CIMOM Configuration Infrastructure Proposal

Hewlett-Packard Company

Contact E-mail: roger_kumpf@hp.com

CIMOM Configuration Infrastructure Proposal

Configuration Specification

Three-tier configuration at start up

1. Configuration properties with the default values stored in memory. These default values are used in case the configuration properties are not defined anywhere else, or if the value for a property specified elsewhere is not valid.
2. Configuration properties with the values stored in configuration files. The CIMOM at startup loads the configuration values from current configuration file. The values specified in the pending configuration file overrides the values specified in the current configuration file and they in turn will override the in-memory default values.
3. Configuration properties with the values specified on the command line. The values specified on the CIMOM startup command line override the values specified in the configuration files and the in-memory default values.

Configuration changes at run time

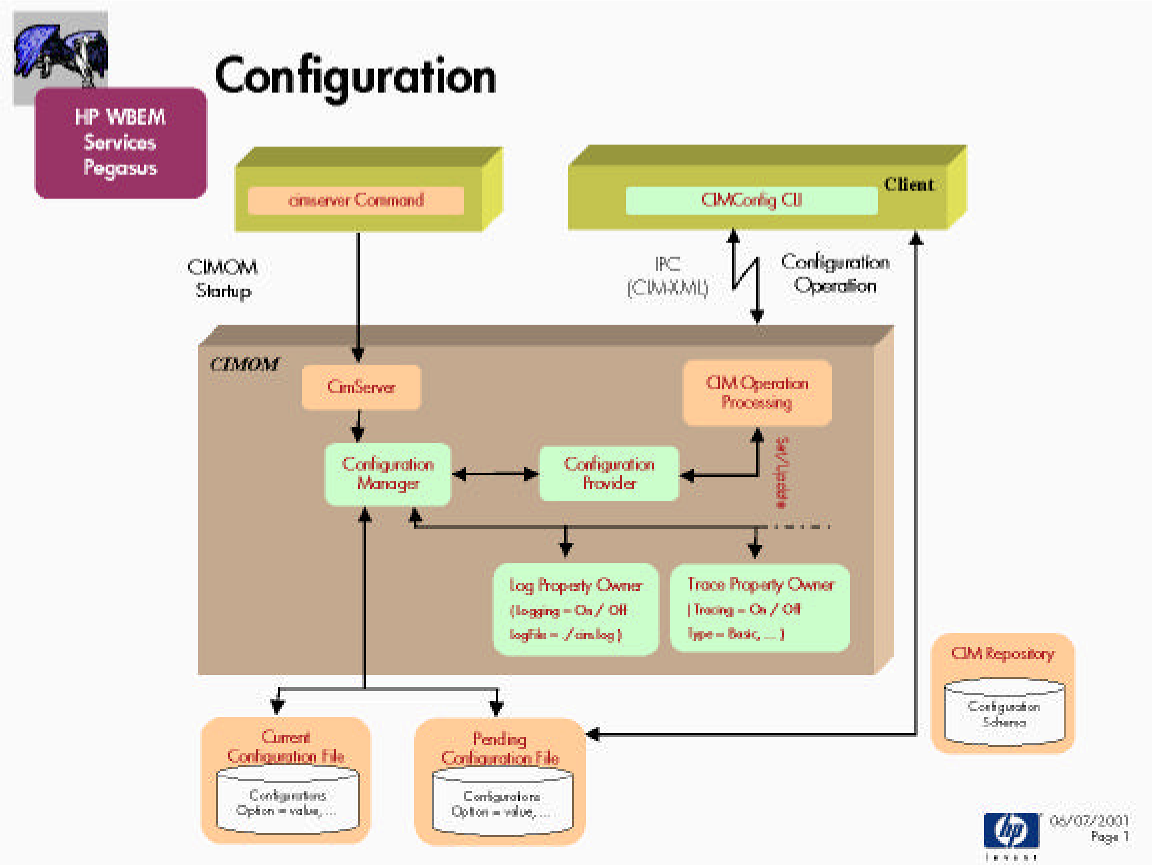
There are two types of configuration changes, immediate and delayed. Delayed changes take effect only at CIMOM restart. Immediate changes take effect right away, but are only permitted for configuration properties that are dynamically enabled.

Configuration Files

There will be two configuration files, current configuration file and pending configuration file. These files will be text files containing *key and value* pairs for each configuration property. Both files will be created before or during CIMOM installation and they will contain predefined configuration properties and values. The configuration of the currently running CIMOM is represented in the current configuration file. The pending configuration file contains the values that will take effect on CIMOM restart.

Configuration Architecture

The CIMOM will be configurable by specifying the configuration properties either in the configuration file or on the command line or both. The diagram shows the CIMOM configuration modules and the interaction between them.

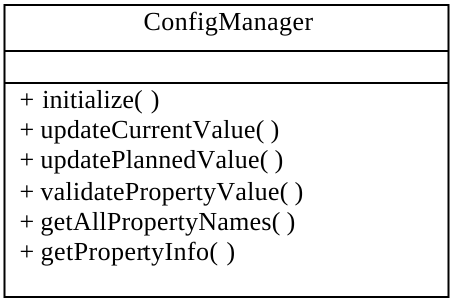


Configuration Property Owner

There will be an owner for a configuration property or for a set of configuration properties. The owner implements a common interface supporting methods to set and update the configuration values. The owner will have full knowledge of the valid configuration values and the default values for the properties it owns. It does the necessary validation of the configuration values that are passed before accepting them. The values for configuration properties are stored in the data members of the owners and they are accessible through their public static member functions.

Configuration Manager

The configuration manager is a module that is responsible for the management of configuration properties within the CIMOM. The ConfigManager class with the public methods is shown below.



ConfigManager

+ initialize( )
+ updateCurrentValue( )
+ updatePlannedValue( )
+ validatePropertyValue( )
+ getAllPropertyNames( )
+ getPropertyInfo( )

The configuration manager module provides the following functionality:

CIMOM Startup

The initialize( ) method will do the following:

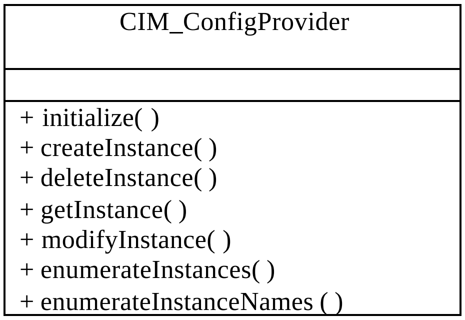
- Replace the content of the current configuration file with the content from the pending configuration file.
- Read configuration properties and values from the current configuration file.
- Create configuration property owner object for each configuration property and call its set method by passing the configuration value.
- Read configuration properties and values from the command line. Validate them by calling the isValid() method of the owner object. Then merge the valid properties over values read from the current configuration file.
- Write the new configuration values in to the current configuration file.

Configuration Provider Request

- validatePropertyValue( ) method will validate the configuration value by calling isValid( ) method of the owner object and return the result.
- updateCurrentValue( ) method will update the current configuration value by calling update( ) method of the property owner object. Then updates the configuration value in the current configuration file.
- updatePlannedValue( ) method will update the configuration value in the planned configuration file.
- getAllPropertyNames( ) method will return a list of all the property names.
- getPropertyInfo( ) method will return detailed information about specified property.

Configuration Provider

There will be a configuration instance provider that implements CIM methods. The provider will be an in-process provider to the CIMOM. It will have access to the Configuration Manager object. The provider will call the appropriate methods in the Configuration Manager to serve the CIM requests from the Client CLI. The CIM_ConfigProvider class with the public methods is shown below.



Following is the CIM_ConfigProvider schema.

```
// ********************************************************************
//    ConfigProperty
//
// ********************************************************************
//
            [Description (
                        "ConfigProperty is a representation of CIMOM specific "
                        "configuration properties. "), provider("CIM_ConfigProvider")
            ]

class CIM_ConfigProperty: CIM_Setting
{
   [Description ("Property Name")]
    string              PropertyName;

   [Description ("Default Value")]
    string              DefaultValue;

   [Description ("Current Value")]
    string              CurrentValue;

   [Description ("Planned Value")]
    string              PlannedValue;

   [Description ("Dynamic Property")]
    boolean             DynamicProperty;
};
```

The CIM Operations supported by CIM_ConfigProvider are shown in Table 1 below.

Table 1: Supported CIM Operations

CIM Operation	
CreateInstance	Not Allowed
DeleteInstance	Not Allowed
GetInstance	Allowed
ModifyInstance	Allowed (See Table 2)
EnumerateInstances	Allowed
EnumerateInstanceNames	Allowed

The following table shows the rules for setting values of the configuration properties.

Table 2: Rules for setting configuration properties

	Property Name	Default Value	Current Value	Planned Value	Dynamic
set	Error	Error	**Allowed**	**Allowed**	Error
set Current Value = new value	Unchanged	Unchanged	**new value**	**Unchanged**	True
	Unchanged	Unchanged	**Error**	**Unchanged**	False
set Planned Value = new value	Unchanged	Unchanged	**Unchanged**	**new value**	True
	Unchanged	Unchanged	**Unchanged**	**new value**	False
set Current Value = new value & Planned Value = new value	Unchanged	Unchanged	**new value**	**new value**	True
	Unchanged	Unchanged	**Error**	**Error**	False

Configuration Command Line (CLI)

There will be a configuration command line (CLI) that will have the required user interface to manipulate the CIMOM configuration properties. The CLI will provide options to set, get, and list the configuration properties. The CLI will make CIM operation calls using the CIM Client API to talk to the CIMOM about configuration changes. If the CIMOM is not running then the CLI will store the configuration changes in the secondary configuration file.