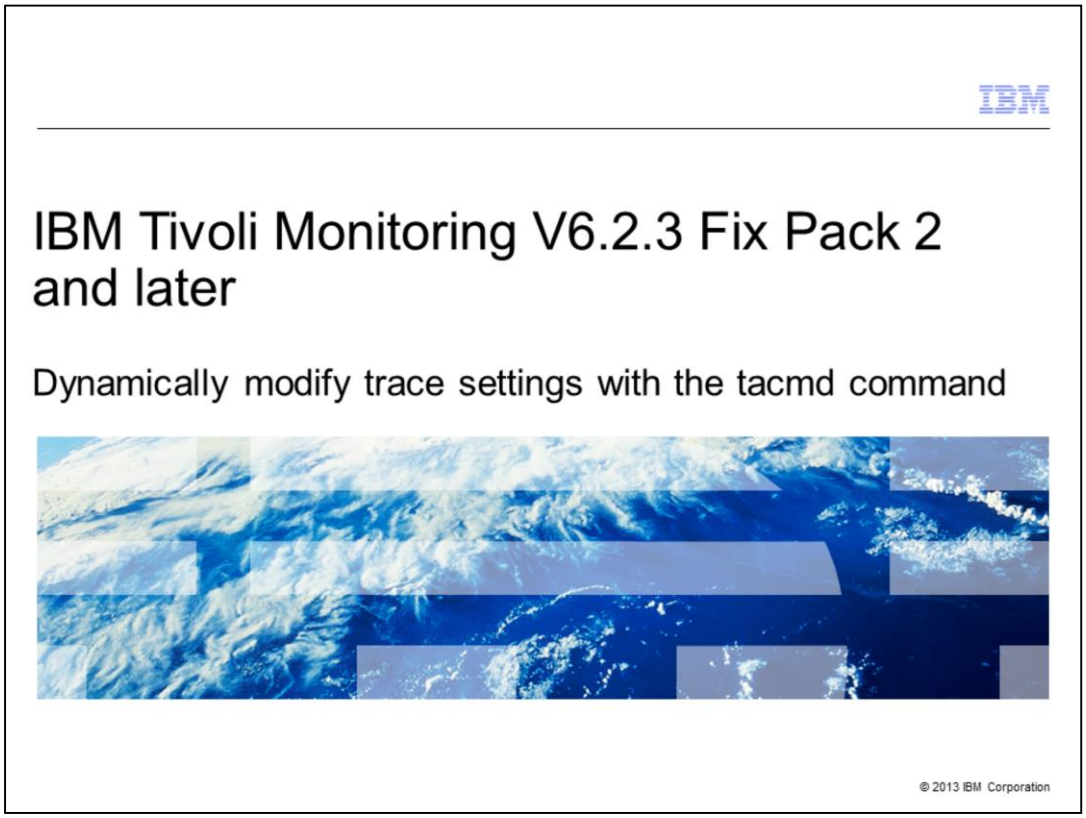
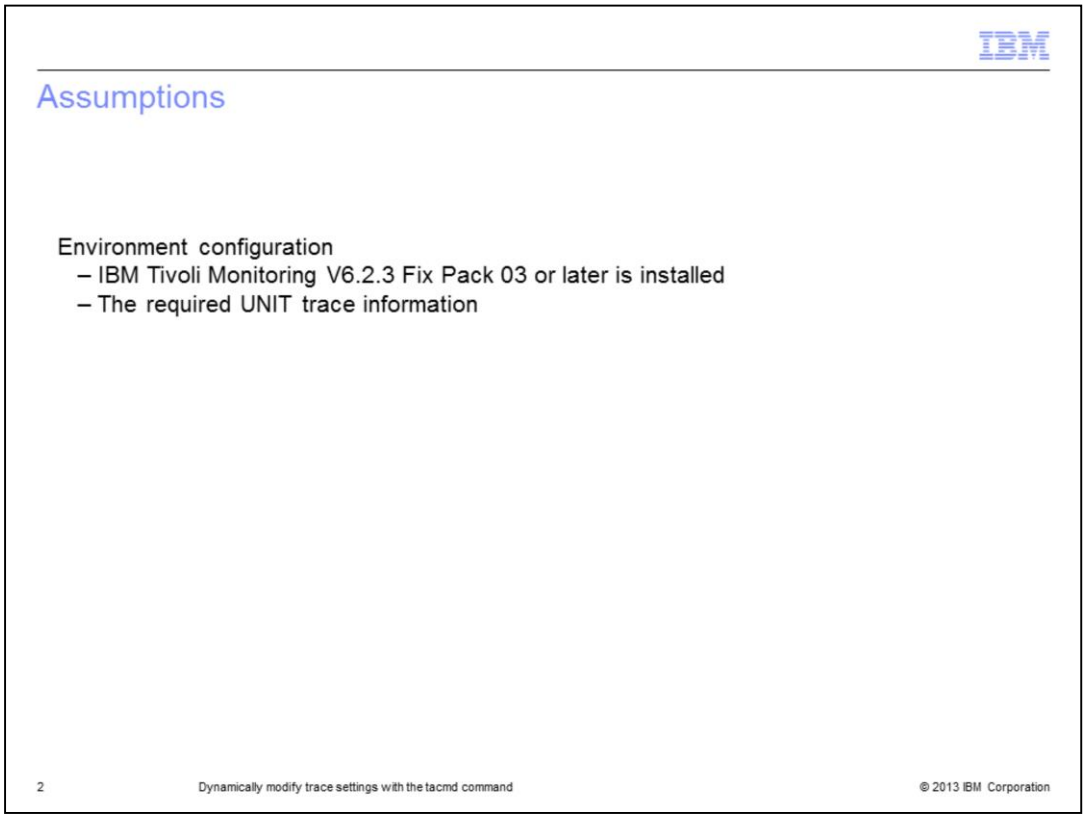
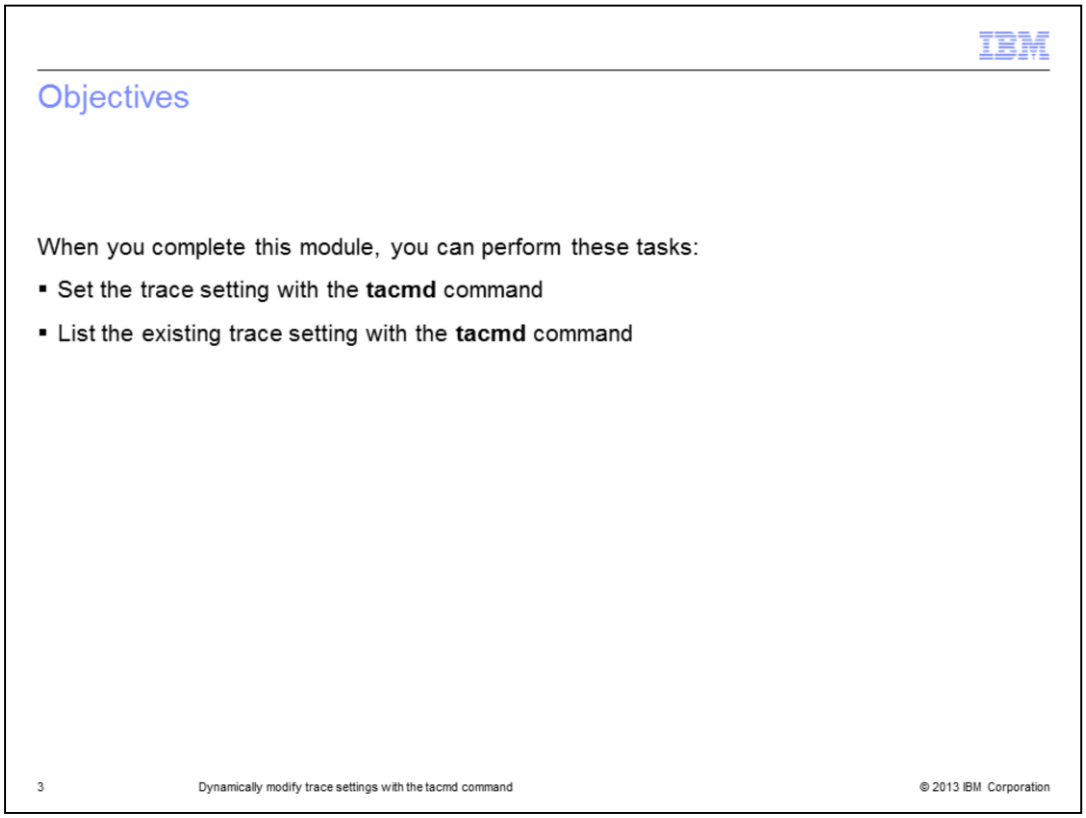



IBM Tivoli® Monitoring V6.2.3 Fix Pack 2 and later, Dynamically modify trace settings with the **tacmd** command line interface (CLI) command.

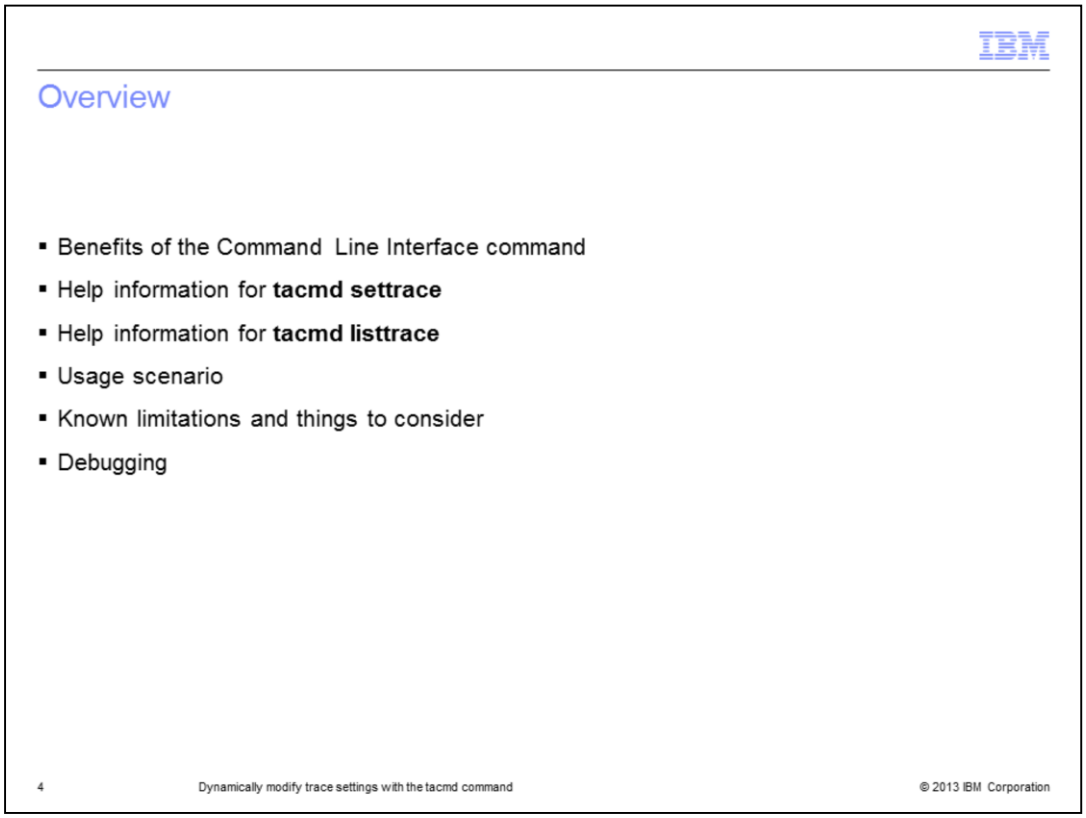



The assumptions for this module are that you have completed installing IBM Tivoli Monitoring V6.2.3 Fix Pack 02 or later. You should have a basic knowledge of the debugging UNIT trace.

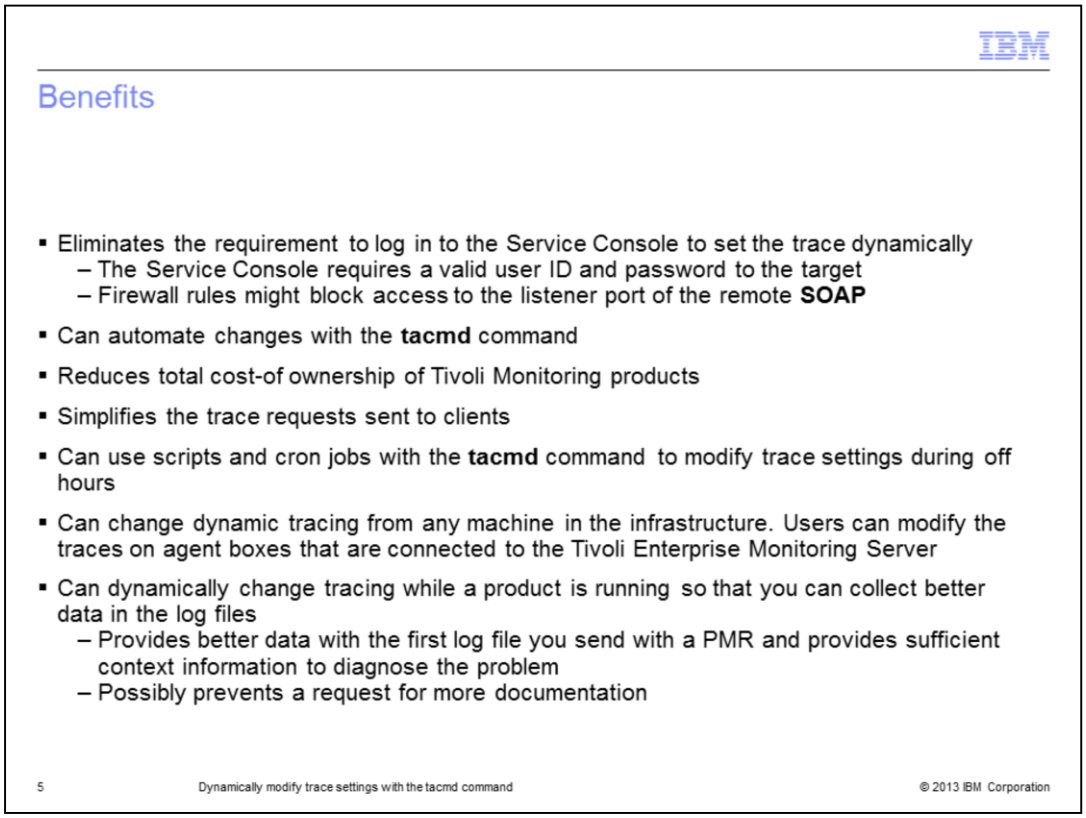
The IBM Tivoli Monitoring support team provides the UNIT trace.




When you complete this module, you can set and list the trace settings with the **tacmd** CLI command.




This module provides you with a brief introduction about setting a trace dynamically with the Command Line Interface (CLI) and the **tacmd** command. It explains the benefits of this command and the syntax for the commands. There is also information about the usage scenario and known limitations. This command is available in IBM Tivoli Monitoring V6.2.3 Fix Pack 02 and subsequent releases.




This slide explains the benefits of the tacmd CLI command.

The command eliminates the need to log in to the Service Console to set a trace dynamically. Sometimes it is not convenient or practical for a customer to use the Service Console on a remote endpoint. The Service Console also requires a valid user ID and password to the target system, whereas the tacmd command eliminates this requirement.

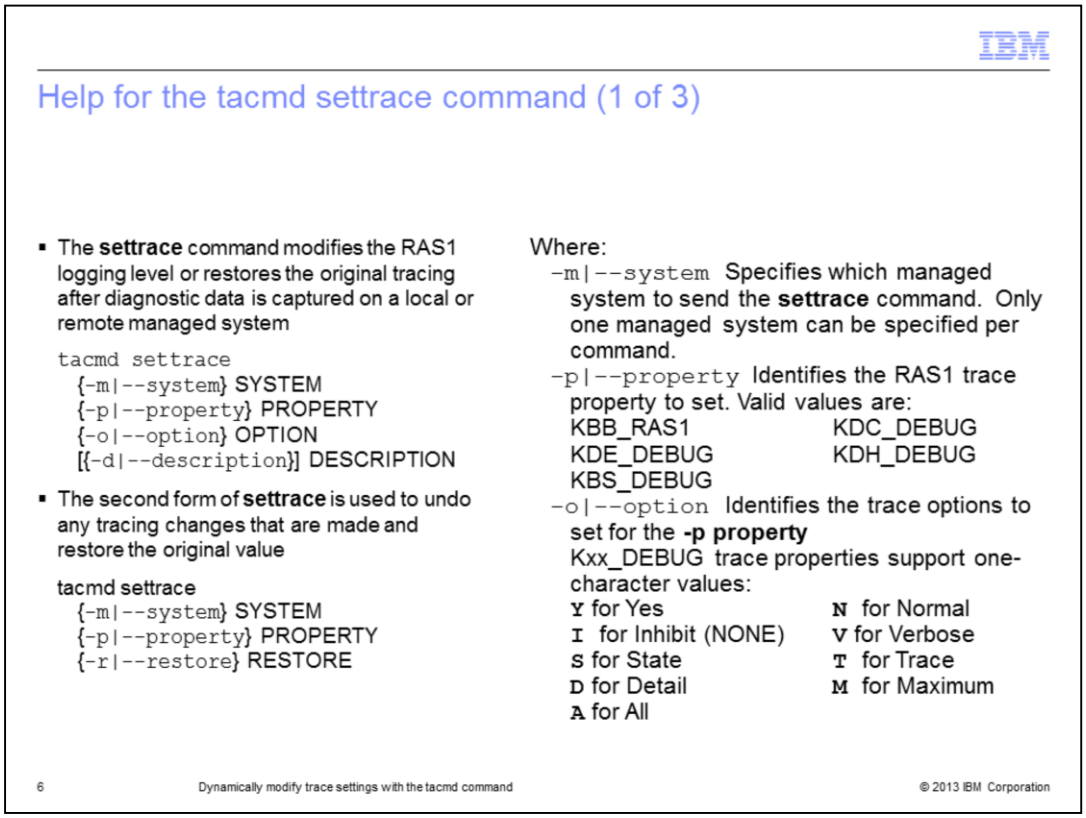
Firewall rules might block access to listener port on the remote SOAP while the CLI does not use any specific port.

You can automate changes with the tacmd command. Automation can provide these benefits:

- Reduce total cost-of ownership of Tivoli Monitoring products.

- Simplify the trace requests sent to clients. The tacmd settrace CLI makes it much easier to set requested traces. There are fewer chances for typographical errors and miscommunication. Customer Support can email a full tacmd settrace command string, that you can copy-and-paste into the command line.

- Scripts and cron jobs can be used with the tacmd command to modify trace settings during off hours with tacmd settrace. You can create a cron job or other time-driven scripts to change RAS1 logging at the time a sporadic problem most often occurs.




The **settrace** command modifies the RAS1 logging level or restores original tracing after diagnostic data is captured on a local or remote managed system. Only one managed system can be specified.

To set a trace dynamically, you need to specify the managed system where you want to set the trace, one of the properties, and any corresponding option for the property. The description is optional.

A hub Tivoli Enterprise Monitoring Server can be specified either with its CMS_NODEID value or as *HUB.

To undo any tracing changes, you must specify the managed system where you want to store the trace, one of the properties and -r to restore to the original value.

## Help for the tacmd settrace command (2 of 3)

- KBB_RAS1 supports more complex options; for help with KBB_RAS1, use these resources:
  - "Setting traces" in the *IBM Tivoli Monitoring Troubleshooting Guide*
  - Use the settings that are provided by your IBM Software Support representative
- The `-o` flag is not required if you specify `-r`
- `-d|--description` Specifies an optional description text string to help identify the reason for changing the trace variable. If a description string contains any embedded blanks, the entire string must be enclosed in double quotation marks
- `-r|--restore` Specifies that the RAS1 trace property should be restored back to the original value it had at product startup. If the -r flag is set, the -o and -d flags are ignored
- Examples
  - `tacmd settrace -m AIX_RTEMS -p KBB_RAS1 -o "ERROR (UNIT:kpx Error State)"`
  - `tacmd settrace -m SystemA:Agent1 -p KBB_RAS1 -r`
  - `tacmd settrace -m *HUB -p KDC_DEBUG -o Y -d "PMR 12345"`

```
# ./tacmd settrace -m itmmt100:TEPS -p KBB_RAS1 -o "ERROR (UNIT:kcq
DETAIL)"
  KUISTR000I: Command was performed successfully
```

7 Dynamically modify trace settings with the tacmd command 

This slide shows a few examples of setting the trace and the rules regarding KBB_RAS1 string specifications.

## Help for the tacmd settrace command (3 of 3)

Notes:

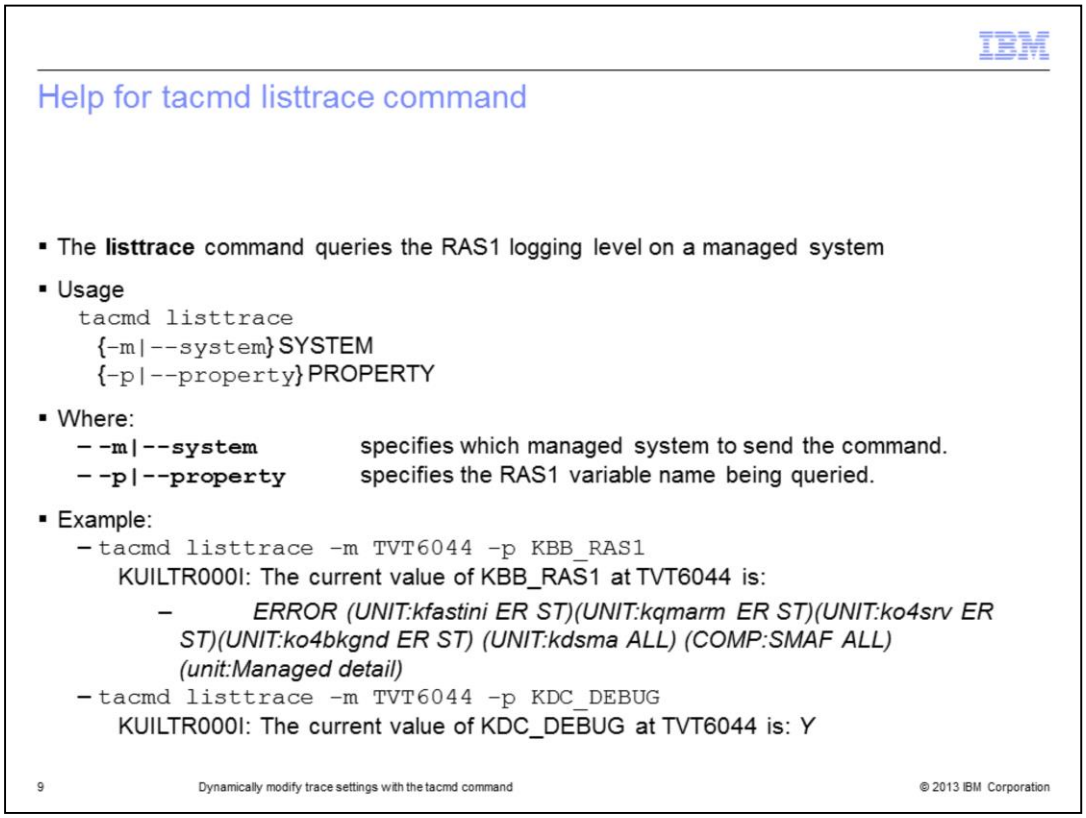
- You can specify a hub Tivoli Enterprise Monitoring Server either with its CMS_NODEID value or as *HUB
- Three rules for KBB_RAS1 string specifications:
  - (1) If there is no default class before the first (UNIT:xxx) parameter, the software assumes a default class of **ERROR**
  - (2) If the string contains embedded blanks, you must enclose the entire string in double quotation marks
  - (3) If the string contains embedded double quotation marks, you must enclose the entire string in outer double quotation marks

    Example

```
tacmd settrace -m zOS:CMS -p KBB_RAS1 -o "ERROR
(UNIT:KFASTINH,ENTRY:"KFA_InsertNodestatus" ALL)"
```




This slide shows a few final notes on the command.




The **listtrace** command queries the RAS1 logging level on a managed system.

To list the trace, you must specify the managed system where you want to list the trace and one of the properties names that you want to query. This example lists the value for the property KBB_RAS1 and KDC_DEBUG.


Here you see how you can set dynamic tracing in a typical customer environment with the CLI.

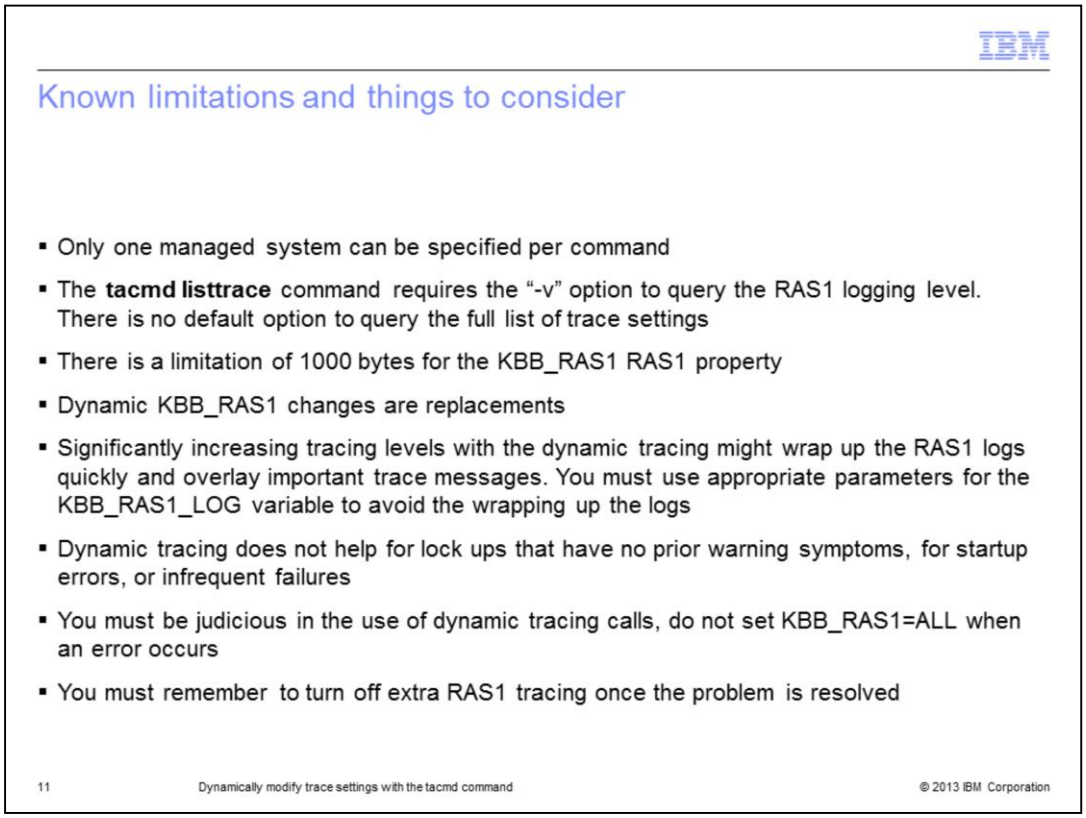
A dynamic KBB_RAS1 change is a replacement; it does not add to the existing value.

For example, the **.ini** file contains KBB_RAS1=ERROR. Level 2 support asked for you to turn on the **kxx** and **kyy** traces for <MSN>. Run the command **tacmd settrace -m <MSN> -p KBB_RAS1 -o "ERROR (UNIT:kxx ALL)(UNIT:kyy ALL)"**.

Later support requests for you to add the **kzz** tracing. Remember that the change is a replacement. The command is identical to the previous command plus the additional **kzz** information. The command looks like **tacmd settrace -m <MSN> -p KBB_RAS1 -o "ERROR (UNIT:kxx ALL) (UNIT:kyy ALL) (UNIT:kzz ALL)"**.

To avoid extra typing, you can copy-and-paste the output from a **tacmd listtrace** command against KBB_RAS1 to set the modified string.

After the diagnostic data is collected, you can revert to the minimal **ERROR-level** tracing. To revert, run the command **tacmd settrace -m <MSN> -p KBB_RAS1 -o ERROR**. This command clears all prior (UNIT:xxx) filters. It is not necessary to specify ANY to clear prior filters.




Here are the known limitations and the important aspects you should consider before setting the trace:

Only one managed system can be specified per command.

The **tacmd listtrace** command requires the "-p" option to query the RAS1 logging level. There is no default option to query the full list of trace settings.

There is a limitation of 1000 bytes for the KBB_RAS1 RAS1 property.

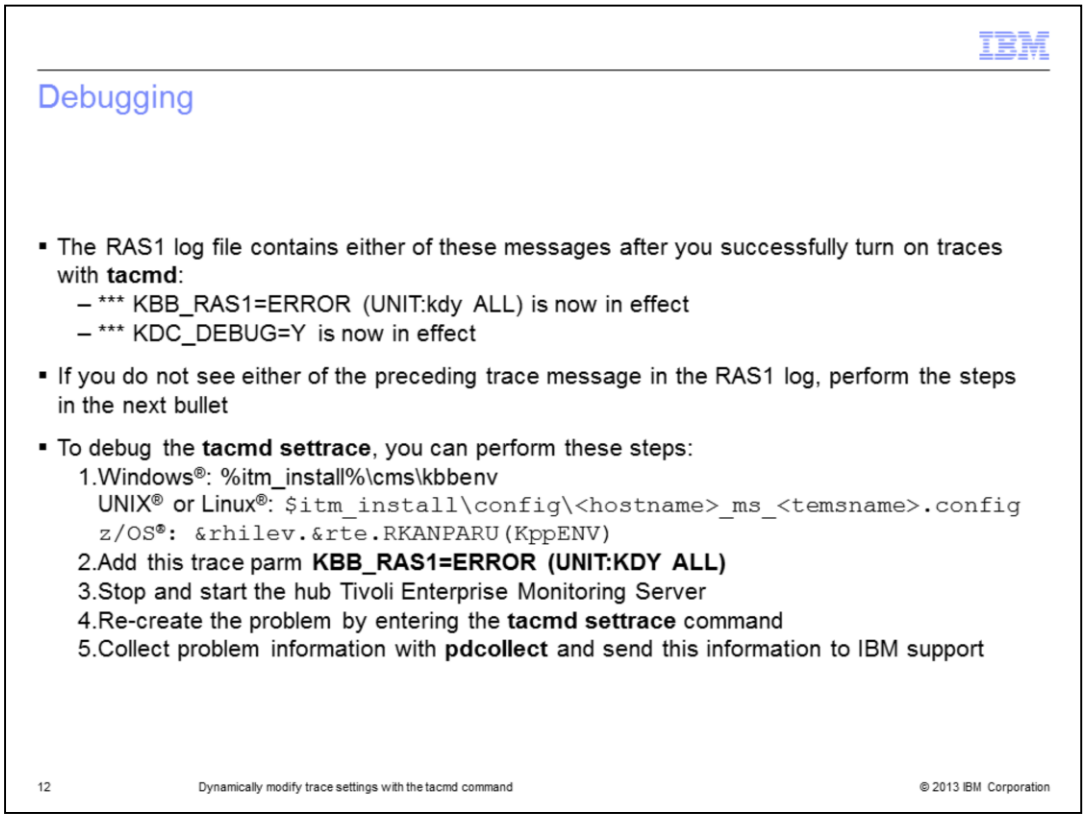
Dynamic KBB_RAS1 changes are replacements.

Significantly increasing tracing levels with dynamic tracing might wrap up the RAS1 logs quickly and overlay important trace messages. You must use appropriate parameters for the KBB_RAS1_LOG variable to avoid wrapping up the logs.

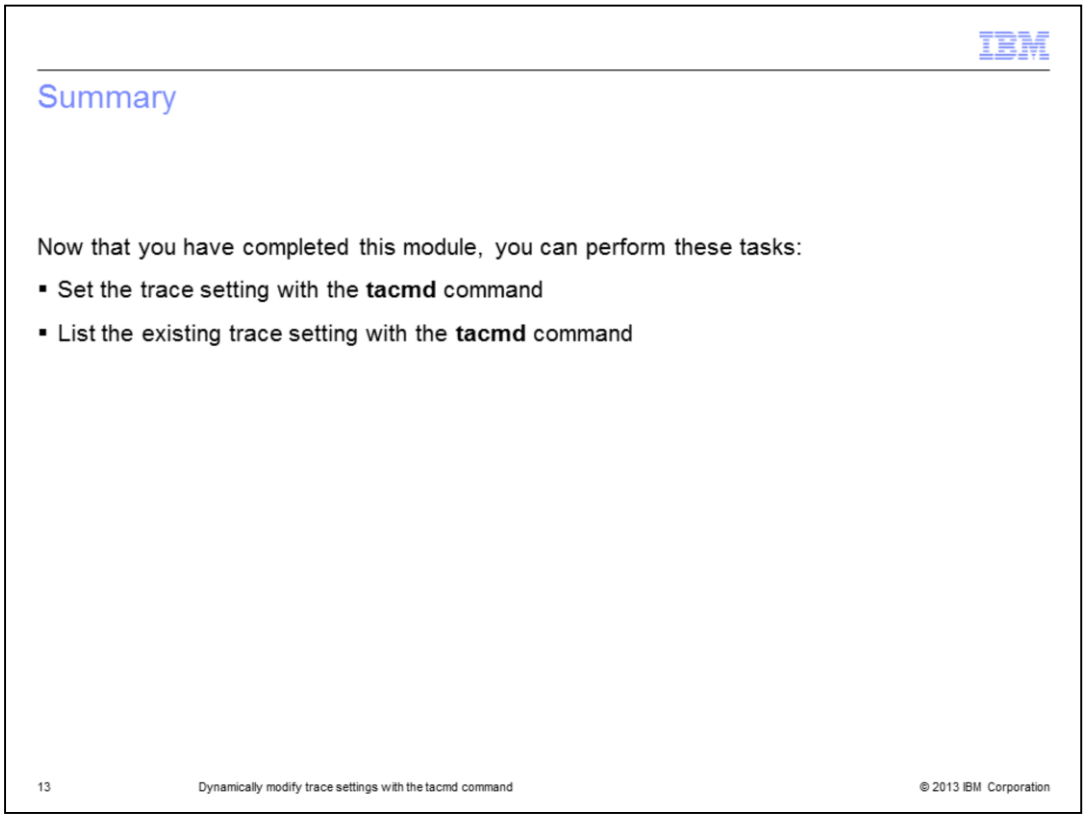
Dynamic tracing does not help for lock ups that have no prior warning symptoms, for startup errors, or infrequent failures.

You must be judicious in the use of dynamic tracing calls; do not set KBB_RAS1=ALL when an error occurs.

You must remember to turn off extra RAS1 tracing after the problem is resolved.




If you do not see either of the standing trace message in the RAS1 log, set the trace manually. Use the steps that are shown and contact IBM support with the information received from running **pdcollect** on the hub Tivoli Enterprise Monitoring Server.




Now that you have completed this module, you can set and list the trace settings with the **tacmd** CLI command.

# Trademarks, disclaimer, and copyright information