SPIKE: Nightwatchman Evaluation tool

# Different Elements of the report:

Following lists the different elements (Report functionalities) that need to be captured in the report

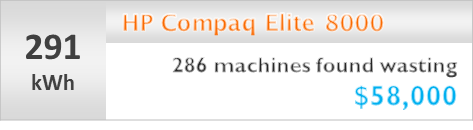
### Report Element 1: Analysis of daily PC power usage for a defined period

**Total possible savings from NightWatchman Enterprise at Massive Dynamic:**

**$7,331,000**

This Exploded graph above analyses the PCS in the organization, the powerdown patterns from a baseline and extrapolate to a graph.

### Report Element 2: Energy Consuming PC Models



So, to show this data we need the PC model information from SCCM database and its power usage

### Report Element 3: Overnight power state by Operating System

## Configuration manager database tables/views to extract:

Initially, I looked at the data extracted by AppClarity and Shopping from our customer backup tables and if anything could be reused by us. But unfortunately, we don’t get any PC power state / usage related information from that. So, based on my own research the following views/tables could be used to get the information that we want to produce the expected report.

1. *v\_R\_System (PC OS information)*
2. *v\_GS\_COMPUTER\_*SYSTEM *(contains PC model information)*
3. *Config Manager 2012:*

### Config Manager 2007

1. v\_PowerComputerDailyActivity (CM 2007 R3)
2. v\_PowerComputerDailyActivityComputerInfo (CM 2007 R3)

### Config Manager 2012:

1. v\_GS\_POWER\_MANAGEMENT\_DAY (TypeOfEvent0) /
2. v\_GS\_POWER\_MANAGEMENT\_MONTH

But to get all the data that we need power management need to be configured in site server.

## Power Management Feature in Config Manager 2007 R3

Steps to configure power management through SCCM

1. Enable the Power Management Client Agent setting on the Site server.
2. On the SCCM client workstation, you’ll find the new Power Management Agent component under the Configuration Manager Properties.
3. Enable Power plans for a collection of machines.
   1. Right-click a collection of machines and select Modify collection settings.
   2. Click on the Power Management tab and select the option to Specify power management settings for the collection.
   3. Define Peak hours and select a Peak plan.
   4. You may select any of predefined power plans or modify the Customized Peak plan.
   5. Similarly for Non-peak plan you may customize the Non-peak power plan and press OK.

About configuring power management in CM 2012

<http://technet.microsoft.com/en-us/library/gg699392.aspx>

## Options and Recommendations:

While we would like to estimate the machine state usage information from the SCCM database, In Configuration Manager 2007 R3 the new Power Management feature was introduced that has all the information that we want. The Power Management feature in Configuration Manager 2012 is fully integrated. In our Analysis tool we may have to look at the database SCCM and extract the info we need from the respective tables or based on different accordingly, So, this could be achieved by different methods as described below:

Method 1: The accurate way: This is derived from the existing PC usage data from the organization and calculates the energy efficiency that could be achieved. So, this is based on analysis of real and past computer data, available from the power management feature.

Method 2**:** In pre-CM 2007 R3, we could retrieve the power details of the machines and determine optimal power efficiency that could be potentially achieved based on the computed power data of the organizations available machines in a collection. The 1E Support maintains the database of the power data information for different make and models. Therefore, This method is by using the power data information of the machines available in the organization and not by the power usage information available through the inbuilt power management feature in Config Manager 2007 R3 and Config Manger 2012.

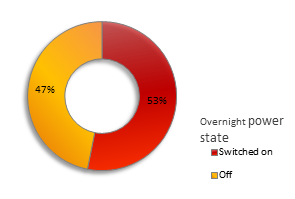
Method 3: In pre-CM 2007 R3, We could look at all of the available state/status timestamp reporting messages of machine data and though not accurate determine a high level estimate.

# Method 3: Algorithm to find active machines during Out of Office Hours

Since there are no full proof methods to find the exact state of machines for every hour, without using power management feature, we may have to develop some heuristics models to determine the machines power state.

## Approach based on heuristics:

1. For the sake of clarity, each of the tables/views  (that has client agent reporting time in it) available is going to be called an **entity**. E.g., Looking at a table/view corresponding to Advertisement Status is an entity, similarly looking at the client deployment state is an entity.
2. Get the date range for which the efficiency reporting, preferably for last **90 days** (Since some of the historical data is deleted after the 90 day window based on a setting)
3. Get the out of office (OOO) hours, e.g., between 20 hours UTC and 7 hours UTC.
4. Look up at each of the entity data for any reported message timestamp between the OOO hours and treat that if any machine is found between the OOO hours for a specific day as 1 unit.
5. Create a stored procedure that would get the list of all the machines and calculate the sum of all those units determined for each of the machine. So, this gives the total number of units where each of the machines not supposed to be in an ON power state within the time period.
6. We could then determine that the rest of the machines in the collection would be in OFF State during the said time period and compute a graph like the one shown below.



Let’s look at each of entities that need to be looked for finding the timestamp status reporting messages. A Synopsis of what each entity does is also presented to understand the entity better.

## Entity 1: Usage of CM Heartbeat messages:

After CM installation, by using one of the CM discovery methods, client and resources could be added to the site. Refresh Configuration Manager client computer discovery data in the site database.

The system discovery table holds data about the discovered information: [System\_DISC]

Since the heartbeat is specific to discovery, and though it doesn’t store all the heartbeat time stamps, its still possible to use this information for our purpose.

**SQLQuery to get the hearbeat information pertaining to a specific machine:**

Select

CS.ResourceID,

CS.Name0 'Machine Name',

AD.AgentTime 'Last Update',

AD.AgentName

From [CMBack].[dbo].[v\_GS\_COMPUTER\_SYSTEM] CS

Join [CMBack].[dbo].[v\_AgentDiscoveries] AD on CS.ResourceID = AD.ResourceId

WHERE AD.ResourceID = 5

Sample output:

|  |  |  |  |
| --- | --- | --- | --- |
| ResourceID | Machine Name | Last Update date time | AgentName |
| 5 | NCHWHC114318 | 1/27/2011 16:26 | MP\_ClientRegistration |
| 5 | NCHWHC114318 | 3/15/2012 12:15 | Heartbeat Discovery |
| 5 | NCHWHC114318 | 1/28/2011 14:24 | SMS\_AD\_SYSTEM\_DISCOVERY\_AGENT |

## Entity 2: Usage of Config Manager State Messages:

Specifies how often in minutes the client agent sends consolidated state messages to the management point. A smaller interval allows more timely state message reporting, but increases the load on the state message system. A larger interval has less impact on performance and might reduce the number of transitory state messages displaying in reports. **The default value is 15 minutes.**

1. State messages are sent by CM agent clients, using a messaging system that is built into the product and that allows clients to send messages to site systems based on important changes of state. State messages are not the same as status messages;
2. State messages provide a snapshot of the state of a process at a specific time.
3. State messages are consolidated on the client and sent according to the **State message reporting cycle (minutes)** value configured in the [Computer Client Agent Properties: General Tab](http://technet.microsoft.com/en-us/library/bb633198.aspx). By default, this cycle is set to occur every 15 minutes.
4. State messages are used with software updates, Network Access Protection (NAP), desired configuration management, client deployment, and client communication

*State messages are sent by Configuration Manager 2007 clients based on important changes of state.* ***State messages are not the same as status messages****; status messages provide information about* ***component behavior and data flow,*** *whereas* ***state messages provide a snapshot of the state of a process at a specific time.***

*In short:*

Status –>provide information about components behavior and data flow

State—> provide snapshot of state of process at specific time.

*Some of the typical State message ID (not the entire list) and description are listed below for more clarity in understanding:*

|  |  |  |
| --- | --- | --- |
| **Topic Type** | **State Message ID** | **State Message Description** |
| 300 | 0 | Compliance state unknown |
| 300 | 1 | Compliant |
| 300 | 2 | Non-compliant |
| 300 | 3 | Conflict detected |
| 301 | 0 | Enforcement state unknown |
| 301 | 1 | Installing update(s) |
| 301 | 2 | Waiting for restart |
| 301 | 3 | Waiting for another installation to complete |
| 301 | 4 | Successfully installed update(s) |
| 301 | 5 | Pending system restart |
| 301 | 6 | Failed to install update(s) |
| 301 | 7 | Downloading update(s) |
| 301 | 8 | Downloaded update(s) |
| 301 | 9 | Failed to download update(s) |
| 301 | 10 | Waiting for maintenance window before installing |
| 302 | 0 | Evaluation state unknown |
| 302 | 1 | Evaluation activated |

*For more info: please refer MSDN here:* <http://technet.microsoft.com/en-us/library/bb932203.aspx>

The state messages are captured in different tables and for each entity the specific task datetime is updated, but unfortunately not captured as it is every time, which makes sense for performance reasons. Therefore, the state messages could be used by extrapolating that something happened at the specific time, though not very accurately.

The different states correspond to entries configured from the state names table, the information of which can be seen at v\_StateNames

Some of the different views captured from various state tables are:

v\_AssignmentState\_Combined,

v\_ClientDeploymentState,

v\_ClientHealthState

e.g,

select ResourceID, StateTime from [CMBack].[dbo].v\_AssignmentState\_Combined

where ResourceID=2

|  |  |
| --- | --- |
| ResourceID | StateTime |
| 2 | 27:50.5 |
| 2 | 27:51.8 |
| 2 | 03:45.0 |

Similarly other state tables could be looked at for the time and machine ID’s could be captured for those specified duration.

## Entity 3: Usage of Config Manager Status Messages:

SMS uses status messages to report information about the health of your site. Status messages are generated from both client and server components.

A status message is similar to a Windows event or a message written to a log file. SMS components generate status messages while they carry out their tasks. The different status messages that could be captured are as follows:

* **Advertisement Status**
* **Package Status**
* **Site Status**
* **Component Status** (under **Site Status**)
* **Site System Status** (under **Site Status**)

The following could be inferred from the status messages:

Message Severity, Message Type and Message Date and Time

***SMS stamps every status message with the system date and time that the message is reported, down to millisecond resolution.***

Status messages provide information about component behavior and data flow.

Query:

Select Distinct

VRS.ResourceID,

SM.TopLevelSiteCode,

SM.MachineName 'Machine Name',

SM.Time 'TIME',

'Advertisement Status' = Case

When ID&0x0000FFFF = 3900 Then '<Distribution Server>'

When ID&0x0000FFFF = 30006 Then '<Console Advertised From>'

When ID&0x0000FFFF = 10000 Then 'Check OFR'

When ID&0x0000FFFF = 10001 Then 'Check PKG'

When ID&0x0000FFFF = 10002 Then 'Received'

When ID&0x0000FFFF = 10003 Then 'CmdLine Not Found'

When ID&0x0000FFFF = 10004 Then 'Invalid Exe Platform'

When ID&0x0000FFFF = 10005 Then 'Started'

When ID&0x0000FFFF = 10006 Then 'Failed'

When ID&0x0000FFFF = 10007 Then 'Failed'

When ID&0x0000FFFF = 10008 Then 'Complete'

When ID&0x0000FFFF = 10009 Then 'Successful'

When ID&0x0000FFFF = 10010 Then 'No Uninstall Found'

When ID&0x0000FFFF = 10011 Then 'Uninstall Exe Not Found'

When ID&0x0000FFFF = 10012 Then 'Uninstall Started'

When ID&0x0000FFFF = 10013 Then 'Uninstall Failed'

When ID&0x0000FFFF = 10014 Then 'Uninstall Failed'

When ID&0x0000FFFF = 10015 Then 'Uninstall Failed'

When ID&0x0000FFFF = 10016 Then 'Uninstall Success'

When ID&0x0000FFFF = 10017 Then 'Removal Key Not Found'

When ID&0x0000FFFF = 10018 Then 'Rejected By OS'

When ID&0x0000FFFF = 10019 Then 'Advertisement Expired'

When ID&0x0000FFFF = 10020 Then 'Slow Network'

When ID&0x0000FFFF = 10021 Then 'Machine Was Restarted'

Else '<No Status !>'

End

From [CMBack].[dbo].[StatusMessages] SM

Join [CMBack].[dbo].[v\_R\_System] VRS on SM.MachineName = VRS.Name0

Join [CMBack].[dbo].[StatusMessageInsStrs] SMI on SM.RecordID = SMI.RecordID

Join [CMBack].[dbo].StatusMessageAttributes SMA on SM.RecordID = SMA.RecordID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ResourceID | TopLevelSiteCode | Machine Name | TIME | Advertisement Status |
| 2 | NCH | NAPLFLNCHSCCM01 | 35:39.3 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 58:17.0 | CmdLine Not Found |
| 2 | NCH | NAPLFLNCHSCCM01 | 33:27.0 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 50:08.1 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 08:38.3 | Advertisement Expired |
| 2 | NCH | NAPLFLNCHSCCM01 | 49:31.0 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 30:24.3 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 09:59.9 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 37:25.3 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 42:39.3 | <No Status !> |
| 2 | NCH | NAPLFLNCHSCCM01 | 22:07.5 | <No Status !> |

**P.S. Most of the status messages are generated from the Site machine.**

## Entity 4: Usage of Operating System info from Resource Explorer :

The views corresponding to the OS infor are [v\_HS\_OPERATING\_SYSTEM] (for historical) and [v\_GS\_OPERATING\_SYSTEM] (for current), but unfortunately the LastBootUp Time() doesn’t seem to be always accurate. It is correct sometimes and it looks like not correct some other times. I found this out by comparing it with the power management data  (available from CM 2007 R3 )from v\_PowerComputerDailyActivity and [v\_PowerComputerDailyActivityComputerInfo]. It doesn’t correlate between them. I also verified the same from another CM 2007 environment in my  VM ‘s. Please see the attached report taken taken for a few days from customer database. Though the data is mostly correct, the entry for 3/11/2012 14:24 doesn’t correlate with those from the power activity data.

So this data could be utilized and check if there are boot up times or timestamps reported within the OOO hours.

### Query to get the last boot up user and time:

Select

SD.Name0 'Machine Name',

SD.User\_Name0 'Last Logged on User Name',

OS.LastBootUpTime0 'Last Boot Date and time'

From [CMBack].[dbo].v\_R\_System SD

Join [CMBack].[dbo].v\_GS\_OPERATING\_SYSTEM OS on SD.ResourceID = OS.ResourceID

Order By 'Machine Name'

Sample output:

|  |  |  |
| --- | --- | --- |
| Machine Name | Last Logged on User Name | Last Boot Date and time |
| 01E44RAD | nch\_rad | 3/11/12 10:49 PM |
| 01F09RAD | mbeasley | 2/21/12 5:32 AM |
| 01G08RAD | nch\_rad | 12/14/11 2:33 PM |
| 01G09RAD | nch\_rad | 3/13/12 10:51 PM |
| 01G33RAD | nch\_rad | 3/1/12 4:08 PM |
| 03A11ER | nch\_er | 3/13/12 3:15 PM |
| 03A13ER | nch\_er | 3/8/12 3:45 AM |

In this entity, We could infact look at both the boot up time as well as the timestamp on when it was sent, so both would be helpful.

SELECT [ResourceID],[LastBootUpTime0],[TimeStamp] ,[Caption0],[CountryCode0],[InstallDate0],[Organization0]

FROM [CMBack].[dbo].[v\_GS\_OPERATING\_SYSTEM]

As well as data from [v\_HS\_Operating\_System] for historical information

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rs ID | LastBootUpTime0 | TimeStamp | Caption0 | CountryCode0 | InstallDate0 | Organization0 |
| 2 | 12/20/2011 12:20 | 12/20/2011 17:03 | Microsoft Windows Server 2008 R2 Standard | 1 | 02:00.0 | NULL |
| 5 | 3/16/2012 8:39 | 3/16/2012 9:20 | Microsoft Windows XP Professional | 1 | 09:00.0 | NCH Healthcare System |
| 6 | 2/21/2012 6:25 | 2/22/2012 10:36 | Microsoft Windows XP Professional | 1 | 18:00.0 | NCH Healthcare System |
| 8 | 3/9/2012 15:55 | 3/11/2012 10:33 | Microsoft(R) Windows(R) Server 2003, Standard Edition | 1 | 52:00.0 | NCH HEALTHCARE |
| 10 | 2/21/2012 15:01 | 2/22/2012 12:11 | Microsoft Windows XP Professional | 1 | 24:00.0 | NCH Healthcare System |
| 12 | 2/25/2012 4:31 | 2/25/2012 11:18 | Microsoft Windows XP Professional | 1 | 15:00.0 | NCH HealthCare System |

## Entity 5: Usage of Hardware Inventory Reporting:

The hardware inventory views contain information about the computer hardware scanned on Configuration Manager 2007 client computers. Many hardware inventory views are created in Configuration Manager by default, and many more can be created depending on which classes and properties are enabled in the ***SMS\_DEF.MOF*** file, what version of Configuration Manager is installed, which Configuration Manager components are configured, and so forth. Because of this, it is likely that Configuration Manager sites collect different hardware inventory resulting in different hardware inventory views.

### Query should be based on the schema table

The hardware inventory schema is important to understand when creating queries for Configuration Manager reports. Most of the client data within Configuration Manager is contained in one of the two hardware inventory schema views: v\_GroupMap and v\_GroupAttributeMap. The v\_GroupMap view contains a list of all the hardware inventory groups and the associated view for each of the groups. The v\_GroupAttributeMap contains all of the attributes that are inventoried for each of the groups. Both views can be joined together by using the GroupID column and joined to the v\_ResourceMap discovery schema view by using the ResourceType column.

Because hardware inventory can be configured through enabling or disabling classes in the SMS\_DEF.MOF, one Configuration Manager site's SQL database might have different hardware inventory views and schema when compared to another site. The following query joins the v\_GroupMap and v\_GroupAttributeMap to generate the hardware inventory view schema, based on the specific SMS\_DEF.MOF file settings for the site:

SELECT DISTINCT GM.DisplayName, GM.InvClassName,

GM.InvHistoryClassName, GAM.AttributeName,

GAM.ColumnName, GM.MIFClass

FROM [CMBack].[dbo].v\_GroupMap GM

INNER JOIN [CMBack].[dbo].v\_GroupAttributeMap GAM

ON GM.GroupID = GAM.GroupID

View to be looked at:

[Computer\_System\_HIST], [Desktop\_Monitor\_HIST]

More information on that could be found in msdn: <http://technet.microsoft.com/en-us/library/dd334659.aspx>

## Entity 6: Usage of Software Inventory Reporting:

The Configuration Manager 2007 software inventory views contain information about the files and products that are found on CM clients during software inventory scanning. Software inventory, by default, will scan for all executable file types (\*.exe) on clients.

v\_GS\_LastSoftwareScan: This view lists the last time each CM client was scanned for software inventory.

# Helpful SQL Scripts:

## Query to get machine information from a specified collection:

Select

VRS.ResourceID 'Machine ID',

VRS.Name0 'Machine Name',

VRS.Resource\_Domain\_OR\_Workgr0 'Resource Domain',

VRS.User\_Name0 'Login ID',

VRS.User\_Domain0 'Account Domain',

PCB.SerialNumber0 'Serial Number',

CS.Manufacturer0 Manufacturer,

CS.Model0 Model,

SAS.SMS\_Assigned\_Sites0 'Assigned Site Code'

From [CMBack].[dbo].[v\_R\_System] VRS

Join [CMBack].[dbo].[v\_FullCollectionMembership] FCM on VRS.ResourceID = FCM.ResourceID

Join [CMBack].[dbo].[v\_Collection] COL on FCM.CollectionID = COL.CollectionID

Join [CMBack].[dbo].[v\_GS\_PC\_BIOS] PCB on VRS.ResourceID = PCB.ResourceID

Join [CMBack].[dbo].[v\_GS\_COMPUTER\_SYSTEM] CS on VRS.ResourceID = CS.ResourceID

Join [CMBack].[dbo].[v\_RA\_System\_SMSAssignedSites] SAS on VRS.ResourceID = SAS.ResourceID

Where COL.Name = 'All Systems'

**Sample Output:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Machine ID** | **Machine Name** | **Resource Domain** | **Login ID** | **Account Domain** | **Serial Number** | **Manufacturer** | **Model** | **Assigned Site Code** |
| **2** | **NAPLFLNCHSCCM01** | **NCHHCS** | **NULL** | **NULL** | **VMware-42 10 0d 45 1c e9 15 db-5b be 0e 65 3a 36 b1 89** | **VMware, Inc.** | **VMware Virtual Platform** | **NCH** |
| **5** | **NCHWHC114318** | **NCHHCS** | **rprieto** | **NCHHCS** | **89SM7M1** | **Dell Inc.** | **Precision M2400** | **NCH** |
| **6** | **12D23OR** | **NCHHCS** | **nch\_or** | **NCHHCS** | **H8G7PN1** | **Dell Inc.** | **OptiPlex 780** | **NCH** |
| **8** | **NCH-EMERGIN** | **NCHHCS** | **test99** | **NCHHCS** | **USE519A396** | **HP** | **ProLiant DL380 G4** | **NCH** |
| **10** | **51D415N** | **NCHHCS** | **nch\_5n** | **NCHHCS** | **5GPSPL1** | **Dell Inc.** | **OptiPlex 780** | **NCH** |
| **12** | **26D405SW** | **NCHHCS** | **nch\_3n** | **NCHHCS** | **3140936-0001** | **micronpc.com** | **ClientPro CR** | **NCH** |
| **13** | **NCH5SE553** | **NCHHCS** | **nco\_bha** | **NCHHCS** | **9LTYA92356** | **Matsushita Electric Industrial Co.,Ltd.** | **CF-52GGNBE2M** | **NCH** |
| **14** | **32C19OR** | **NCHHCS** | **nch\_or** | **NCHHCS** | **H8C3PN1** | **Dell Inc.** | **OptiPlex 780** | **NCH** |
| **15** | **NN2133** | **NCHHCS** | **nco\_pcu** | **NCHHCS** | **J8JX8C1** | **Dell Inc.** | **Latitude D520** | **NCH** |

## Query to get Computer Activity Information for a range of machines

SELECT DISTINCT db1.MachineID,db1.[Date]

, db1.Hour HourOfTheDay

, db3.MinutesActive ComputerActiveMinutes

, db4.MinutesActive ComputerONMinutes

, db5.MinutesActive MonitorONMinutes

, db2.ComputerState

FROM [CMBack].[dbo].[v\_PowerComputerDailyActivity] db1

INNER JOIN [CMBack].[dbo].[v\_PowerComputerDailyActivity] db3

ON db1.MachineID = db3.MachineID

AND db1.Hour = db3.Hour

AND db1.Date = db3.Date

AND db3.TypeOfEvent LIKE 'ComputerActive'

INNER JOIN [CMBack].[dbo].[v\_PowerComputerDailyActivity] db4

ON db1.MachineID = db4.MachineID

AND db1.Hour = db4.Hour

AND db1.Date = db4.Date

AND db4.TypeOfEvent LIKE 'ComputerOn'

INNER JOIN [CMBack].[dbo].[v\_PowerComputerDailyActivity] db5

ON db1.MachineID = db5.MachineID

AND db1.Hour = db5.Hour

AND db1.Date = db5.Date

AND db5.TypeOfEvent LIKE 'MonitorOn'

JOIN [CMBack].[dbo].[v\_PowerComputerDailyActivityComputerInfo]AS db2

ON db1.MachineID = db2.MachineID

AND db1.Date = db2.Date

AND db1.Hour = db2.Hour

WHERE db1.[MachineID] BETWEEN 2 AND 5

order by db1.MachineID, db1.[Date] desc, db1.[Hour]

Sample output:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Machine ID | Date time | Hour Of The Day | Computer Active Minutes | Computer ON Minutes | Monitor ON Minutes | Computer State |
| 5 | 15-Feb-12 | 0 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 1 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 2 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 3 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 4 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 5 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 6 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 7 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 8 | 23 | 48 | 48 | Active |
| 5 | 15-Feb-12 | 9 | 45 | 60 | 60 | Active |
| 5 | 15-Feb-12 | 10 | 0 | 60 | 0 | Active |
| 5 | 15-Feb-12 | 11 | 0 | 60 | 0 | Active |
| 5 | 15-Feb-12 | 12 | 5 | 60 | 5 | Active |
| 5 | 15-Feb-12 | 13 | 60 | 60 | 60 | Active |
| 5 | 15-Feb-12 | 14 | 30 | 60 | 55 | Active |
| 5 | 15-Feb-12 | 15 | 7 | 60 | 17 | Active |
| 5 | 15-Feb-12 | 16 | 0 | 60 | 0 | Active |
| 5 | 15-Feb-12 | 17 | 0 | 25 | 0 | Active |
| 5 | 15-Feb-12 | 18 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 19 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 20 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 21 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 22 | 0 | 0 | 0 | NotActive |
| 5 | 15-Feb-12 | 23 | 0 | 0 | 0 | NotActive |

## Query to get all the site codes:

Select all SMS\_Site.Sitecode from Sites AS SMS\_Site

Sitecode

--------

NCH

(1 row(s) affected)

# Comparison of our AFR hardware inventory and Config Manager hardware inventory

The task is to identify whether we could utilize the model information from SCCM database to compute the power consumption. When I executed a simple query across the AFR database and the customer database, we could easily compute the power consumption based on the model information. But there are models for which we don’t get the power values, in those cases probably we need to have a default power ratings.

SELECT DISTINCT HCON.[HardwareId]

,[Type]

,[Manufacturer]

,[Model]

,HCON.State

,HCON.Watts

FROM [AgilityFrameworkReporting].[dbo].[tbAFR\_Lookup\_HardwareComponent] HCOM

Join [AgilityFrameworkReporting].[dbo].[tbNWM\_Lookup\_HardwareConsumption] HCON

ON HCOM.HardwareId=HCON.HardwareId

JOIN [CMBack].[dbo].Computer\_System\_Data CS

ON HCOM.Model= CS.Model0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HardwareId | Type | Manufacturer | Model | State | Watts |
| 6 | Chassis | Dell Inc. | OptiPlex GX280 | 2 | 52.1 |
| 22 | Chassis | Dell Inc. | OptiPlex 780 | 1 | 52.1 |
| 11 | Chassis | Dell Inc. | OptiPlex 755 | 5 | 2 |
| 18 | Chassis | Dell Inc. | Latitude E6510 | 2 | 20 |
| 17 | Chassis | Dell Inc. | Latitude E6420 | 1 | 30 |
| 10 | Chassis | Dell Inc. | Latitude E6410 | 4 | 1 |
| 6 | Chassis | Dell Inc. | OptiPlex GX280 | 5 | 2 |
| 22 | Chassis | Dell Inc. | OptiPlex 780 | 4 | 2 |
| 21 | Chassis | Dell Inc. | OptiPlex 780 | 0 | 2 |
| 11 | Chassis | Dell Inc. | OptiPlex 755 | 2 | 52.1 |
| 18 | Chassis | Dell Inc. | Latitude E6510 | 5 | 1 |
| 9 | Chassis | Dell Inc. | Latitude E6400 | 0 | 1 |
| 21 | Chassis | Dell Inc. | OptiPlex 780 | 3 | 43.1 |
| 17 | Chassis | Dell Inc. | Latitude E6420 | 4 | 1 |
| 10 | Chassis | Dell Inc. | Latitude E6410 | 1 | 30 |
| 12 | Chassis | Dell Inc. | OptiPlex 760 | 4 | 2 |
| 11 | Chassis | Dell Inc. | OptiPlex 755 | 0 | 2 |
| 22 | Chassis | Dell Inc. | OptiPlex 780 | 5 | 2 |
| 11 | Chassis | Dell Inc. | OptiPlex 755 | 3 | 43.1 |
| 10 | Chassis | Dell Inc. | Latitude E6410 | 2 | 20 |
| 9 | Chassis | Dell Inc. | Latitude E6400 | 3 | 1.8 |
| 18 | Chassis | Dell Inc. | Latitude E6510 | 0 | 1 |

# Consolidated list of views for extraction:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Component area | View No. | View name | Key column to be used for machine activity determination |
| 1 | Heartbeat discovery | 1 | v\_AgentDiscoveries | AgentTime |
| 2 | State message | 2 | v\_AssignmentStatePerTopic | StateTime |
|  | 3 | v\_GS\_PATCHSTATE | TimeStamp |
|  | 4 | v\_GS\_PATCHSTATEEX | TimeStamp |
|  | 5 | v\_HS\_PATCHSTATE | TimeStamp |
|  | 6 | v\_HS\_PATCHSTATEEX | TimeStamp |
| 3 | Status message | 7 | v\_ClientAdvertisementStatus | LastStatusTime |
|  |  | 8 | v\_CIAssignmentStatus | LastEnforcementMessageTime, LastEvaluationMessageTime |
| 4 | System Activity | 9 | V\_GS\_OPERATING\_SYSTEM | TimeStamp |
|  |  | 10 | v\_GS\_WORKSTATION\_STATUS | TimeStamp |
| 5 | Software inventory | 11 | v\_GS\_LastSoftwareScan: | LastScanDate |