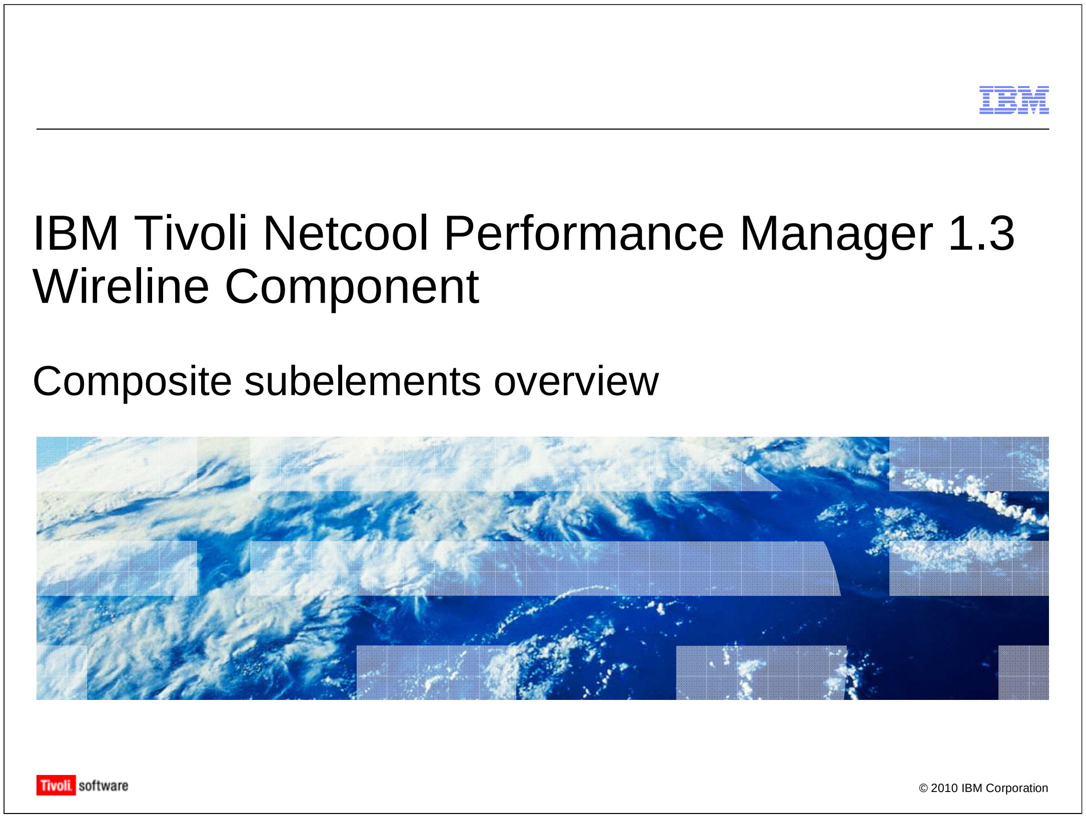
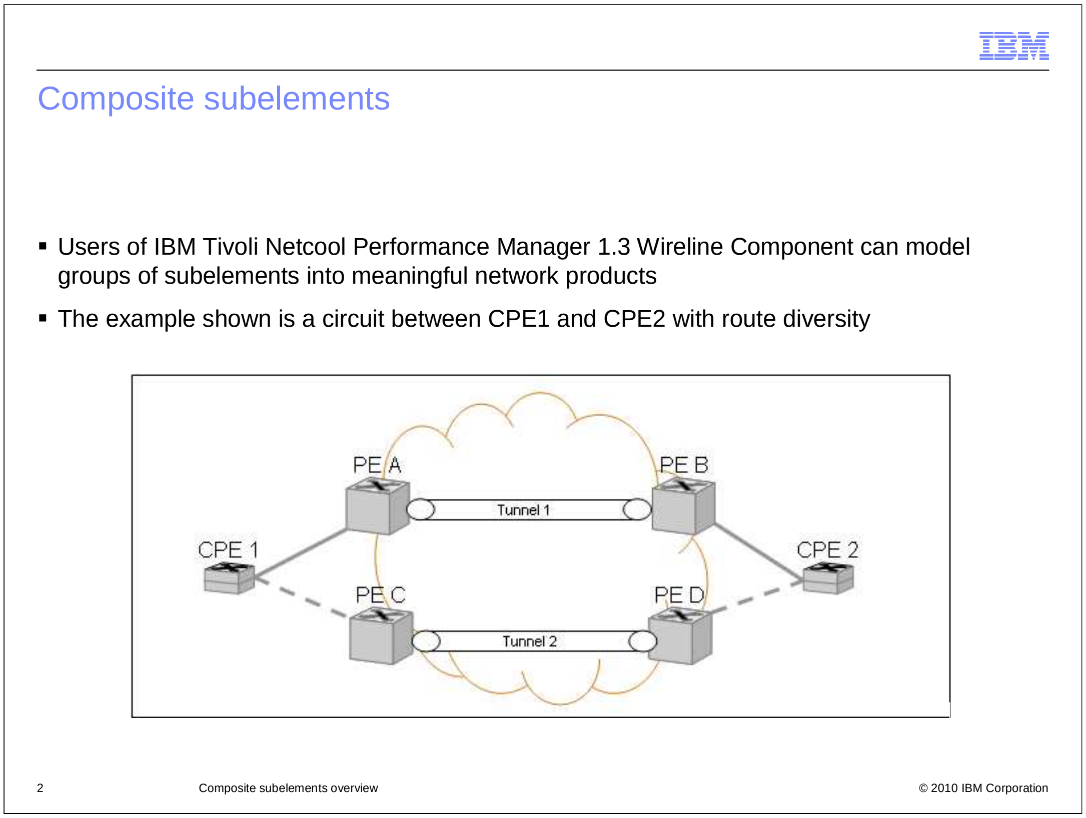


IBM Tivoli Netcool Performance Manager 1.3 Wireline Component

Composite subelements overview

This IBM Education Assistant module provides an overview of Composite Subelements (CSEs) in IBM Tivoli® Netcool® Performance Manager 1.3 Wireline Component.



Composite subelements

- Users of IBM Tivoli Netcool Performance Manager 1.3 Wireline Component can model groups of subelements into meaningful network products
- The example shown is a circuit between CPE1 and CPE2 with route diversity

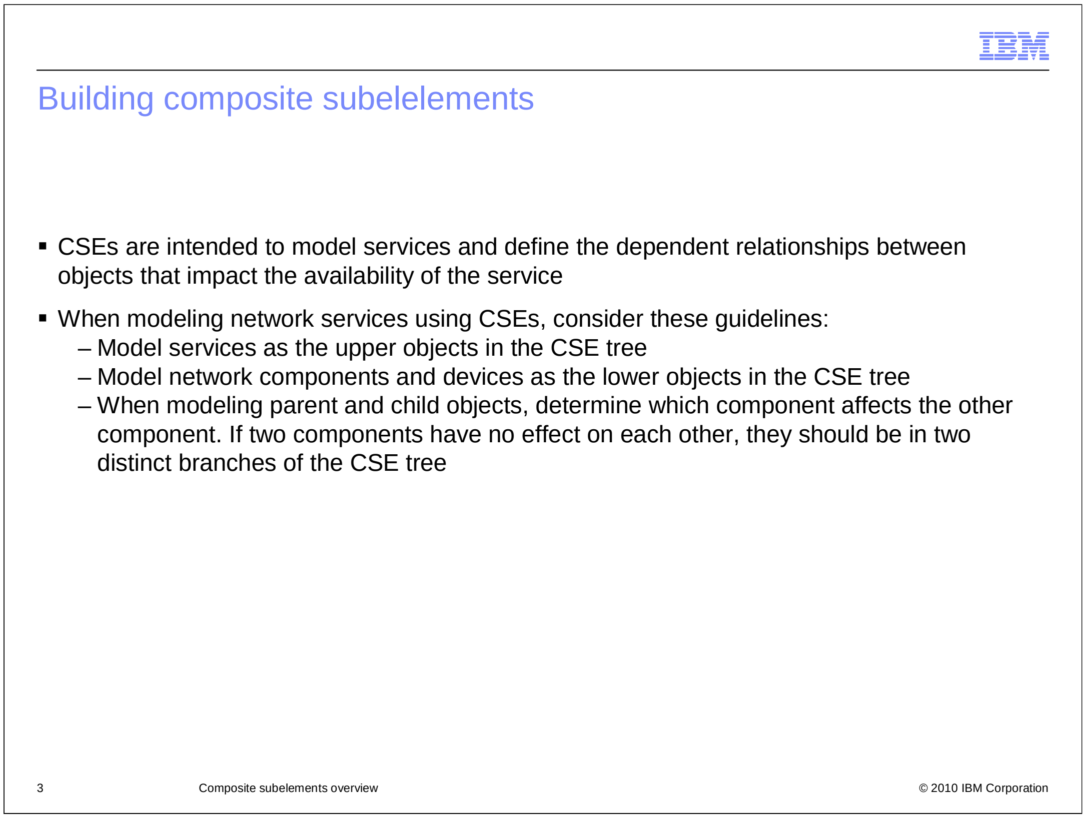
2 Composite subelements overview © 2010 IBM Corporation

IBM Tivoli Netcool Performance Manager Wireline Component has traditionally used groups to describe networks and services. While groups are scalable in different environments, they are limited because of a lack of structure between the subelements within groups. Subelements are placed into a group with no concepts of a hierarchical structure or how the subelements are related to each other. In practice, this implies that there is no first or last subelement. The only defined relationship is between each subelement and its parent group.

This approach is sufficient for modeling interfaces deployed for a customer, but groups cannot easily model more complex situations such as network paths or network services. Groups are suitable for modeling populations of subelements of the same type or family. They are not well adapted for storing heterogeneous subelements that rely on the definition of a relationship to provide an appropriate context.

Composite subelements are designed for scenarios that involve a more complex blend of interrelationships. CSEs are not intended as an alternative or a replacement to groups, but are a complementary method to be used in the appropriate situation. It is the responsibility of the network service manager who models new services to choose between groups or CSEs.

In this training module, you do not learn how to create or deploy CSEs. Instead, you learn how CSEs can affect the architecture of Tivoli Netcool Performance Manager Wireline Component. For more information about creating CSEs, see the Composite Subelement Configuration Guide.



Building composite subelelements

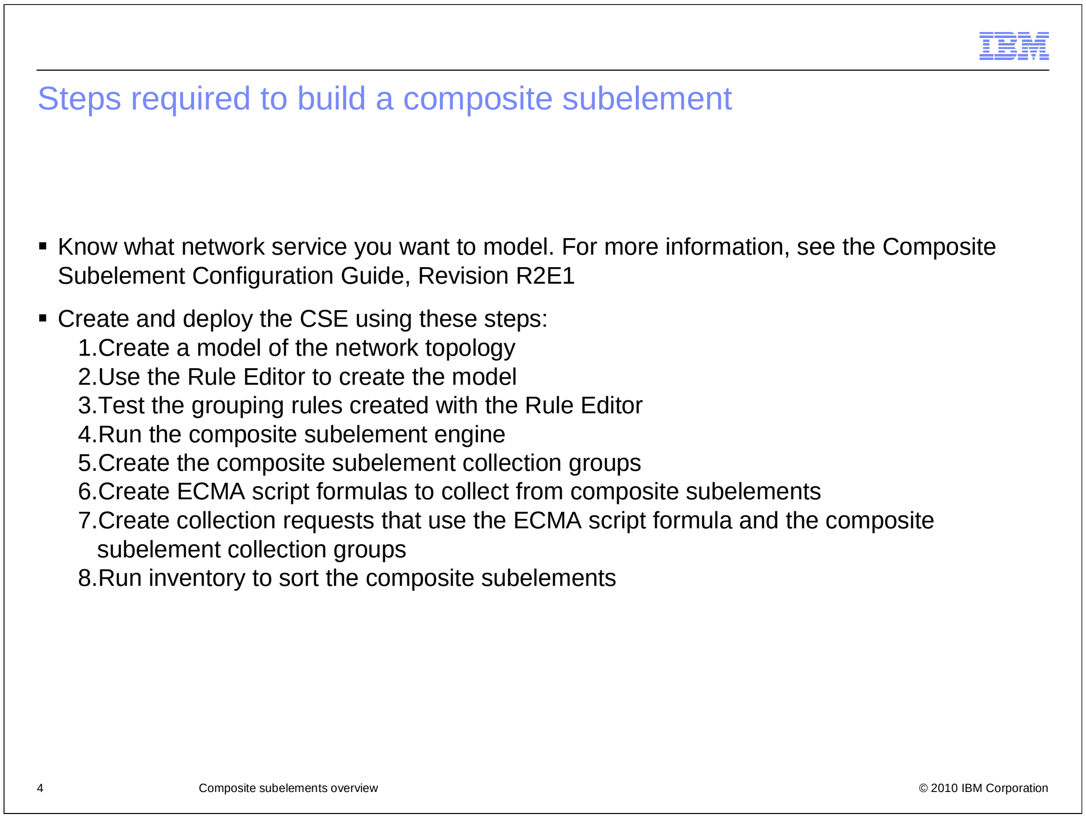
- CSEs are intended to model services and define the dependent relationships between objects that impact the availability of the service
- When modeling network services using CSEs, consider these guidelines:
  - Model services as the upper objects in the CSE tree
  - Model network components and devices as the lower objects in the CSE tree
  - When modeling parent and child objects, determine which component affects the other component. If two components have no effect on each other, they should be in two distinct branches of the CSE tree

3 Composite subelements overview © 2010 IBM Corporation

CSEs are for modeling services and defining the dependent relationships between objects that impact the availability of the service.

When modeling network services using CSEs, model services as the upper objects in the CSE tree, and model network components and devices as the lower objects in the CSE tree.

When modeling parent and child objects, determine which component affects the other component. If two components have no effect on each other, they should be in two distinct branches of the CSE tree.



Steps required to build a composite subelement

- Know what network service you want to model. For more information, see the Composite Subelement Configuration Guide, Revision R2E1
- Create and deploy the CSE using these steps:
  1. Create a model of the network topology
  2. Use the Rule Editor to create the model
  3. Test the grouping rules created with the Rule Editor
  4. Run the composite subelement engine
  5. Create the composite subelement collection groups
  6. Create ECMA script formulas to collect from composite subelements
  7. Create collection requests that use the ECMA script formula and the composite subelement collection groups
  8. Run inventory to sort the composite subelements

4 Composite subelements overview © 2010 IBM Corporation

CSE structures are created in Tivoli Netcool Performance Manager Wireline Component using the DataMart Rule Editor. After a structure has been created, Complex Metric Engine (CME) formula files are written in ECMA script language. They are deployed against the composite subelement structure. The formulas that are defined within the file use the composite subelement definitions to navigate the structure and retrieve the appropriate input metrics for their calculations.

Several steps are required to create a composite subelement. First, you must know what network service you want to model. You can find more information in the Composite Subelement Configuration Guide.

The steps needed to create and deploy a CSE are to:

1. Create a model of the network topology.

2. Use the Rule Editor to create the model.

3. Test the grouping rules created with the Rule Editor.

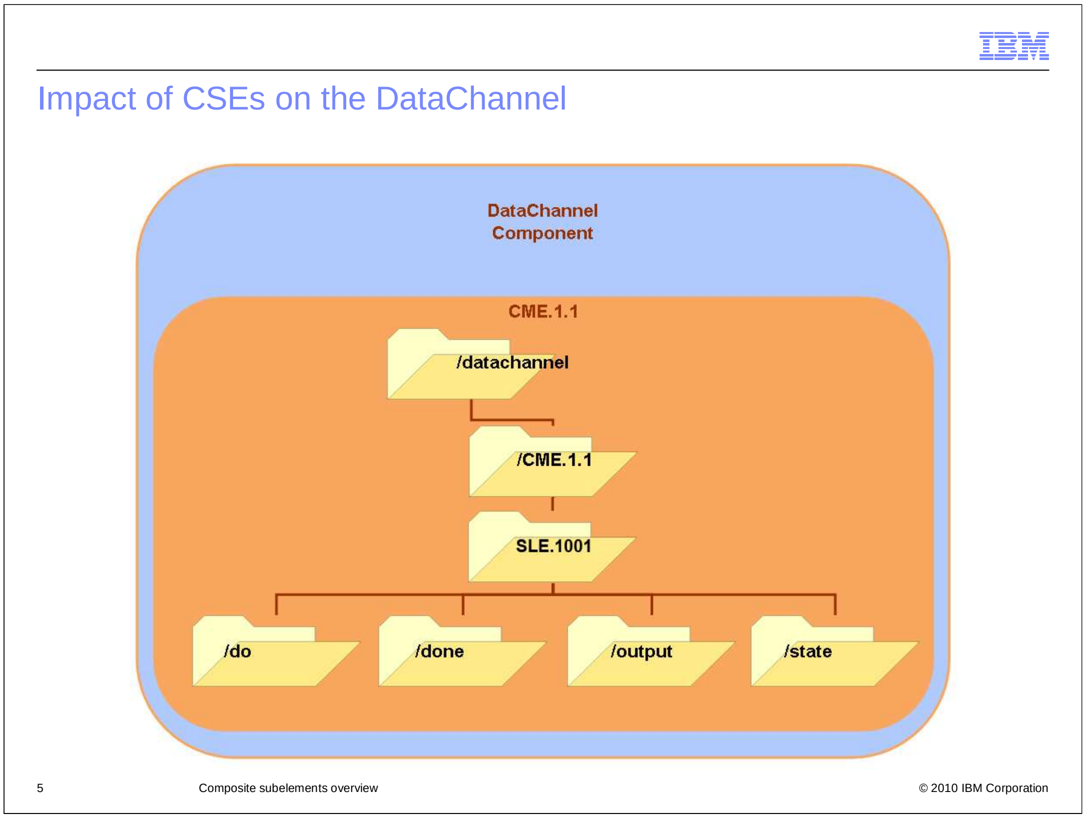
4. Run the composite subelement engine.

5. Create the composite subelement collection groups.

6. Create ECMA script formulas to collect from composite subelements.

7. Create collection requests that use the ECMA script formula and the composite subelement collection groups.

8. Run inventory to sort the composite subelements.



Impact of CSEs on the DataChannel

5 Composite subelements overview © 2010 IBM Corporation

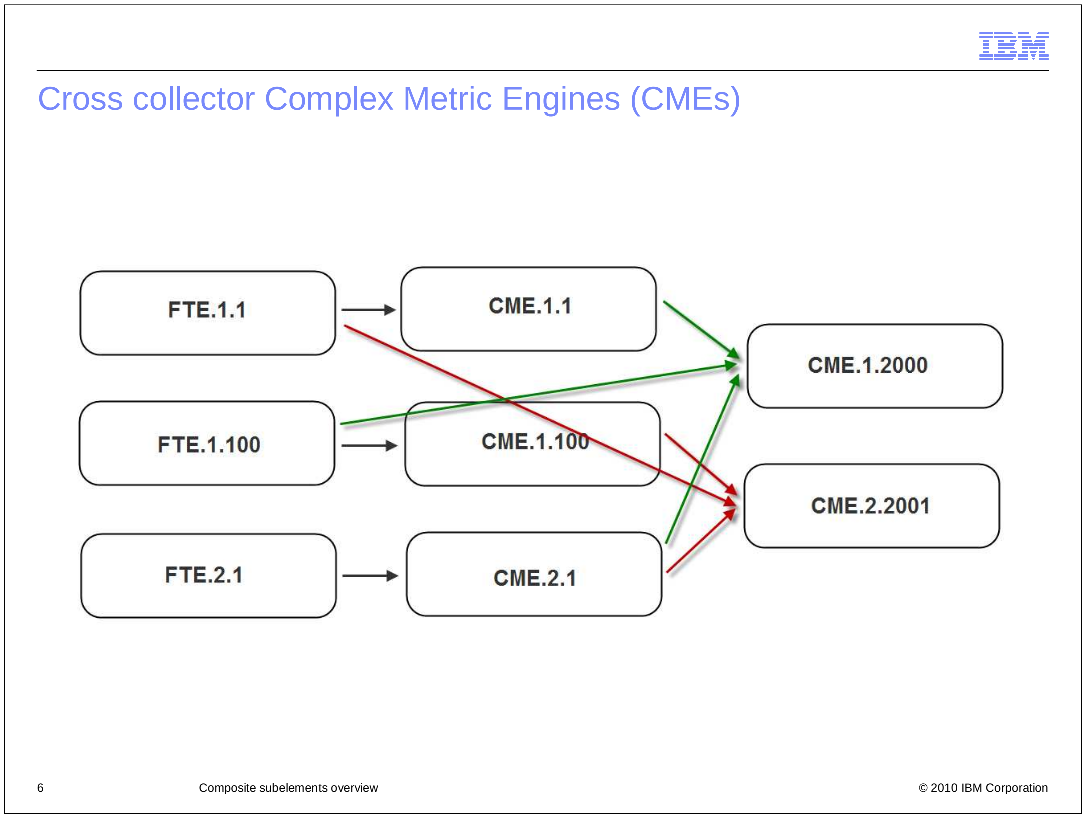
The CME directory path contains a new directory structure between the root directory and the **/do**, **/done**, **/output**, and **/state** directories. The directory structure acquires its name from a concept called Single Label Engine (SLE). There is one SLE directory structure per CME in Tivoli Netcool Performance Manager 1.3 Wireline Component.

As an example, /CME.1.1 has a subdirectory named SLE.1001. The CME directory /CME.1.100 has a subdirectory called SLE.1100.

The **CME.1.1 /SLE.1001/do** directory holds data moved from the FTE. Typically, there is only one metric data file in the **/do** directory at a time.

The product also uses a new tool called Plan Builder (PBL) that runs as an application in the DataChannel. The PBL synchronizes inventory and metadata. It also identifies computation clusters (the SLEs) for CMEs using a labeling algorithm. The PBL assigns cross-resource formulas to dynamic (2000 level) CMEs. There is only one PBL per installation.

The dynamic CMEs are also known as Cross Collector CMEs. These CMEs can be created and deployed using the topology editor.



Cross collector Complex Metric Engines (CMEs)

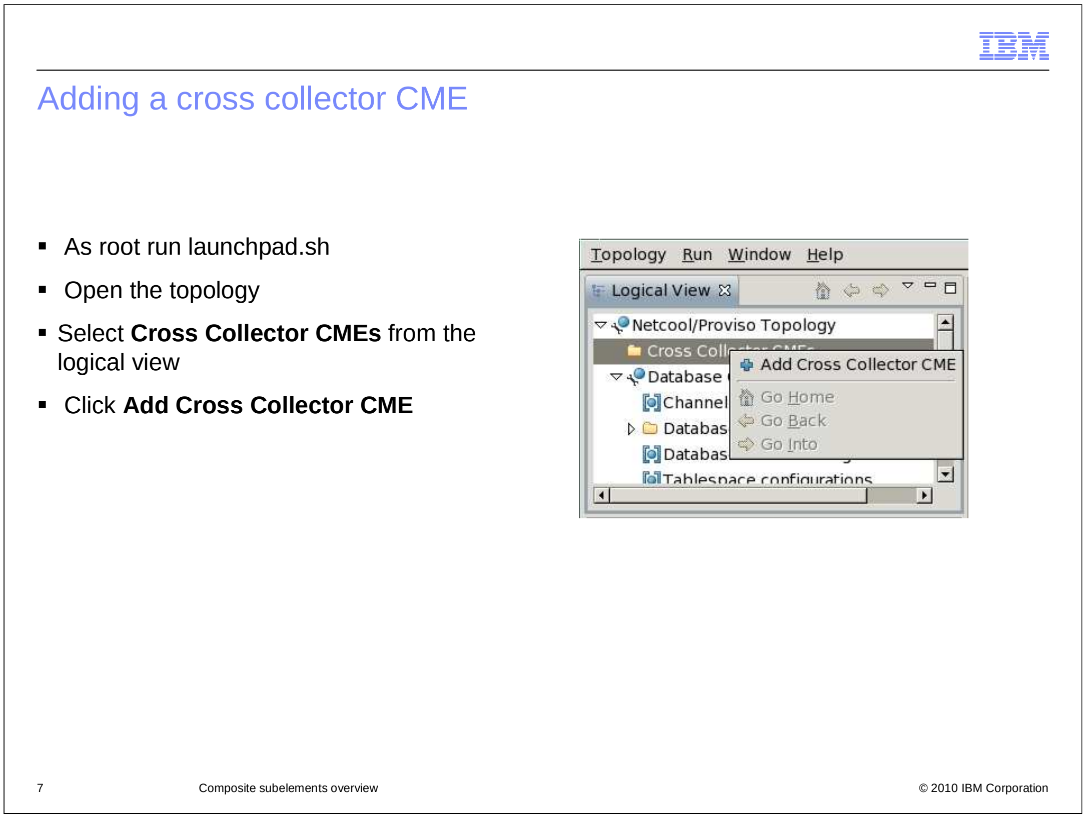
6 Composite subelements overview © 2010 IBM Corporation

IBM Tivoli Netcool Performance Manager Wireline Component requires a cross collector/channel CME capability. A composite subelement can consist of metric data from many FTEs and CMEs and from many channels. To use CSEs in an installation, at least one cross collector CME must be created.

In the illustration, the cross collector CME, CME.1.2000, receives data from CME.1.1, FTE.1.100, and CME.2.1. Cross collector CME numbering starts with CME.x.2000. In this case, CME.1.2000 is the first cross collector CME created in this installation.

The cross collector CME.2.2001, the second cross collector created in the installation, takes input from CME.1.100, CME.2.1, and FTE.1.1.

The aggregations computed by the cross collector CMEs are forwarded to the LDR and DLDR. The channel number that the cross collector CME is associated with is specified. The output of the cross collectors CMEs is treated exactly like regular CMEs.



Adding a cross collector CME

- As root run launchpad.sh
- Open the topology
- Select **Cross Collector CMEs** from the logical view
- Click **Add Cross Collector CME**

To add a cross collector to handle CSEs, you use the Topology Editor and the Deployer.

To add a cross collector CME, you must run either the launchpad.sh or Topology Editor as root.

1. To run the launchpad.sh, go to the installation root directory and locate the launchpad.sh:

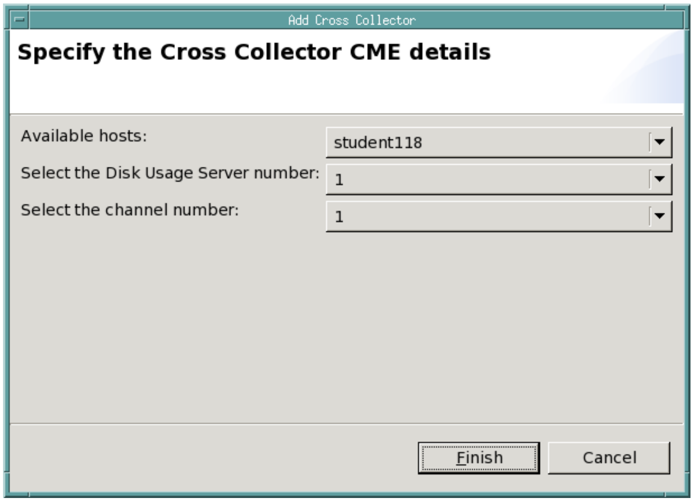
   # ./launchpad.sh

2. Open the topology to modify.

3. Highlight and right-click **Cross Collector CMEs** under **Netcool/Proviso Topology**.

4. Click **Add Cross Collector CME**.

Specify the cross collector CME details



Specify the Cross Collector CME details

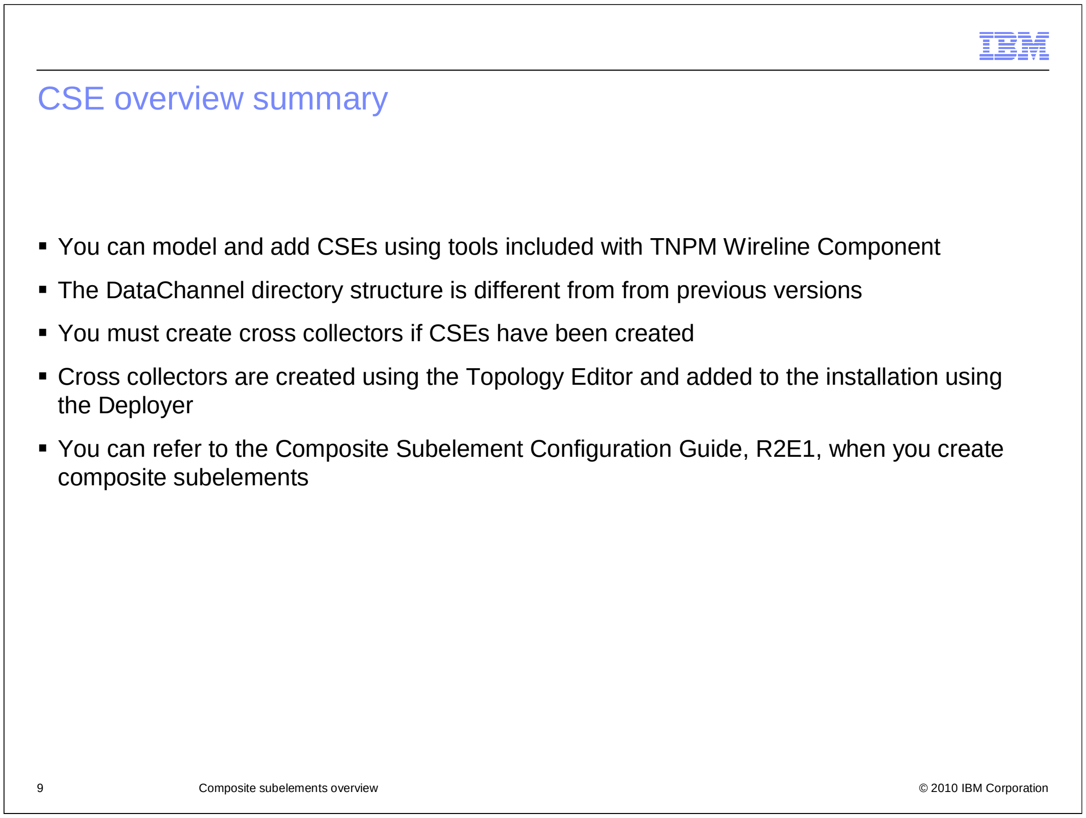
Available hosts: student118

Select the Disk Usage Server number: 1

Select the channel number: 1

Specify the host to install the cross collector CME on, the Disk Usage Server that it will use for flow control, and the channel number. You must then deploy the cross collector CME.

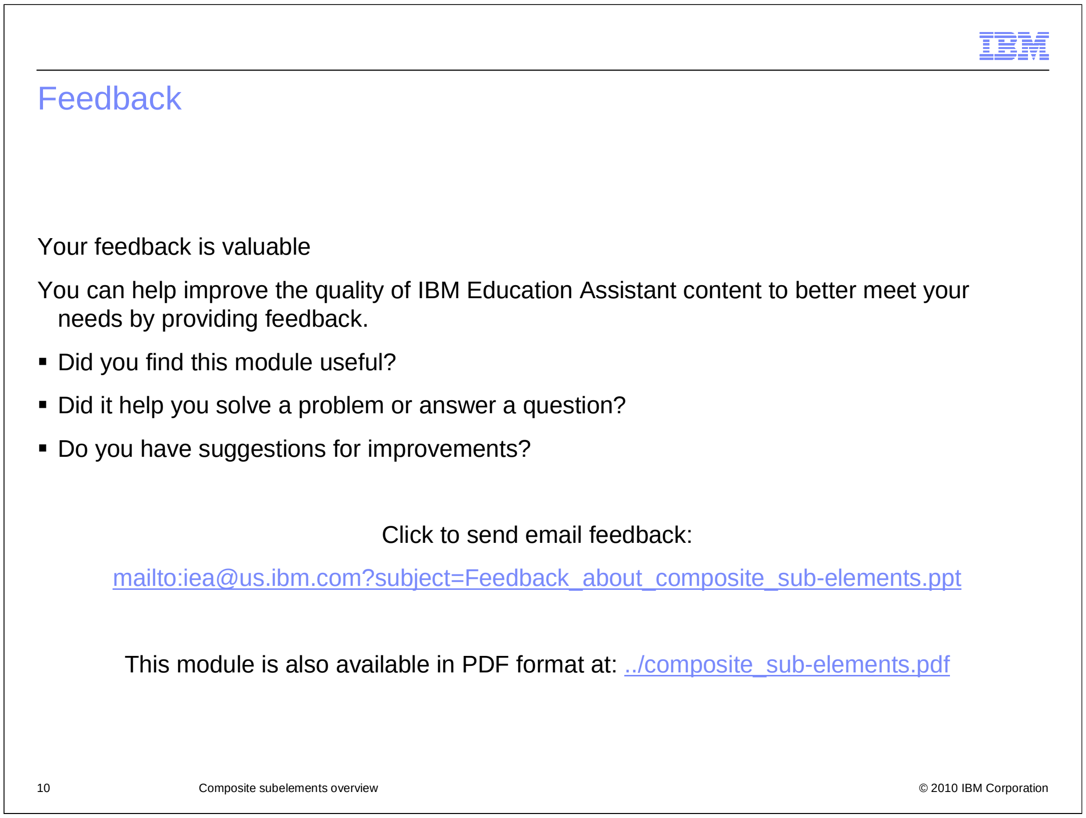
Cross collector CMEs should only be deployed when they are required to handle composite subelements.



CSE overview summary

- You can model and add CSEs using tools included with TNPM Wireline Component
- The DataChannel directory structure is different from from previous versions
- You must create cross collectors if CSEs have been created
- Cross collectors are created using the Topology Editor and added to the installation using the Deployer
- You can refer to the Composite Subelement Configuration Guide, R2E1, when you create composite subelements

In summary, you can model and add CSEs using tools included with Tivoli Netcool Performance Manager Wireline Component. Refer to the Composite Subelement Configuration Guide, for more information about creating composite subelements.



Feedback

Your feedback is valuable

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

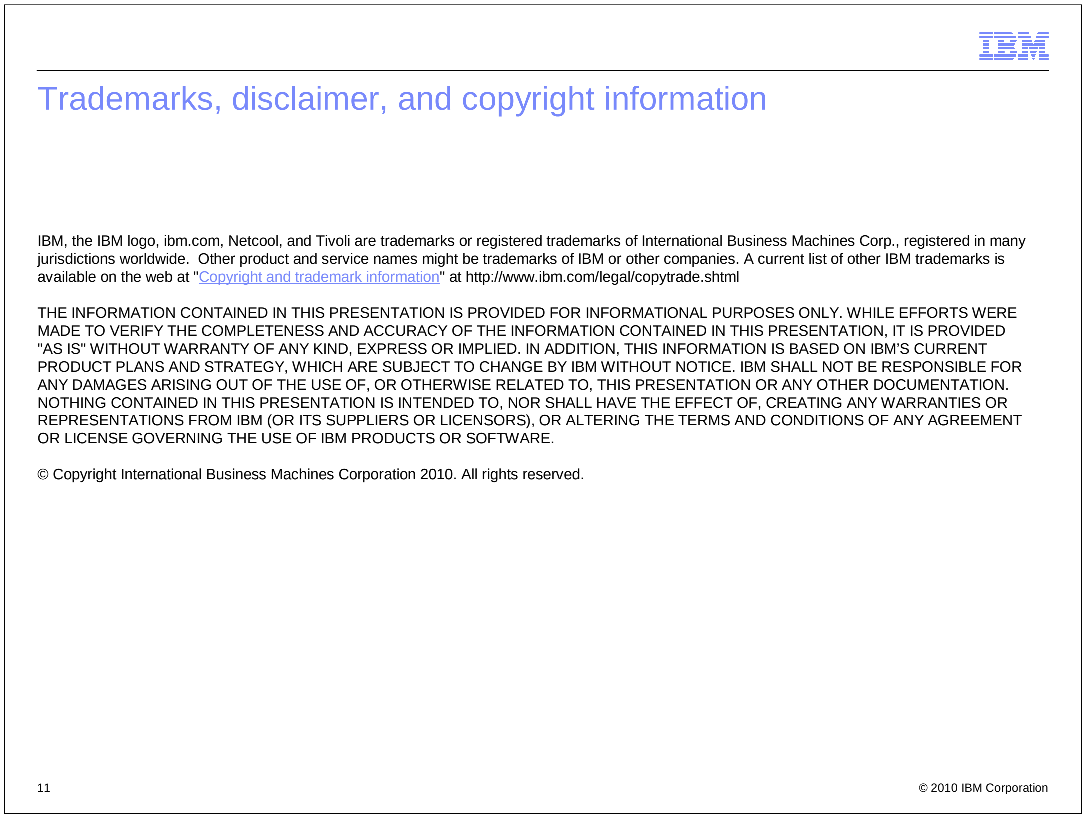
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_composite_sub-elements.ppt

This module is also available in PDF format at: ../composite_sub-elements.pdf

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



Trademarks, disclaimer, and copyright information

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.