



December 3-6, 2007, Santa Clara Marriott, Santa Clara, CA

# CDM Conformance Testing

Ravindra Muddinagiri

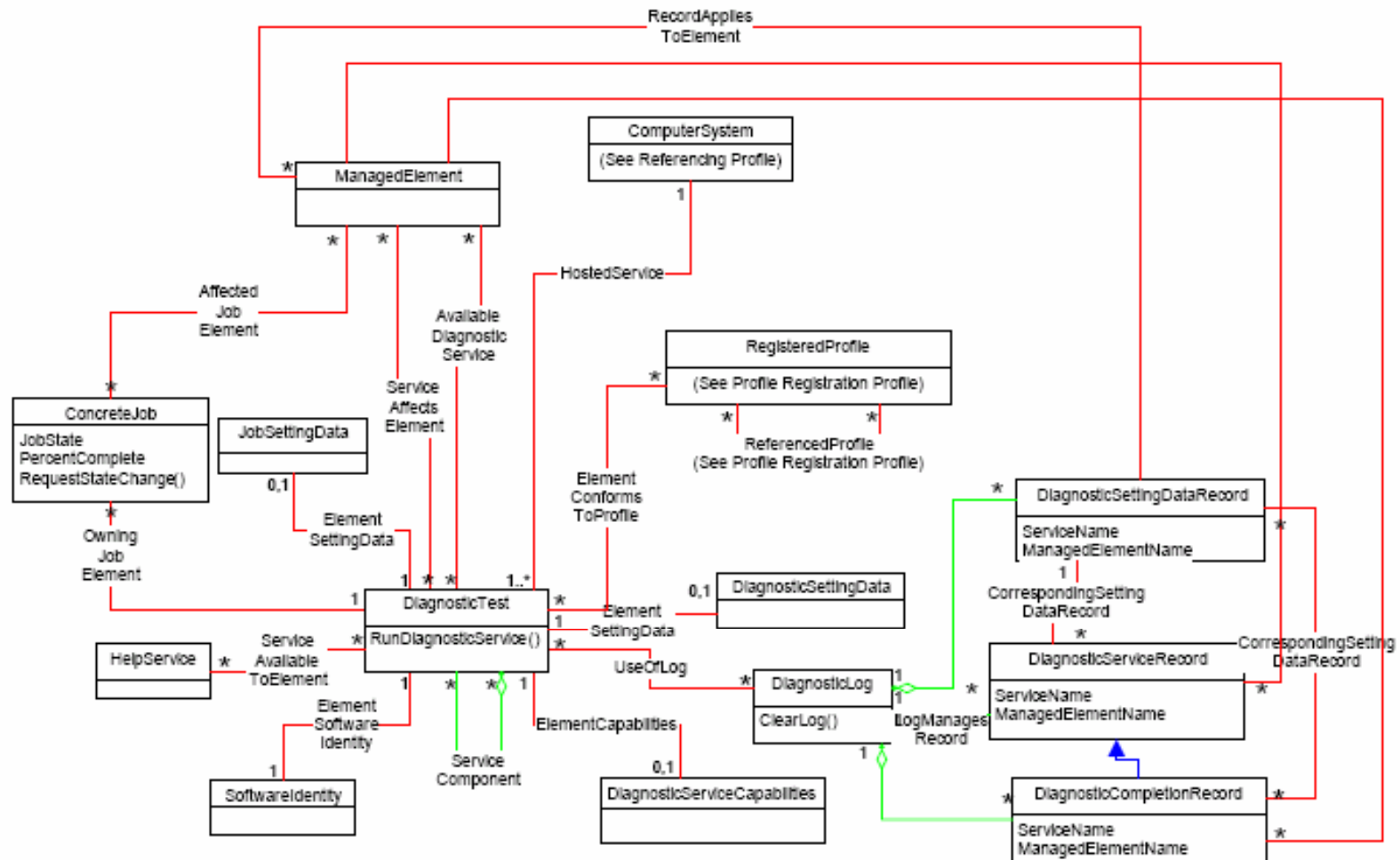




## Presentation at a Glance

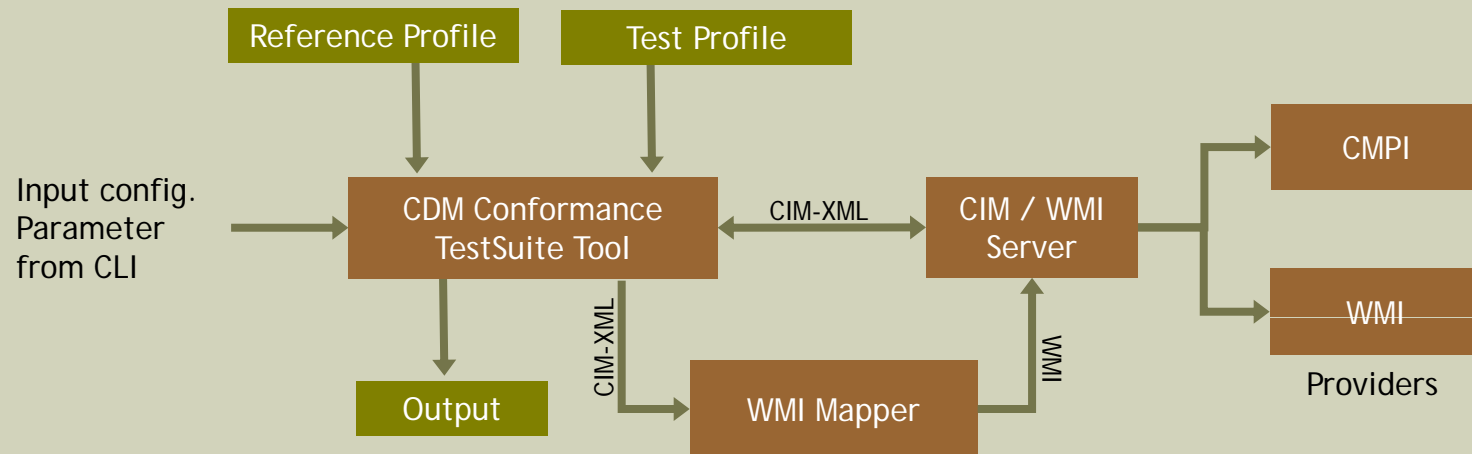
- Brief introduction to the CDM profile
- CDM Conformance Test Tool architecture
- Releases of the Test Tool
  - ❖ Static tests
  - ❖ Release 2a - Dynamic Testing/non-persistent classes
  - ❖ Release 2b - Extrinsic Method Testing/persistent and non-persistent classes
  - ❖ Release 3 - Behavior Testing
  - ❖ Release 4 - Class and Property Consistency Testing
  - ❖ Release 5 - Extrinsic Method Consistency Testing
  - ❖ Release 6 - Optional Implementation Testing
  - ❖ Release 7 - Linux and Windows testing

# CDM Profile



Copyright (c) 2007 DMTF. All Rights Reserved

# CDM Testing Tool architecture



CDM Testing Tool Architecture

- Testing of the CDM provider for conformance to CDM profile specifications
- Uses Pegasus Client library for generation of CIM-XML request messages for communication with CIMOM
- Validates the response messages received and generates test results file
- Generates html reports
- Generates digitally signed XML results file
- Designed for easy extensibility.



# Releases of the Conformance Test Tool

## Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

- Includes two tests:
  - Static Test before running the diagnostic service
  - Static Test after running the diagnostic service
- The tests traverse the classes and associations.
- Validates the properties for valid ranges
- Validates the methods for correct arguments

```
<CLASSES>
<CLASS NAME="CIM_DiagnosticsLog" />
<CLASS NAME="CIM_LogRecord" />
<CLASS NAME="CIM_DiagnosticTest">
<QUALIFIER NAME="TopStart" TYPE="string">
<VALUE>true</VALUE>
</QUALIFIER>
<QUALIFIER NAME="Required" TYPE="boolean">
<VALUE>true</VALUE>
</QUALIFIER>
</CLASS>
</CLASSES>
```



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

Dynamic Testing is used for validating the transient classes created during the service test run

An example of such class is CIM\_ConcreteJob

```
<CLASSES>
<CLASS NAME="CIM_ConcreteJob">
<QUALIFIER NAME="Required" TYPE="boolean">
<VALUE>true</VALUE>
</QUALIFIER>
</CLASS>
<CLASS NAME="CIM_DiagnosticTest">
<QUALIFIER NAME="TopStart" TYPE="string">
<VALUE>true</VALUE>
</QUALIFIER>
<QUALIFIER NAME="Required" TYPE="boolean">
<VALUE>true</VALUE>
</QUALIFIER>
</CLASS>
<CLASS NAME="CIM_DiagnosticServiceRecord" />
<CLASS NAME="CIM_DiagnosticSettingRecord" />
<CLASS NAME="CIM_LogRecord" />
</CLASSES>
```



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

- Testing of extrinsic functions of the classes in the profile
- Parameters are configured using XML files

```
<METHOD NAME="ClearLog" TYPE="uint32">
<QUALIFIER NAME="TestID" TYPE="string">
<VALUE>TST-019</VALUE>
</QUALIFIER>
<QUALIFIER NAME="Description" TYPE="string">
<VALUE>Invoke ClearLog when Log entries exist</VALUE>
</QUALIFIER>
<QUALIFIER NAME="Precondition" TYPE="string">
<VALUE>SupportedLog Storage should support RecordLog</VALUE>
</QUALIFIER>
<QUALIFIER NAME="ExpectedResult" TYPE="string">
<VALUE>Return code will be 0(Request was successfully executed)</VALUE>
</QUALIFIER>
</METHOD>
```



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

**Release 3 - Behavior Testing**

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

## Testing the Use Cases of the CDM profile

- Discover Available Diagnostics

`GetAllDiagnostics()`, `GetAllDiagnosticMEPairs()`,  
`GetDiagnosticsForME()`, `GetMEsForDiagnostic()`,  
`GetCharacteristicsOfDiagnostic()`,  
`GetDiagnosticsWithCharacteristicsForME()`,  
`GetPackageSubtests()`.

- Execute and Control Diagnostic

`RunDiagnostic()`, `SuspendDiagnostic()`, `ResumeDiagnostic()`,  
`AbortDiagnostic()`, `KillDiagnostic()`

- Discover Diagnostic Executions

`GetSpecificDiagnosticExecutions()`

- Discover Diagnostic Results

`GetLogRecordsForDiagnostic()`, `GetLogRecordsForME()`,  
`GetLogRecordsForMEAndDiagnostic()`,  
`GetDiagnosticExecutionResults()`,  
`GetDiagnosticExecutionFinalResults()`,  
`GetDiagnosticProgress()`



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

Testing the consistency of the classes and properties of the CDM profile

Examples:

- Class Consistency - Checking the existence of CompletionRecord when the ConcreteJob is created
- Property Consistency - Checking whether ConcreteJob.ErrorDescription is filled in if ConcreteJob.ErrorCode is implemented

```
<TESTCASE>
<QUALIFIER NAME="TESTID" TYPE="string">
<VALUE>TST-006</VALUE>
</QUALIFIER>
<QUALIFIER NAME="PropertyConsistency" TYPE="boolean">
<VALUE>TRUE</VALUE>
</QUALIFIER>
<QUALIFIER NAME="Description" TYPE="string">
<VALUE>Tests the ResultantProperty CIM_ConcreteJob.ErrorDescription if
  CIM_ConcreteJob.ErrorCode is implemented.</VALUE>
</QUALIFIER>
<QUALIFIER NAME="ConditionalProperty" TYPE="string">
<VALUE>CIM_ConcreteJob.ErrorCode</VALUE>
</QUALIFIER>
<QUALIFIER NAME="ResultantProperty" TYPE="string">
<VALUE>CIM_ConcreteJob.ErrorDescription</VALUE>
</QUALIFIER>
</TESTCASE>
```



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

**Release 5 - Extrinsic Method Consistency Testing**

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

## Consistency test for methods

If method is executed that is expected to create an instance, the newly created instance is validated for consistency

```
<QUALIFIER NAME="InitialJobState" TYPE="uint16">
<VALUE>4</VALUE>
</QUALIFIER>
<QUALIFIER NAME="ConditionalProperties" TYPE="string">
<VALUE.ARRAY>
<VALUE>CIM_ConcreteJob.ErrorCode{</VALUE>
</VALUE.ARRAY>
</QUALIFIER>
<QUALIFIER NAME="ResultantProperties" TYPE="string">
<VALUE.ARRAY>
<VALUE>CIM_ConcreteJob.ErrorDescription</VALUE>
</VALUE.ARRAY>
</QUALIFIER>
<QUALIFIER NAME="ExpectedResult" TYPE="string">
<VALUE>Return code will be 0 and ConcreteJob.Jobstate will be
Killed(9)</VALUE>
</QUALIFIER>
<QUALIFIER NAME="RequestedState" TYPE="uint16">
<VALUE>5</VALUE>
</QUALIFIER>
</TESTCASE>
```

(CONTINUED)...



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing (contd...)

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

( CONTINUED )...

```
<QUALIFIER NAME="InitialJobState" TYPE="uint16">
<VALUE>4</VALUE>
</QUALIFIER>
<QUALIFIER NAME="ConditionalProperties" TYPE="string">
<VALUE.ARRAY>
<VALUE>CIM_ConcreteJob.ErrorCode{}</VALUE>
</VALUE.ARRAY>
</QUALIFIER>
<QUALIFIER NAME="ResultantProperties" TYPE="string">
<VALUE.ARRAY>
<VALUE>CIM_ConcreteJob.ErrorDescription</VALUE>
</VALUE.ARRAY>
</QUALIFIER>
<QUALIFIER NAME="ExpectedResult" TYPE="string">
<VALUE>Return code will be 0 and ConcreteJob.Jobstate will be
Killed(9)</VALUE>
</QUALIFIER>
<QUALIFIER NAME="RequestedState" TYPE="uint16">
<VALUE>5</VALUE>
</QUALIFIER>
</TESTCASE>
```



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing

**Release 6 - Optional Implementation Testing**

Release 7 - Linux and Windows testing

Optional Testing includes:

- **Extrinsic method testing**
  - GetHelp()
- **Behavior testing**
  - GetCapabilitiesOfDiagnostic()
  - GetDiagnosticswithCapabilitiesForME()
  - CreateDiagnosticSettings()
  - GetDefaultJobSettings()
  - CreateJobSettings()
  - GetAffectedMEs()
  - GetAllDiagnosticExecutionsForME()
  - GetSpecificDiagnosticExecutionsForME()
  - GetDiagnosticExecutionSettings()
- **Consistency testing**

If the classes DiagnosticSetting and DiagnosticsLog exist, validate the existence of Diagnostic Setting Record



# Releases of the Conformance Test Tool

Release 1 - Static Tests

Release 2a - Dynamic Testing/non-persistent classes

Release 2b - Extrinsic Method Testing/persistent and non-persistent classes

Release 3 - Behavior Testing

Release 4 - Class and Property Consistency Testing

Release 5 - Extrinsic Method Consistency Testing

Release 6 - Optional Implementation Testing

Release 7 - Linux and Windows testing

- Porting the conformance test tool to Linux
- Testing the conformance test tool on both windows and Linux platforms

Thank You

Questions ?