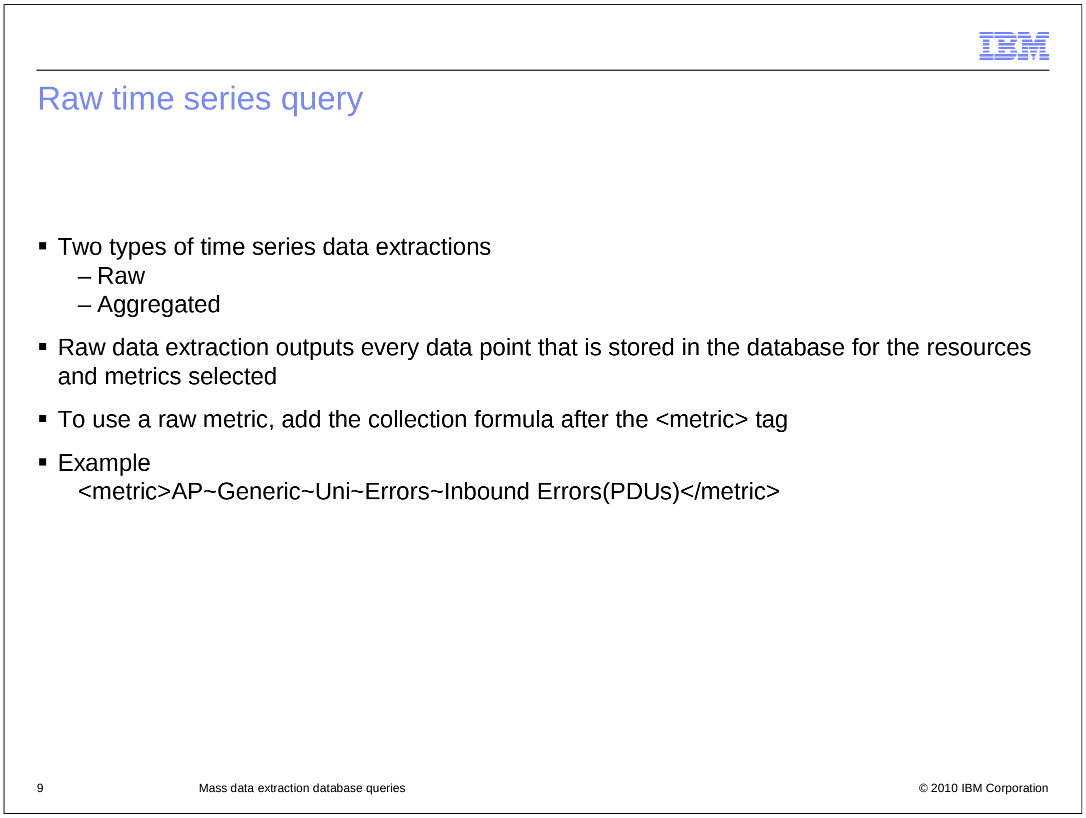
```
<?xml version="1.0" encoding="UTF-8"?>
<query>
    <resourcegroup>NOC Reporting~Devices~Interfaces<resourcegroup>
    <attribute>NAME</attribute>
    <attribute>AP_ifType</attribute>
    <attribute>AP_ifStatus</attribute>
    <attribute>AP_ifSpeed</attribute>
<queryidentifier>myinterfaces</queryidentifier>
</query>
```

Snapshot queries, also called inventory snapshots, only extract metadata from the database. They are used to obtain a list of resources and properties from a set point in time.

In this example of a snapshot query file, the resource group that is being targeted is **NOC Reporting~Devices~Interfaces**. Note the tilde symbol used as the delimiter.

From this resource group, the attributes, or properties, of the **Name, AP_ifType, AP_ifStatus,** and **AP_ifSpeed** are going to be extracted from the database.

The name of the query file, denoted by the **queryidentifier** tag, is *myinterfaces*.



Raw time series query

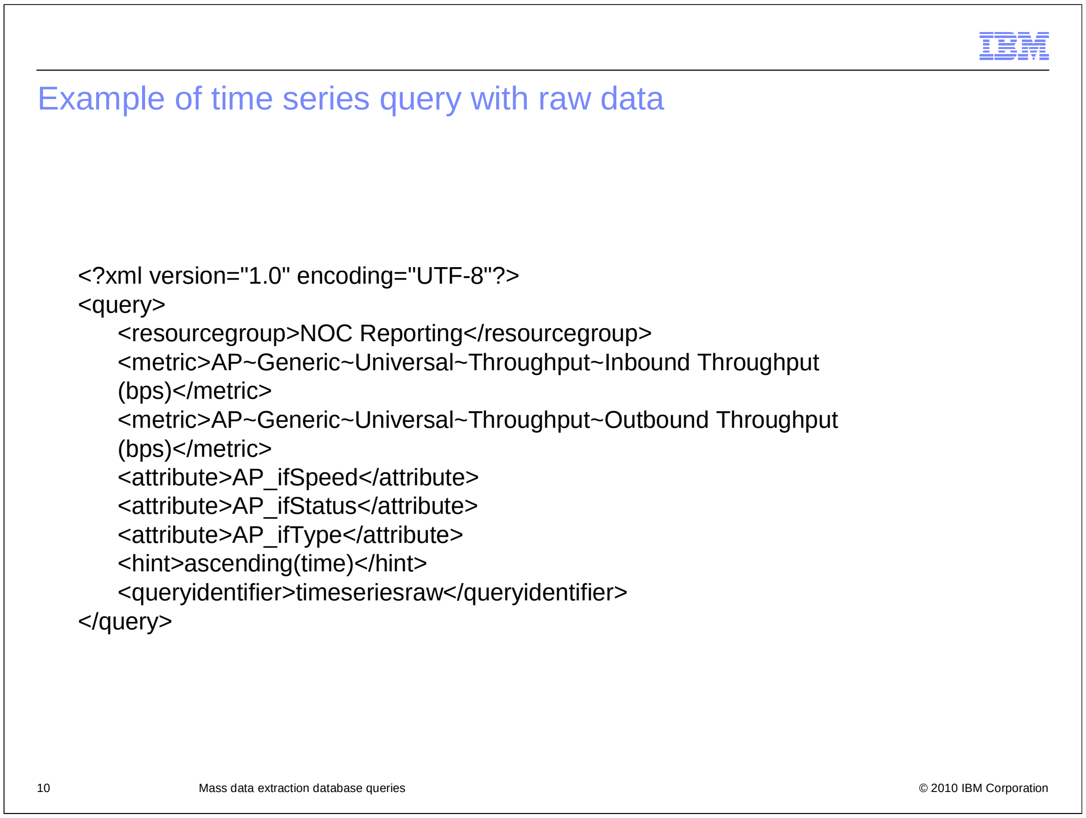
- Two types of time series data extractions
  - – Raw
  - – Aggregated
- Raw data extraction outputs every data point that is stored in the database for the resources and metrics selected
- To use a raw metric, add the collection formula after the <metric> tag
- Example
  <metric>AP~Generic~Uni~Errors~Inbound Errors(PDUs)</metric>

9 Mass data extraction database queries © 2010 IBM Corporation

There are two types of time series data extractions: Raw and aggregated.

Raw data extraction outputs every data point that is stored in the database for the resources and metrics selected.

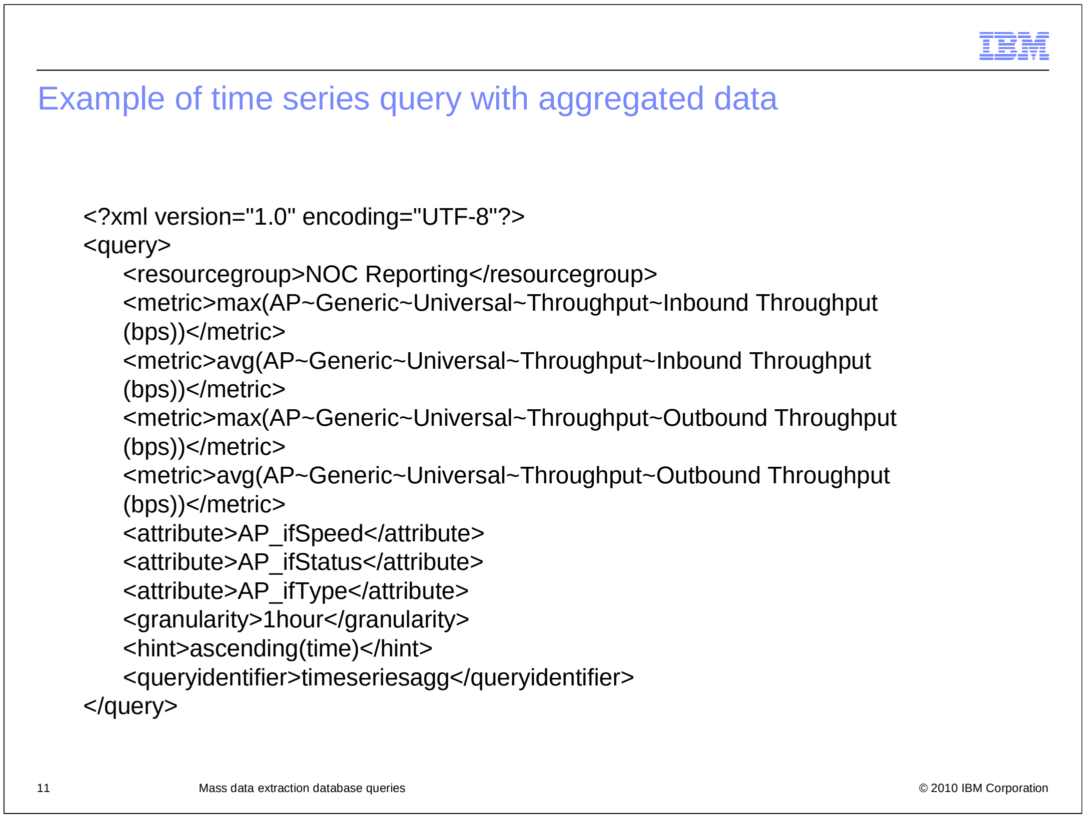
To use a raw metric, you add the collection formula after the **metric** tag. The example on this slide shows the tag structure to extract raw inbound errors. Note the tilde character used as the delimiter.



Example of time series query with raw data

```
<?xml version="1.0" encoding="UTF-8"?>
<query>
    <resourcegroup>NOC Reporting</resourcegroup>
    <metric>AP~Generic~Universal~Throughput~Inbound Throughput
    (bps)</metric>
    <metric>AP~Generic~Universal~Throughput~Outbound Throughput
    (bps)</metric>
    <attribute>AP_ifSpeed</attribute>
    <attribute>AP_ifStatus</attribute>
    <attribute>AP_ifType</attribute>
    <hint>ascending(time)</hint>
    <queryidentifier>timeseriesraw</queryidentifier>
</query>
```

In this example, the query is extracting two throughput metrics from the database: Inbound and outbound. The metrics are extracted from all resources in the NOC Reporting group. Three properties are also extracted with the **attribute** tag; AP_ifSpeed, AP_IfStatus, and AP_ifType. The value in the **hint** tag sets the sort order of the output file to ascending by time. The query name, set by the **queryidentifier** tag, is *timeseriesraw*.



Example of time series query with aggregated data

```
<?xml version="1.0" encoding="UTF-8"?>
<query>
    <resourcegroup>NOC Reporting</resourcegroup>
    <metric>max(AP~Generic~Universal~Throughput~Inbound Throughput
    (bps))</metric>
    <metric>avg(AP~Generic~Universal~Throughput~Inbound Throughput
    (bps))</metric>
    <metric>max(AP~Generic~Universal~Throughput~Outbound Throughput
    (bps))</metric>
    <metric>avg(AP~Generic~Universal~Throughput~Outbound Throughput
    (bps))</metric>
    <attribute>AP_ifSpeed</attribute>
    <attribute>AP_ifStatus</attribute>
    <attribute>AP_ifType</attribute>
    <granularity>1hour</granularity>
    <hint>ascending(time)</hint>
    <queryidentifier>timeseriesagg</queryidentifier>
</query>
```

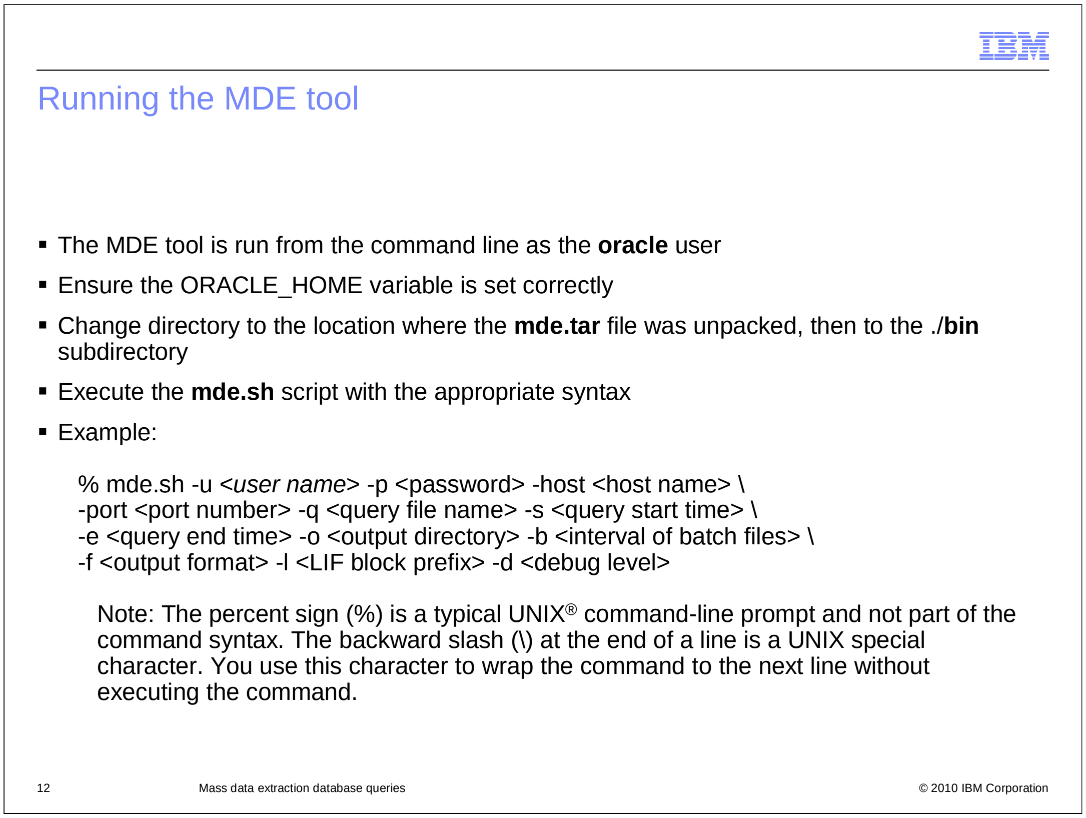
11 Mass data extraction database queries © 2010 IBM Corporation

In this example, the query is extracting four throughput metrics from the database:

- •Maximum Inbound Throughput in bits per second for each hour
- •Average Inbound Throughput in bits per second for each hour
- •Maximum Outbound Throughput in bits per second for each hour
- •Average Outbound Throughput in bits per second for each hour

Each of these metrics is specified by either the *max,* for maximum, or *avg,* for average, designator. These are what differentiate an aggregated time series query from a raw time series query.

The metrics are extracted from all resources in the NOC Reporting group. Three properties are also extracted with the **attribute** tag. The value of the **granularity** tag is set to one hour. The value in the **hint** tag sets the sort order of the output file to ascending by time. The query name, set by the **queryidentifier** tag, is *timeseriesagg*.



Running the MDE tool

- The MDE tool is run from the command line as the **oracle** user
- Ensure the ORACLE_HOME variable is set correctly
- Change directory to the location where the **mde.tar** file was unpacked, then to the ./**bin** subdirectory
- Execute the **mde.sh** script with the appropriate syntax
- Example:

```
% mde.sh -u <user name> -p <password> -host <host name> \
-port <port number> -q <query file name> -s <query start time> \
-e <query end time> -o <output directory> -b <interval of batch files> \
-f <output format> -l <LIF block prefix> -d <debug level>
```

Note: The percent sign (%) is a typical UNIX® command-line prompt and not part of the command syntax. The backward slash (\) at the end of a line is a UNIX special character. You use this character to wrap the command to the next line without executing the command.

The MDE tool is run from the command line as the **oracle** user.

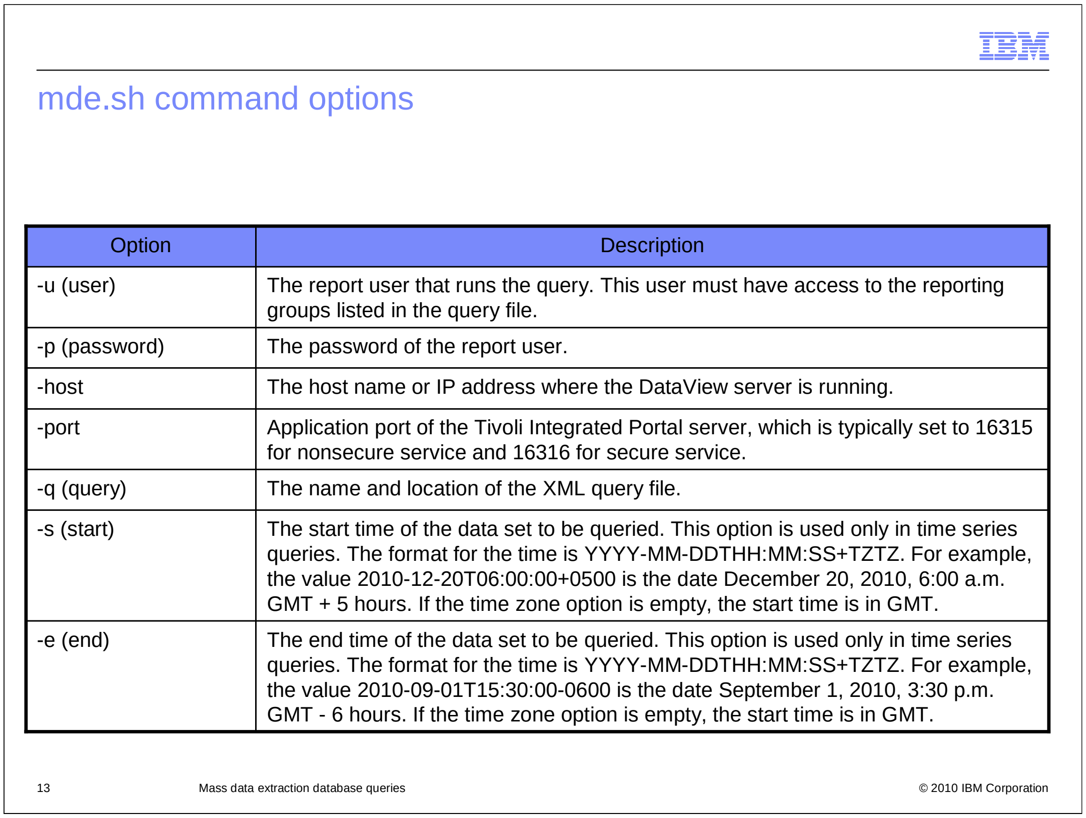
Before running the **mde.sh** script, ensure the ORACLE_HOME variable is set correctly.

Next, change the directory to the location where the **mde.tar** file was unpacked, then to the ./**bin** subdirectory.

Finally, start the **mde.sh** script with the appropriate syntax for the type of data extraction you want to perform.

Note that the percent sign is a typical UNIX command-line prompt and not part of the command syntax. Also, the backward slash at the end of a line is a UNIX special character that wraps the command to the next line without executing the command. The prompt and special characters are repeated in command examples throughout this IBM Education Assistant module.

The next two slides show a complete listing of available options for the **mde.sh** command.



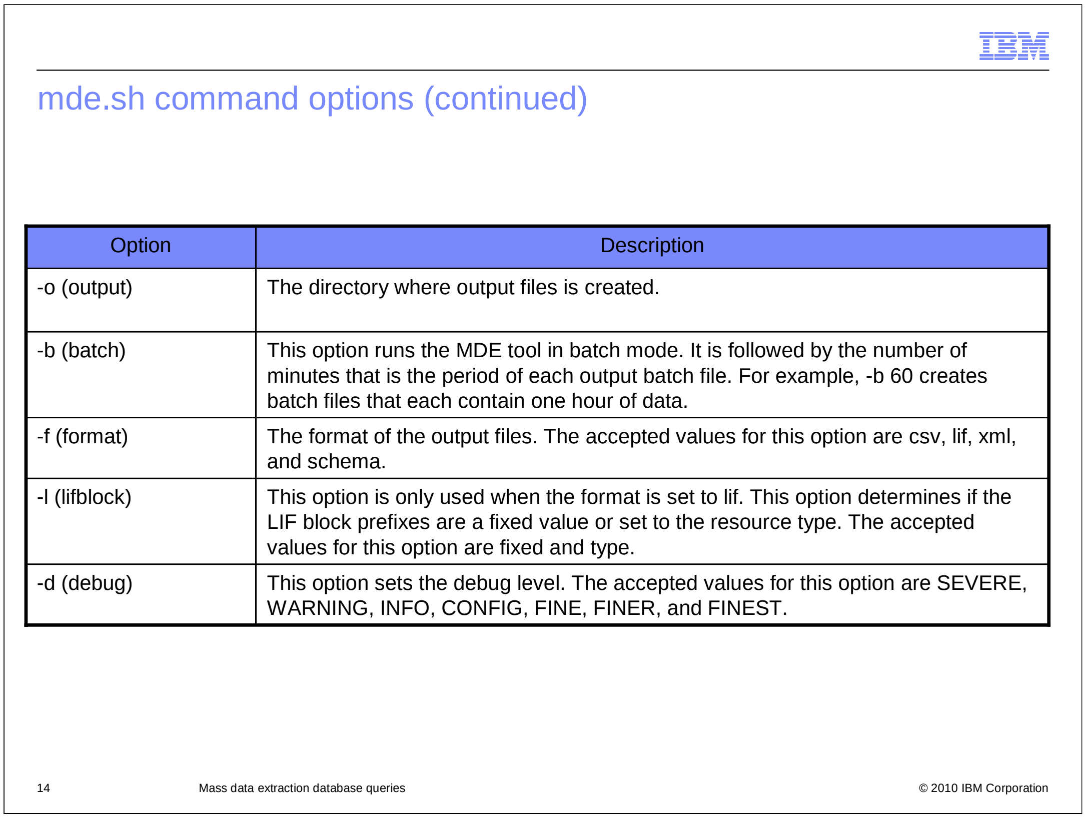
mde.sh command options

Option	Description
-u (user)	The report user that runs the query. This user must have access to the reporting groups listed in the query file.
-p (password)	The password of the report user.
-host	The host name or IP address where the DataView server is running.
-port	Application port of the Tivoli Integrated Portal server, which is typically set to 16315 for nonsecure service and 16316 for secure service.
-q (query)	The name and location of the XML query file.
-s (start)	The start time of the data set to be queried. This option is used only in time series queries. The format for the time is YYYY-MM-DDTHH:MM:SS+TZTZ. For example, the value 2010-12-20T06:00:00+0500 is the date December 20, 2010, 6:00 a.m. GMT + 5 hours. If the time zone option is empty, the start time is in GMT.
-e (end)	The end time of the data set to be queried. This option is used only in time series queries. The format for the time is YYYY-MM-DDTHH:MM:SS+TZTZ. For example, the value 2010-09-01T15:30:00-0600 is the date September 1, 2010, 3:30 p.m. GMT - 6 hours. If the time zone option is empty, the start time is in GMT.

13 Mass data extraction database queries

This slide shows the **mde.sh** script command options that are available. Some of the options are specific to certain types of data extractions.

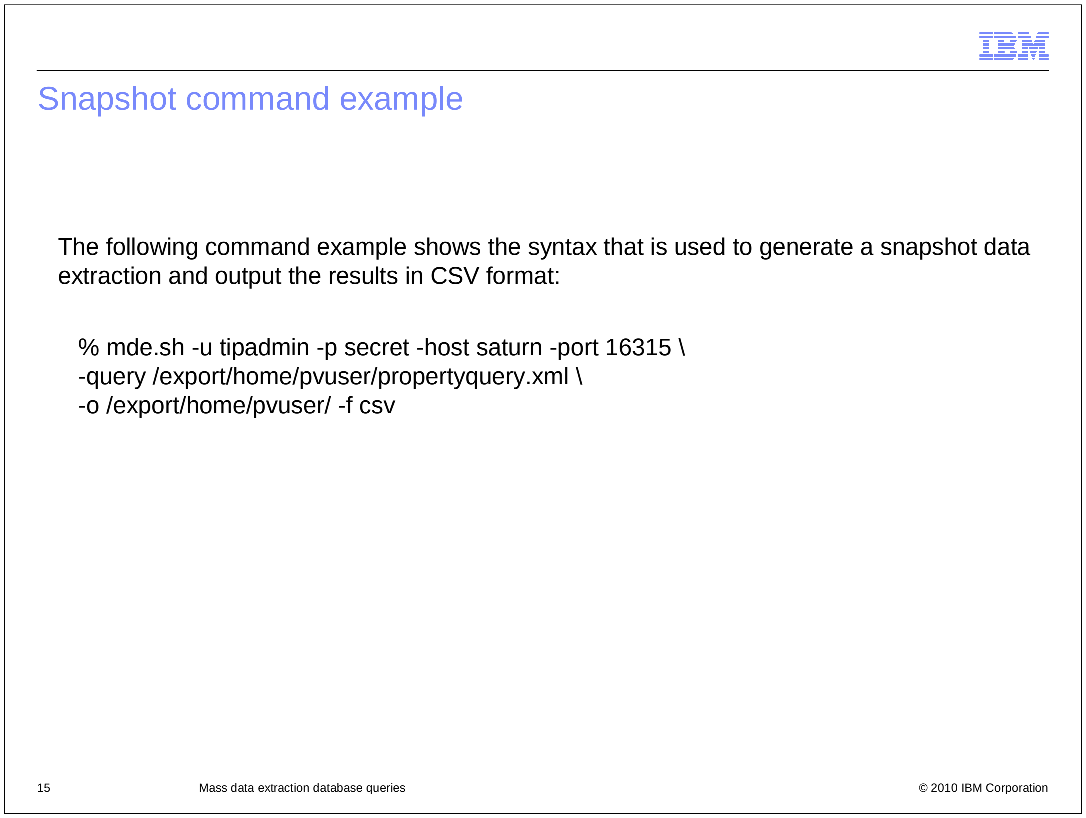
For instance, the options –s, for start, and –e, for end, are used only in time series queries.



mde.sh command options (continued)

Option	Description
-o (output)	The directory where output files is created.
-b (batch)	This option runs the MDE tool in batch mode. It is followed by the number of minutes that is the period of each output batch file. For example, -b 60 creates batch files that each contain one hour of data.
-f (format)	The format of the output files. The accepted values for this option are csv, lif, xml, and schema.
-l (lifblock)	This option is only used when the format is set to lif. This option determines if the LIF block prefixes are a fixed value or set to the resource type. The accepted values for this option are fixed and type.
-d (debug)	This option sets the debug level. The accepted values for this option are SEVERE, WARNING, INFO, CONFIG, FINE, FINER, and FINEST.

This slide continues the list of command options.

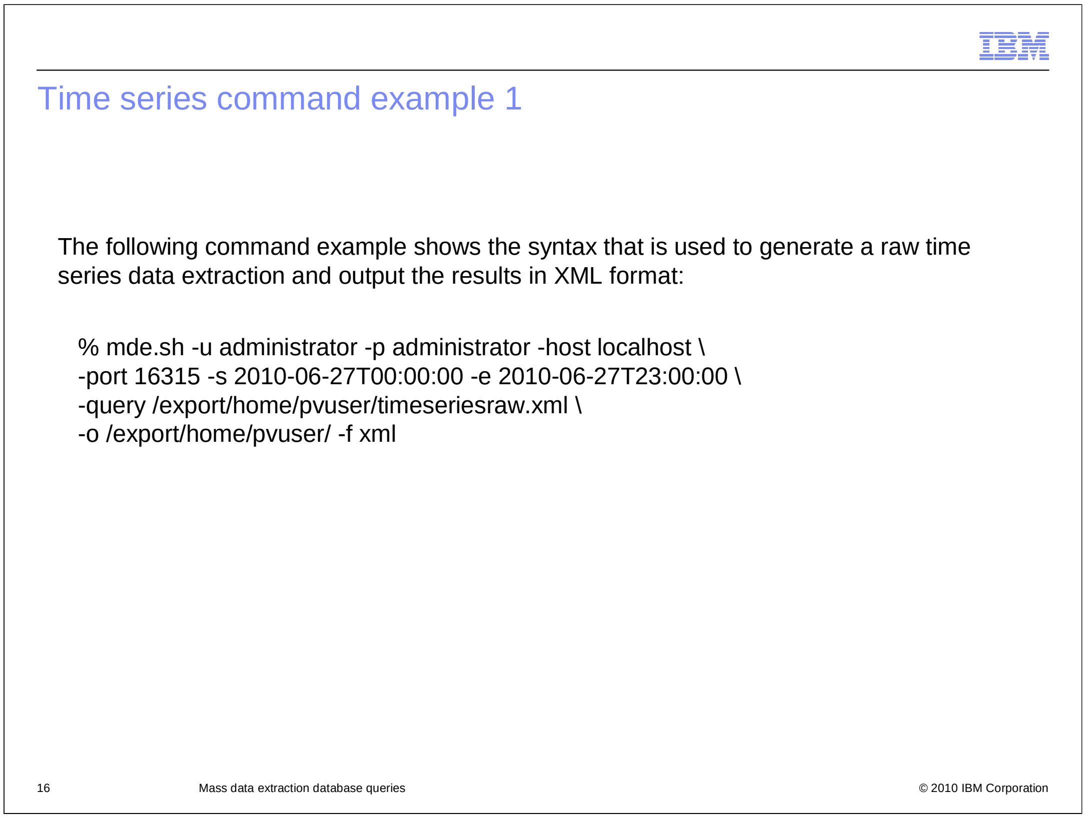


Snapshot command example

The following command example shows the syntax that is used to generate a snapshot data extraction and output the results in CSV format:

```
% mde.sh -u tipadmin -p secret -host saturn -port 16315 \
-query /export/home/pvuser/propertyquery.xml \
-o /export/home/pvuser/ -f csv
```

This slide shows an example of a snapshot data extraction command. It is run with the username *tipadmin* with the password *secret* on the host *saturn* using port 16315. Because this is a snapshot query, there is no start or end time. It uses the parameters contained in the file *propertyquery.xml* and outputs the results in CSV format into the directory **/export/home/pvuser**.

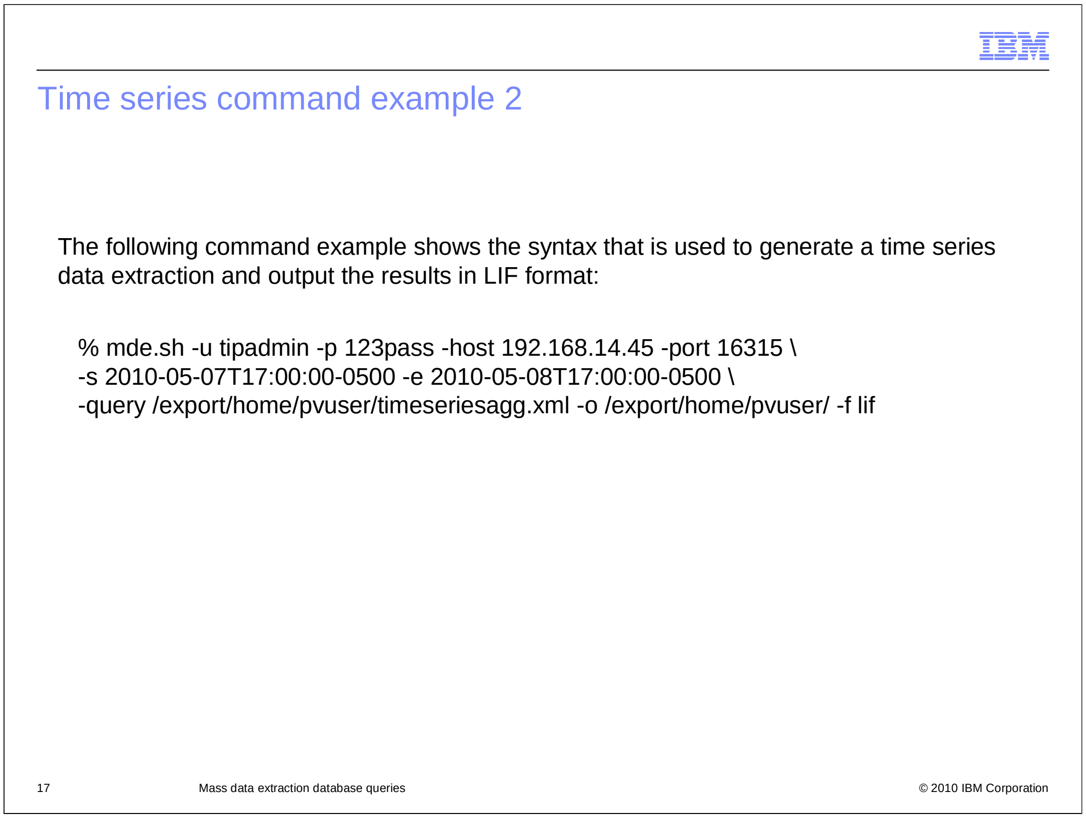


Time series command example 1

The following command example shows the syntax that is used to generate a raw time series data extraction and output the results in XML format:

```
% mde.sh -u administrator -p administrator -host localhost \
-port 16315 -s 2010-06-27T00:00:00 -e 2010-06-27T23:00:00 \
-query /export/home/pvuser/timeseriesraw.xml \
-o /export/home/pvuser/ -f xml
```

This slide shows an example of a raw time series data extraction command. It is run with the username *administrator* with the password *administrator* on the localhost using port 16315. The time series start time, following the **–s** option, is June 27, 2010 at midnight GMT. The end time, following the **–e** option, is June 27, 2010 at 11:00 p.m. GMT. It uses the parameters contained in the file *timeseriesraw.xml* and outputs the results in XML format into the directory **/export/home/pvuser**.

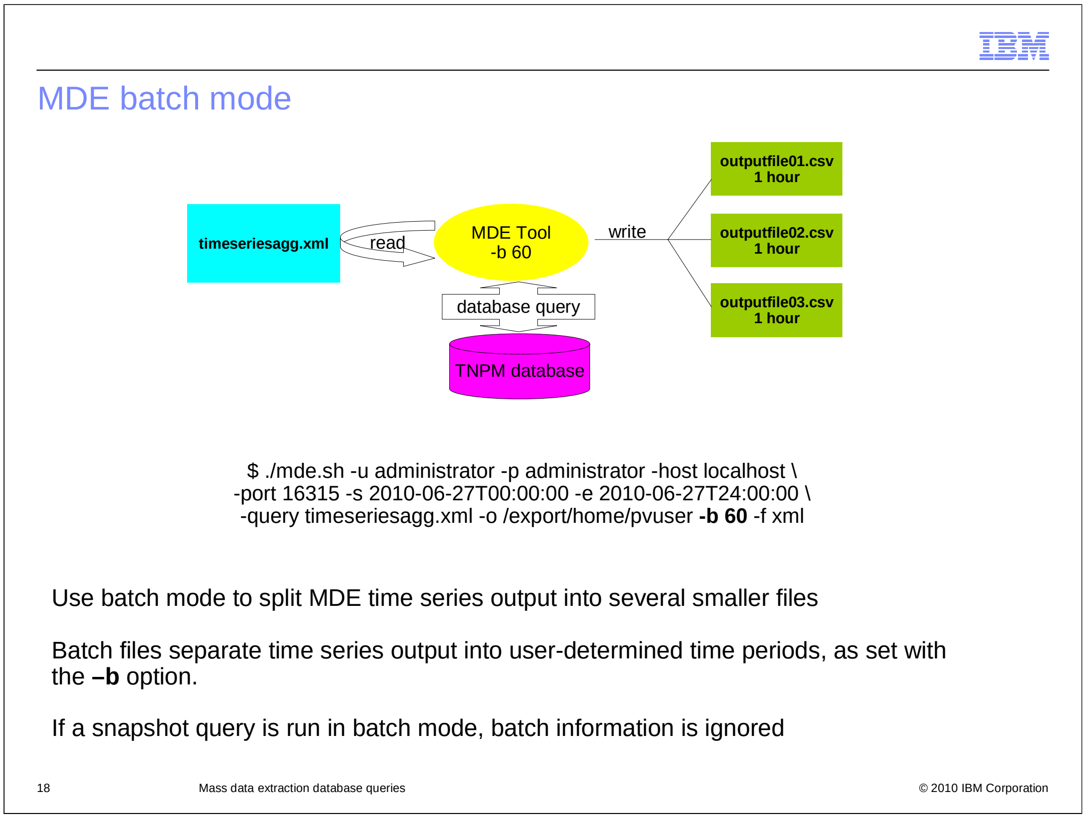


Time series command example 2

The following command example shows the syntax that is used to generate a time series data extraction and output the results in LIF format:

```
% mde.sh -u tipadmin -p 123pass -host 192.168.14.45 -port 16315 \
-s 2010-05-07T17:00:00-0500 -e 2010-05-08T17:00:00-0500 \
-query /export/home/pvuser/timeseriesagg.xml -o /export/home/pvuser/ -f lif
```

This slide shows another example of an aggregated time series data extraction command. It is run with the username *tipadmin* with the password *123pass* on the host with the IP address 192.168.14.45 using port 16315. The time series start time is May 7, 2010 at 5:00pm GMT -5 hours. The end time is May 8, 2010 at 5:00 p.m. GMT -5 hours. It uses the parameters contained in the file *timeseriesagg.xml* and outputs the results in LIF format into the directory **/export/home/pvuser**.



MDE batch mode

timeseriesagg.xml

read

MDE Tool
-b 60

write

outputfile01.csv
1 hour

outputfile02.csv
1 hour

outputfile03.csv
1 hour

database query

TNPM database

```
$ ./mde.sh -u administrator -p administrator -host localhost \
-port 16315 -s 2010-06-27T00:00:00 -e 2010-06-27T24:00:00 \
-query timeseriesagg.xml -o /export/home/pvuser -b 60 -f xml
```

Use batch mode to split MDE time series output into several smaller files

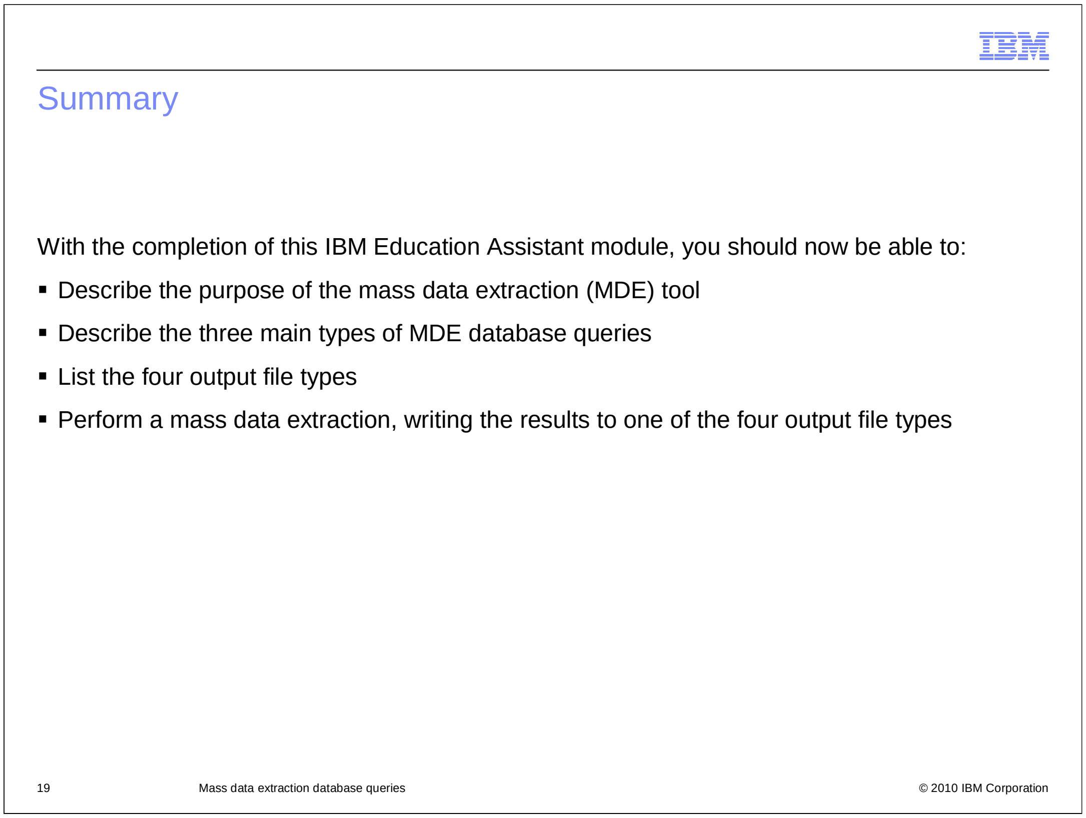
Batch files separate time series output into user-determined time periods, as set with the **–b** option.

If a snapshot query is run in batch mode, batch information is ignored

You can use the MDE utility to batch extracted time series data. The extracted data is split up and output as a set of files. Each file contains data for a set time period. The advantage of batch mode is that you can reduce file size and increase readability.

Batch mode is invoked with the **-b** option, followed by the number of minutes of data each file contains.

In this example, the query file *timeseriesagg.xml* is used to provide the information on the data to extract. The command line contains the start and end times of the time series extraction, which is for one 24-hour period.
The *–b 60* option is included in the command line to break up the 24-hour time series into 60-minute segments. It outputs each one into a separate file containing one hour of data. At the end of the command execution, there are 24 separate output files, containing one hour of data each.

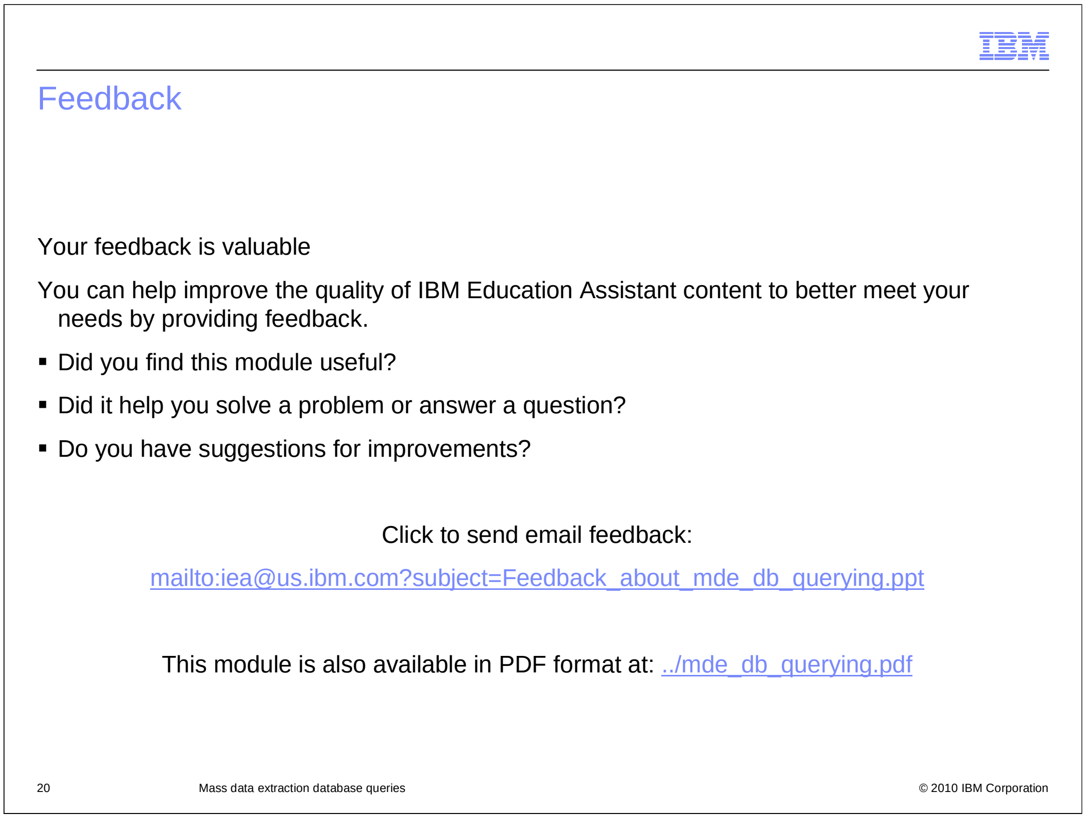


Summary

With the completion of this IBM Education Assistant module, you should now be able to:

- Describe the purpose of the mass data extraction (MDE) tool
- Describe the three main types of MDE database queries
- List the four output file types
- Perform a mass data extraction, writing the results to one of the four output file types

This concludes the IBM Education Assistant module for Tivoli Netcool Performance Manager 1.3 Wireline Component entitled *Mass data extraction database queries*.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

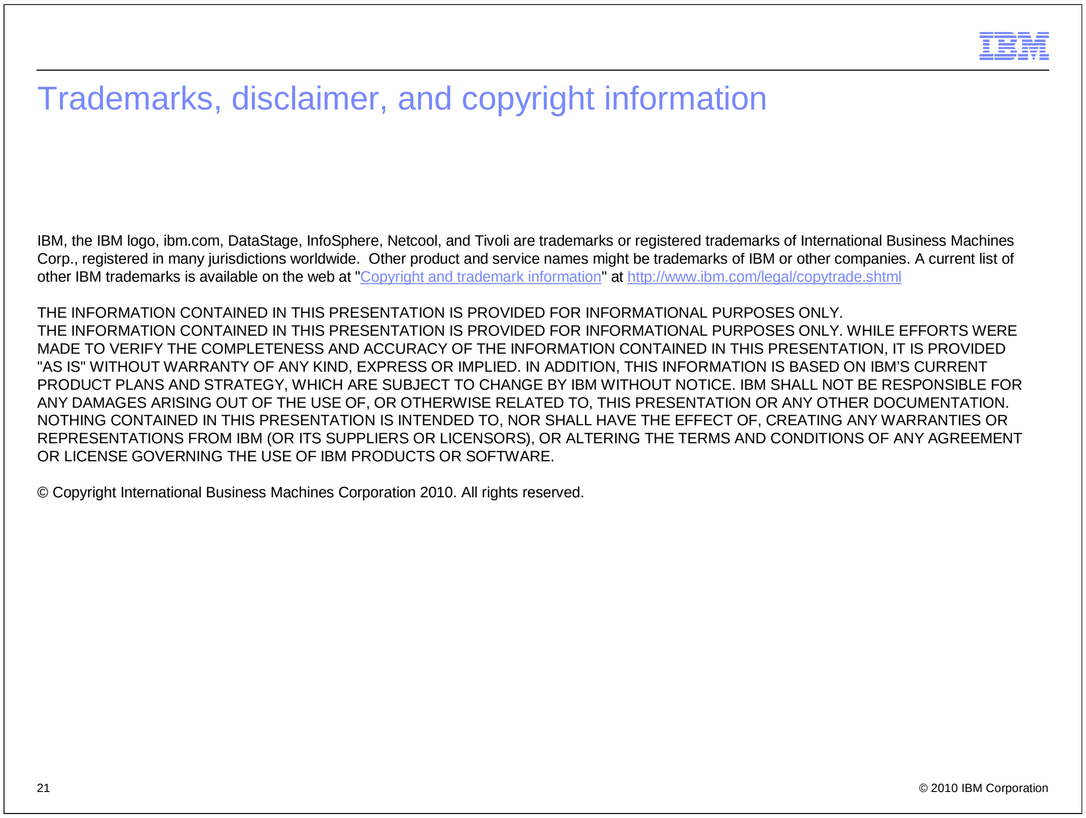
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_mde_db_querying.ppt

This module is also available in PDF format at: ../mde_db_querying.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DataStage, InfoSphere, Netcool, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.