

Introduction to WfM Component Instrumentation

Intel Corporation

September 29, 1997



*Third-party brands and names are the property of their respective owners

Agenda

- **What's Instrumentation**
- **WfM Baseline Instrumentation Requirements**
- **Delivering Instrumentation**
- **WfM Instrumentation Information Sources**

**Demystifying WfM
component instrumentation**



*Third-party brands and names are the property of their respective owners

Definitions

- **Desktop Management Interface (DMI)**
- **Service Provider (SP)**
- **Management Information Format (MIF)**
- **Management Interface (MI)**
- **Component Interface/Instrumentation (CI)**
- **Management Application (MA)**

What's Instrumentation

Instrumentation ?



- A WfM Baseline component technology to help reduce TCO
- Gives access/control to PC system/components
- Enables PCs to report unusual conditions autonomously

**Instrumentation makes
PC manageable**

intel®

*Third-party brands and names are the property of their respective owners

WfM Baseline Instrumentation Requirements

- Instrumentation requirements specified in terms of DMI 2.00
- Platforms must host DMI SP 2.00 and utilize MI and CI
- Platforms must instrument DMI standard groups/attributes

Standard instrumentation makes Baseline-managed PCs remotely manageable

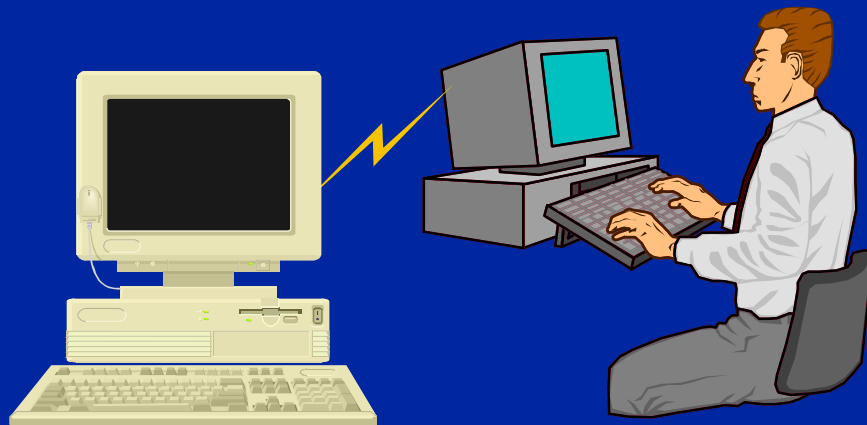


Delivering Instrumentation

Three Steps

Service Provider

Instrumentation



- **DMI Service Provider 2.0**
- **WfM Baseline-compliant instrumentation**
- **Interoperability validation**

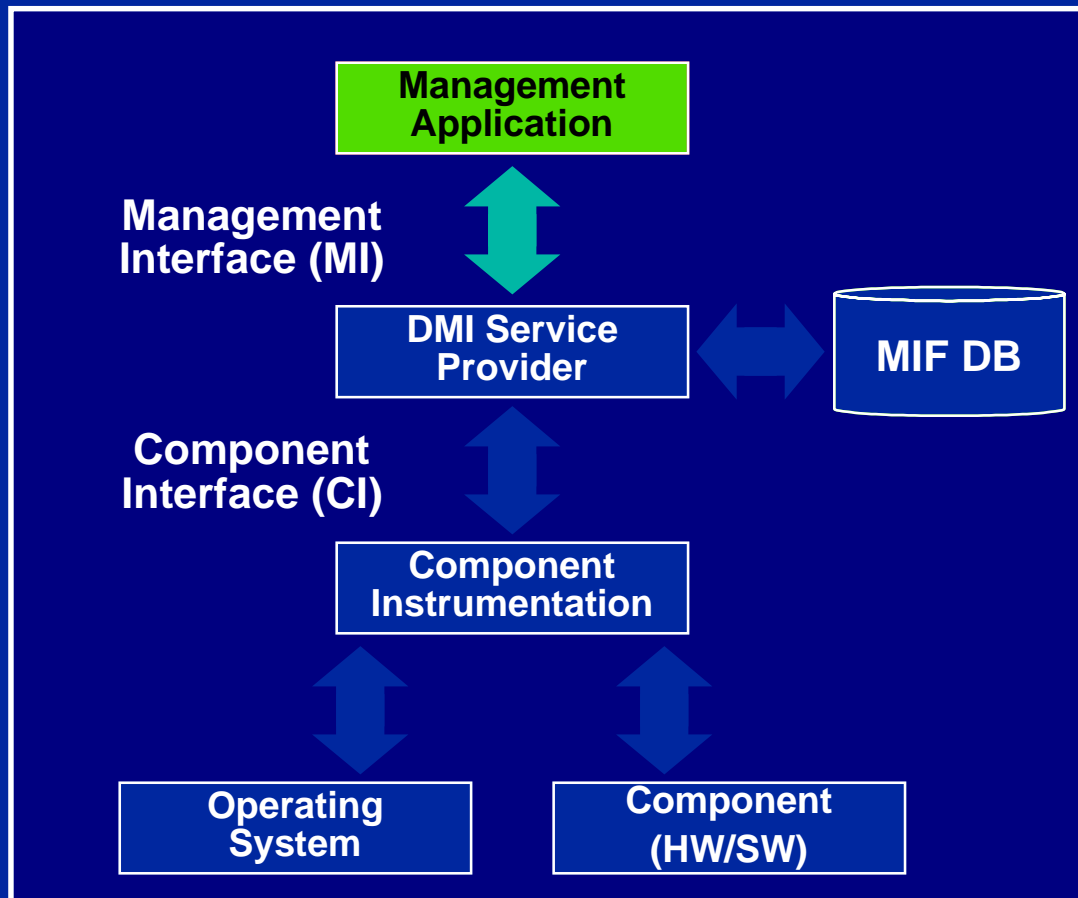
intel[®]

*Third-party brands and names are the property of their respective owners

Delivering Instrumentation

DMI 2.0 Architecture

DMI Service Provider

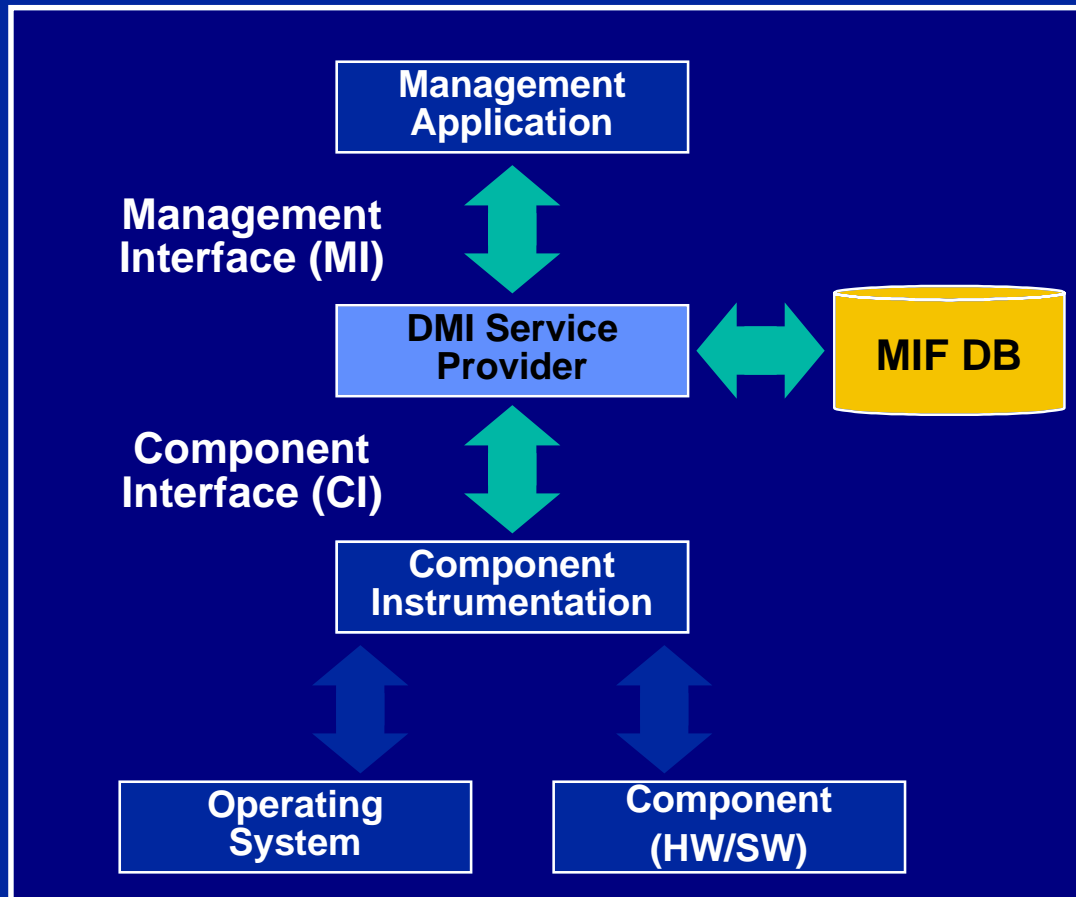


- Mas control/ manage elements of desktops or servers
- Applications communicate with SP through MI

Delivering Instrumentation

DMI 2.0 Architecture

DMI Service Provider

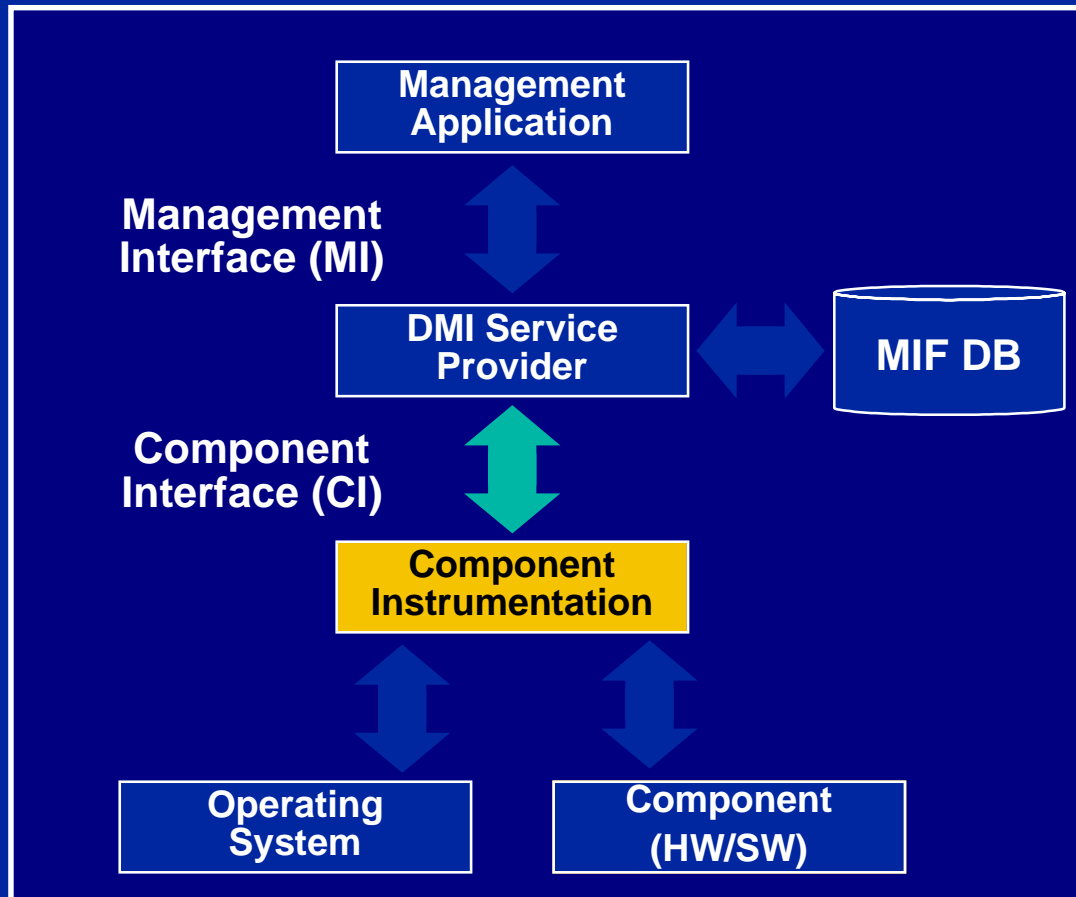


- Three interfaces: CI, MI and interface to the MIF database
- MIF database contains descriptions of managed products
- Each component is described by its MIF

Delivering Instrumentation

DMI 2.0 Architecture

DMI Service Provider



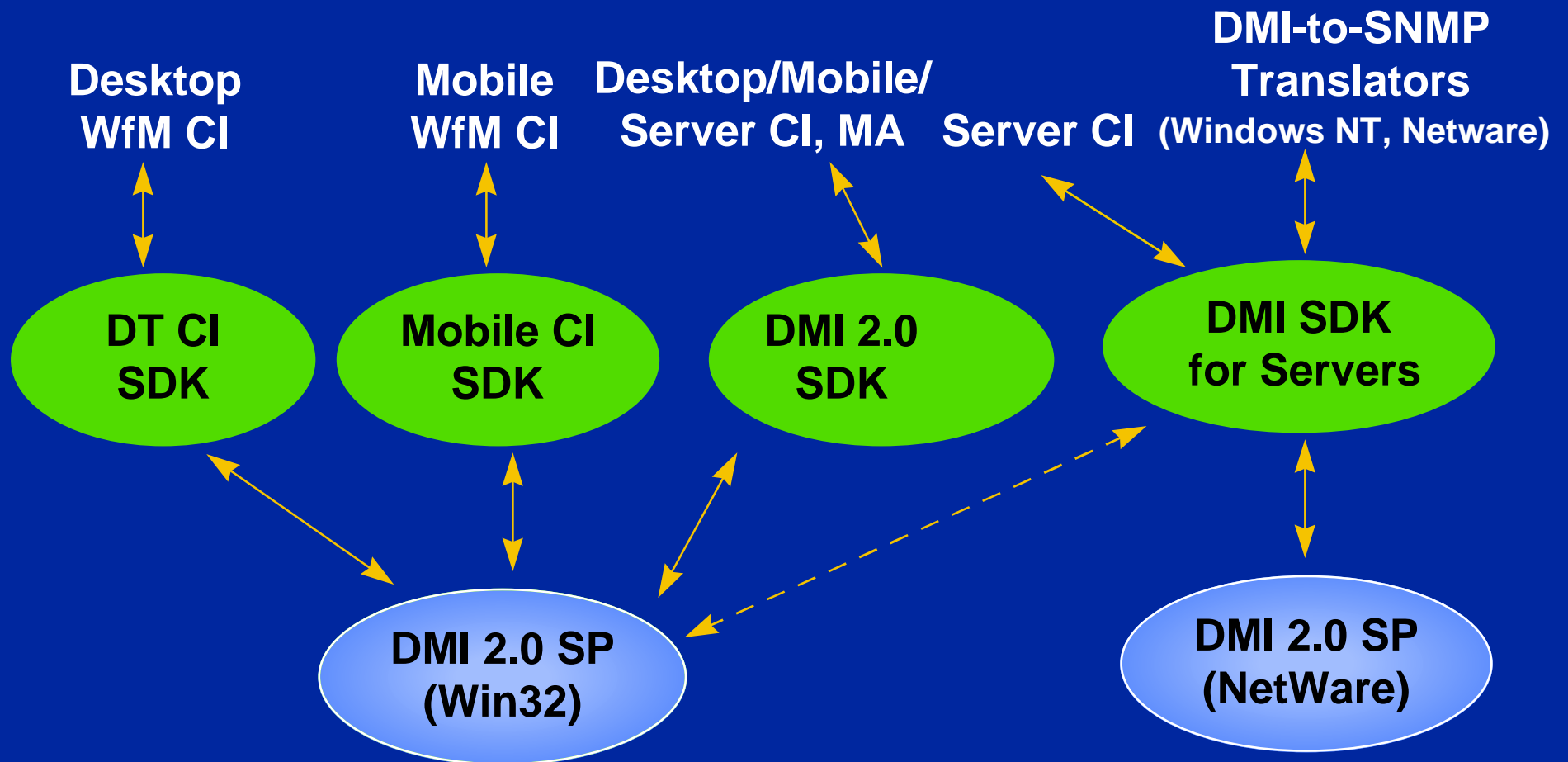
- CI consists of 7 main entry points
- Instrumentation communicates with SP through CI
- Components (HW/SW) are modeled with attributes



DMI provides powerful manageability

Delivering Instrumentation Development Support Tools

Instrumentation



*Third-party brands and names are the property of their respective owners

Delivering Instrumentation

CI Development Tools

Instrumentation



- Includes DMI SP 2.0 in binary form
- Allows CI development for desktops and laptops
- Interoperability validation tools for CI testing (DCTS2, CompCheck)

intel®

*Third-party brands and names are the property of their respective owners

Delivering Instrumentation

CI Development Tools

Instrumentation



- Setup utility for installing and redistributing DMI SP
- Comes with DMI MA and CI examples
- Is free

Several CI development tools are available

intel®

*Third-party brands and names are the property of their respective owners

Delivering Instrumentation

Working With DMI 2.0 SP SDK

Instrumentation



- **Runs on Windows 95, and Windows NT**
- **Requires familiarity with MIF files**
- **Must know DMI Specification 2.00**
- **Requires C/C++ experience**

Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owners' benefit, without intent to infringe.

intel®

*Third-party brands and names are the property of their respective owners

Delivering Instrumentation

Working With DMI 2.0 SP SDK

Instrumentation



- SDK header files and libraries:
 - ◆ Error codes in `\include\dm2err.h`
 - ◆ DMI 2.0 API in `\include\dm2com.h`
 - ◆ Memory handling API in `\include\dm2mem.h`
 - ◆ Indication handling API in `\include\clidmi.h`

**Find API info quickly
using header files**

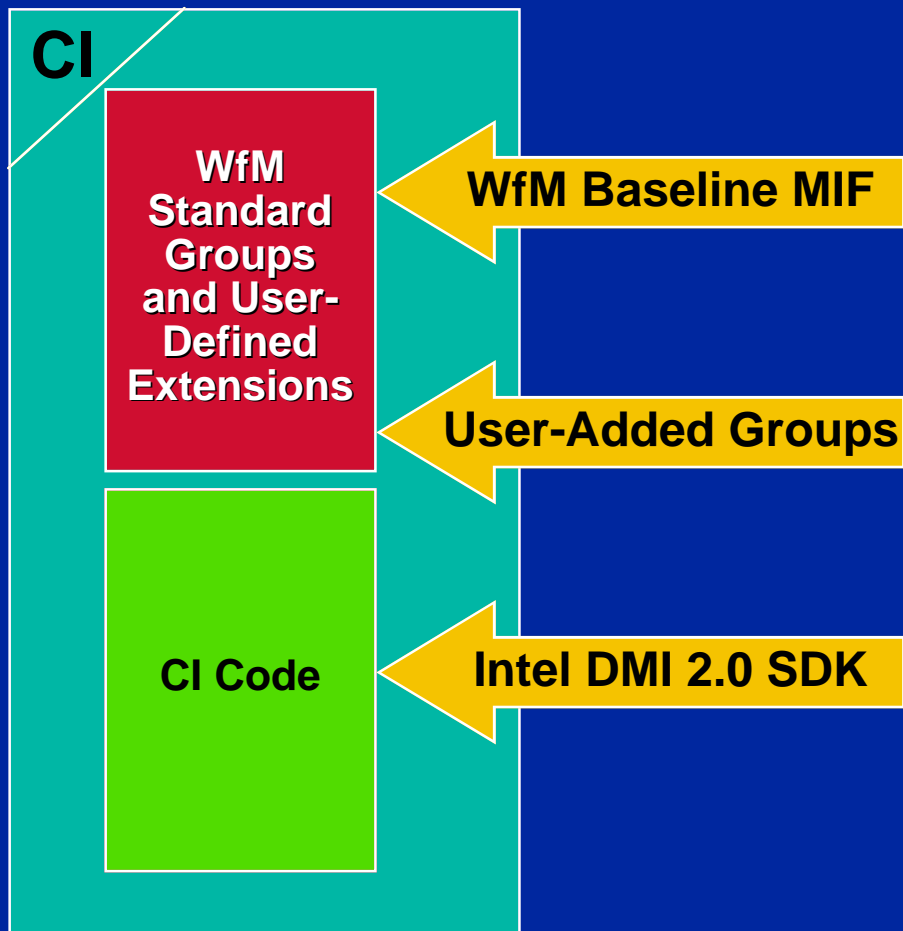
intel®

*Third-party brands and names are the property of their respective owners

Delivering Instrumentation

Instrumentation Guidelines

Instrumentation



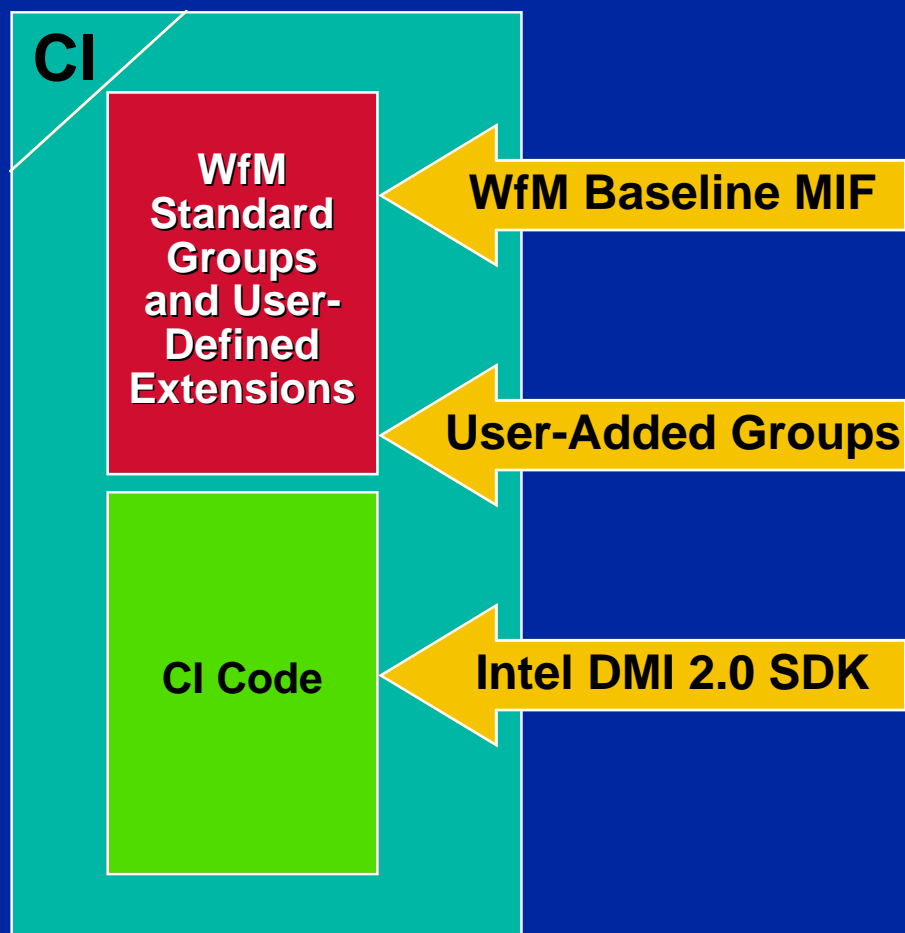
- Start with WfM standard groups
- Add product-specific groups
- Use CompCheck to validate your MIF
- Choose instrumentation model (.EXE, .DLL, device driver)



Delivering Instrumentation

Instrumentation Guidelines

Instrumentation



- Use SDK examples as templates
- Implement the 7 instrumentation entry points
- Install the component and register with DMI SP
- Use DCTS and Compcheck for testing

Delivering Instrumentation

Start with WfM standard groups

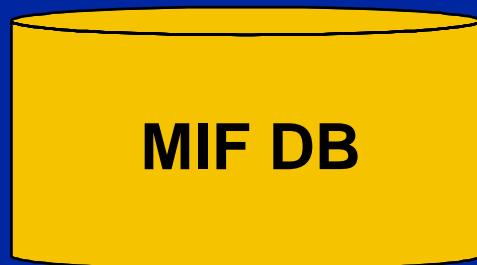
Instrumentation

Start Path

Name = "DUMMY
INSTRUMENTATION"

Win32 = Direct-
Interface

End Path



- Select a symbolic name for your component
- **Win32 =** Means Windows 95 and Windows NT platforms

Delivering Instrumentation

Extending Standard Groups

Instrumentation

Start Attribute

Name =

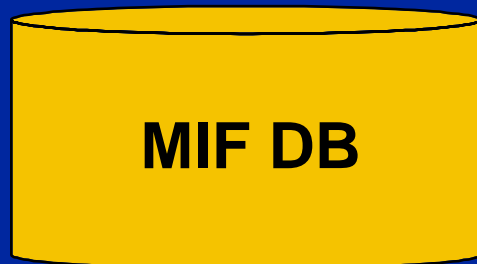
"Attribute_Name"

ID = 1

Value = * "DUMMY
INSTRUMENTATION"

End Attribute

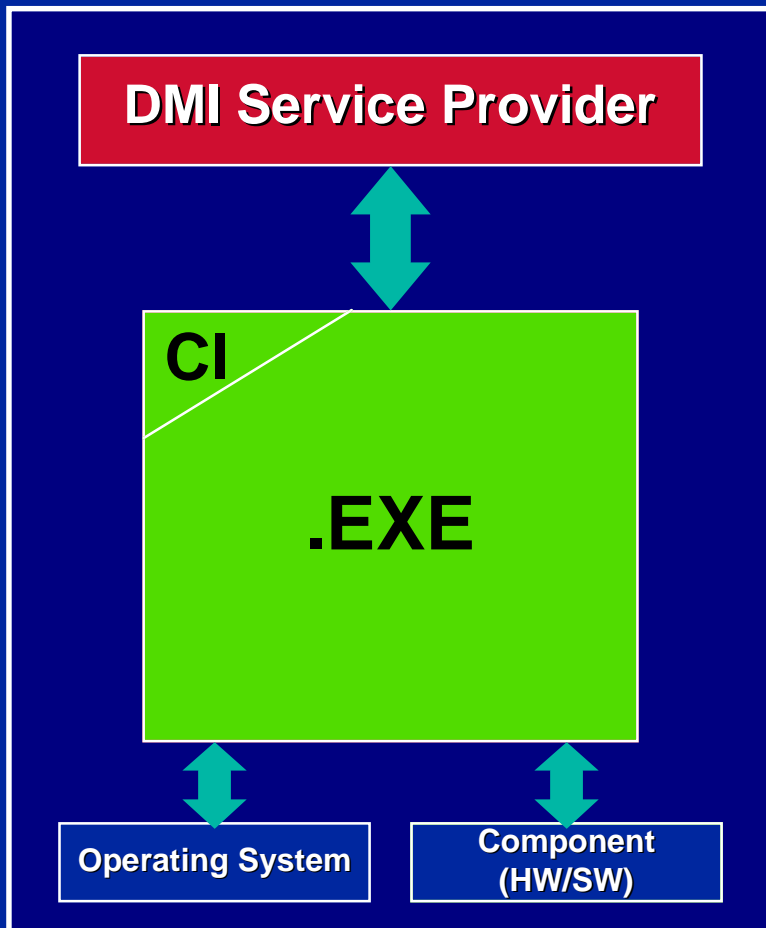
- Add product-specific groups and attributes
- "*" means the attribute is instrumented



Delivering Instrumentation

Choose Instrumentation Model

Instrumentation

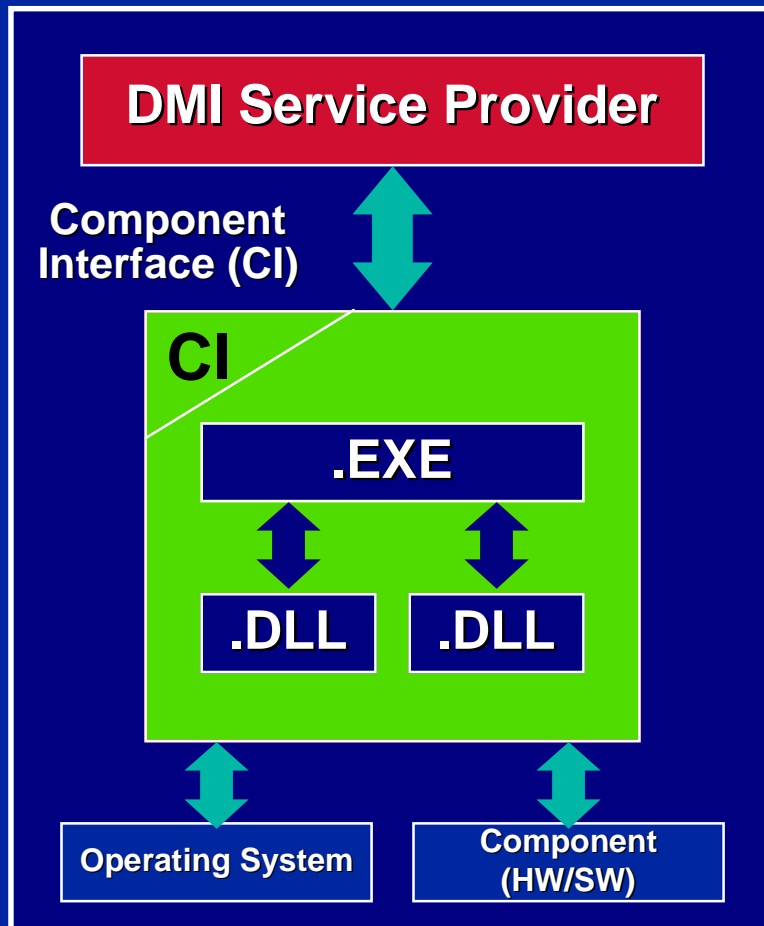


- A stand-alone .EXE module that can run as a service

Delivering Instrumentation

Choose Instrumentation Model

Instrumentation

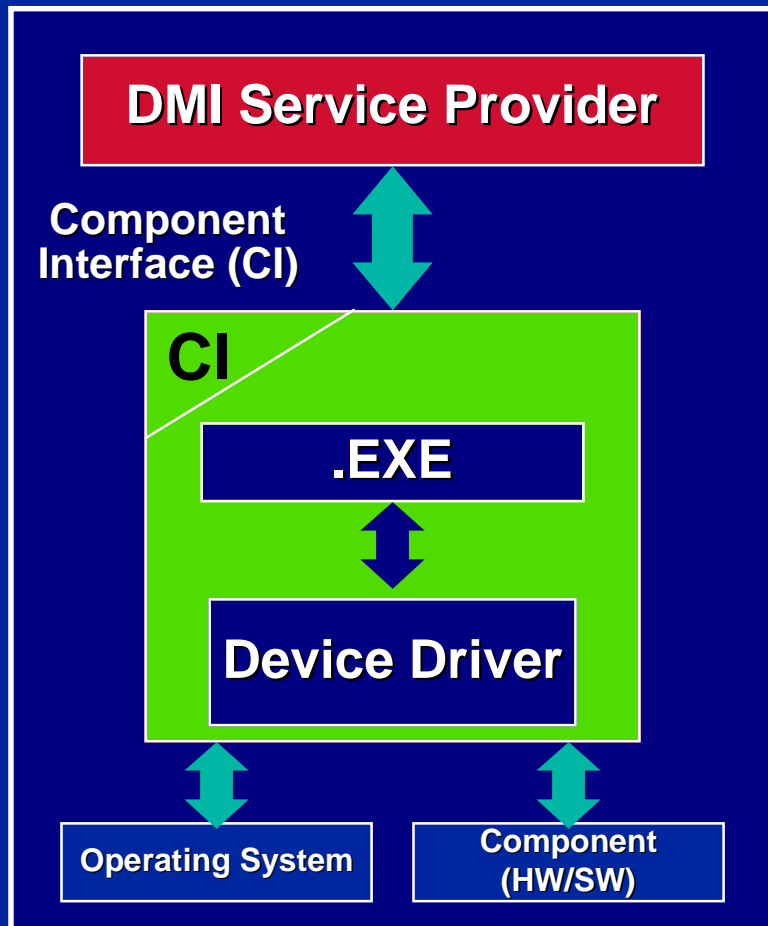


- A combination of .EXE and .DLLs to reduce memory footprint

Delivering Instrumentation

Choose Instrumentation Model

Instrumentation



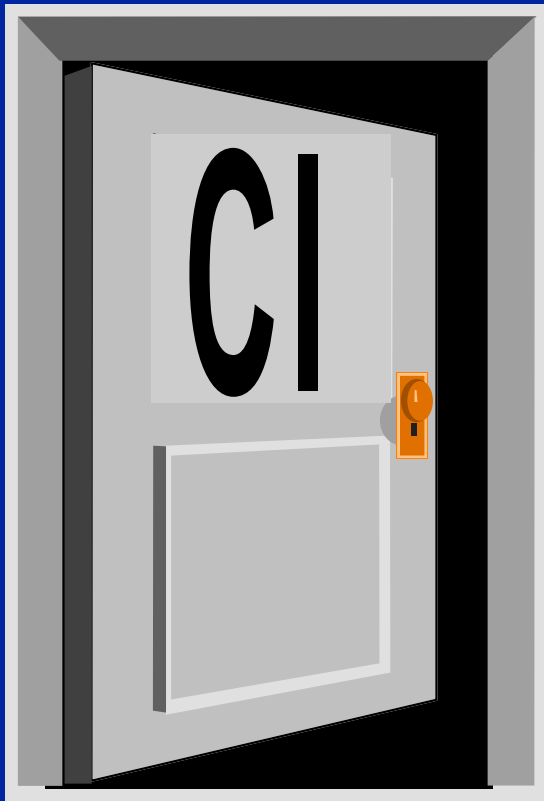
- A combination of .EXE and device driver for getting ring zero data

Instrumentation can be implemented as EXEs, DLLs or device drivers

Delivering Instrumentation

Implement CI Entry Points

Instrumentation



- CiGetAttribute()
- CiGetNextAttribute()
- CiSetAttribute()
- CiReserveAttribute()
- CiReleaseAttribute()
- CiAddRow()
- CiDeleteRow()

You must implement all seven!

intel®

*Third-party brands and names are the property of their respective owners

Delivering Instrumentation Implementation Checklist

Instrumentation



- Create a new node address



- Set node address and register with SP

- Check for installed components



- Find component ID using “exactMatch”

- Install the component MIF

- Initialize the CI entry points



- Fill group and attribute info

- Register component as Direct-Interface

Delivering Instrumentation

Test the instrumentation



- Run CompCheck for DMI compliance checking
- Use DCTS2 to exercise/test your instrumentation
- Test your CI code locally and remotely

Test for interoperability and compliance with WfM baseline.



Stage Demo

- **Step through the Multi-Timer instrumentation code example**
- **Run Multi-Timer instrumentation on the managed node**
- **Use DCTS2 on the managing node to subscribe for, and receive events from the managed node**

WfM Instrumentation Information Sources



- **Intel DMI 2.0 SP SDK Updates**
 - ◆ www.intel.com/managedpc
- **Technical Support**
 - ◆ Newsgroups and FAQs in developer area of www.intel.com/managedpc
 - ◆ developer_support@intel.com
 - ◆ support@dmtof.org
- **DMI 2.0 Specification and Other DMTF Documents**
 - ◆ WfM CD, www.dmtf.org

Summary

- **DMI provides powerful manageability**
- **Three WfM development tools are available**
- **Save time by using the SDK examples**
- **Instrumentation can be implemented as EXEs, DLLs or device drivers**
- **Instrumentation must implement the seven CI entry points**

Call to Action



- **Develop WfM compliant instrumentation for your products**

intel®

*Third-party brands and names are the property of their respective owners