Understanding Managed Status in ITNM.

IBM Middleware User Community
Ask the Experts Series

Bruce Urquhart - urquharb@us.ibm.com
&
Krishna M Kodali – kmkodali@us.ibm.com

Agenda

- Background
- Maintenance Mode
- Unmanaged by disco mode
- Rules for setting the value of Managed status of an entity
- Status propagation
- Events
- RCA
- More about TagManagedEntities.stch

Background

Managed status values:

0 - Managed state. The entity is managed.

1 - Unmanaged state. The entity is unmanaged – can be set via the GUI or perl scripts.

2 - Permanently unmanaged state. The entity is permanently unmanaged. This setting cannot be modified from the GUI. This value is set by the TagManagedEntities.stch.

3 - Unmanaged because the IP address is out of the discovery scope. The device has been discovered through another IP address that is within the discovery scope. This value is set by the TagManagedEntities.stch.

Background

3.9:

- Data is stored in model's master.entityByName and optionally in Disco's scratchTopology.entityByName
- Field is ExtraInfo->m_ManagedStatus
- Model is master and its managed status value is used by RCA.
- Managed Status is subfield in ExtraInfo so a record not necessarily present in Disco or Model.
- Entity record with no ExtraInfo->m_ManagedStatus field represents a managed status=0, i.e: entities are considered managed by default.

Background

4.x:

- No ExtraInfo in Disco nor model.
- Managed status is now stored in managedStatus table in DNCIM and ncimCache because disco's scratchTopology.entityByName is now DNCIM.
- Model is master and its managed status value is used by RCA. (same as 3.9).
- Entity with no record in managedStatus tables indicates managed status=0(managed by default – same as 3.9).

Background

- Managed status also stored in ncim.managedStatus table.
- Separate table allows most common use case to unmanage devices to perform maintenance:
  - subcomponents inside an unmanaged chassis are considered unmanaged.
- No need to add every interface to ncim.managedStatus table as it's redundant.
- If an entry is required for an interface, then a row must be inserted, or updated if already exists.

Background

- Managed status is represented differently in NCIM and model.
- Model has managed status assigned to every interface – this is required for the gateway and RCA processes.
- Entity in model with managed status = {1, 2 or 3} will have all its events excluded from RCA processing.
- Ncim.managedStatus table is excluded from ModelNcimDb.cfg.
  - Special case table that can be updated by GUI and disco process, unlike all other tables in ModelNcimDb.cfg for which the data comes exclusively from disco process.

Maintenance Mode (i.e: unmanaged)

- Temporary state where device(s) or interface(s) are taken off-line for maintenance and then removed or brought back on-line.
- Operators won't need to respond to alerts.
- In GUI, right click on device or interface and select Unmanage, or use UnmanageNode.pl script.
  - Both set managed status=1 and can only happen if current managed status=0.
- Reverse can be done to manage - i.e: right click > Manage or use ManageNode.pl script.
- Using the scripts one can take advantage of using the bulk capability with a list of ips in a file.
- RemoveNode.pl sets managed status =1 and LingerTime=0 in 3.9 and 4.1.1.
- In 4.2, no managed status change, device is removed from ncimCache and subsequently from NCIM – no need for another disco to be run.

Maintenance Mode (i.e: unmanaged)

- Changes to managed status via GUI or scripts are done by an SQL insert/update statement to ncim.managedStatus.
- Model receives these updates as it automatically polls ncim.managedStatus every 30 secs.
  - This is set in the ManagedStatusUpdateInterval field in model.config in $NCHOME/etc/precision/ModelSchema.cfg.
- Model broadcasts any changes to Event Gateway for use by event enrichment including RCA.
- Poller also checks the table in 30 second intervals and resets internal poll scopes.
  - This is set by ManagedStatusUpdateInterval field of the config.properties table in $NCHOME/etc/precision/NcPollerSchema.<domain>.cfg.

Unmanaged by Disco Mode

- Unlike maintenance mode, unmanaged by disco state does not typically fluctuate.
- Automatically set during discovery based on attributes set in TagManagedEntities.stch for certain types of interfaces.
- Sets managed status = 2 or 3.
- Managed status = 3 indicates entity is unmanaged because it is out of discovery scope.
- The GUI and the UnmanageNode.pl/ManageNode.pl scripts do not allow an entity unmanaged by Disco to change its state.

Rules for setting the value of Managed Status of an entity

- Manage status values are stored in model(3.9 - master.entityByName, 4.x- ncimCache.managedStatus) and in ncim.managedStatus.
- Managed status value can be altered by ncp_disco by setting the value in disco (3.9 - scratchTopology.entityByName , 4.x - DNCIM.managedStatus) then sending topology to model.
- GUI can alter value of ncim.managedStatus via SQL interface to NCIM.
- There are rules about which managed status values can be set by disco and GUI - rules also depend on existing managed status values.
- Following tables summarize rules and apply to chassis and subcomponents.

Managed Status values that disco can set

1	0	Set MS=0	Yes	0	#1 Disco is forbidden from setting an entity to managed (0) IF the GUI set the current unmanaged status (1)
2	1		No	1	
3	2		Yes	0	#2 Disco can set an entity to managed (0) only IF the current unmanaged status was set by Disco, i.e. is 2 or 3
4	3		Yes	0	
5	Any (N)	Set MS=1	No	N	#3 Disco is forbidden from setting the value of managed status to 1; only the GUI is allowed to do this.
6	Any (N)	Set MS=2	Yes	2	#4 Disco can always set the value to 2 or 3, even if the GUI previously set it to 1. Disco overrides GUI.
7	Any (N)	Set MS=3	Yes	3	

Managed Status values that GUI can set

#	Existing MS	Action	Is Action Allowed?	New MS	The general rule
1	0	Set MS=0	Yes	0	#5 The GUI is permitted to set an entity to managed (0) IF it was the GUI that previously set it to unmanaged (1)
2	1		Yes	0	
3	2		No	2	#6 The GUI is forbidden from setting an entity to managed (0) IF it is unmanaged by Disco (2, 3)
4	3		No	3	
5	0	Set MS=1	Yes	1	#7 The GUI is permitted to unmanage (1) an entity only IF it is currently managed (0)
6	1		Yes	1	
7	2		No	2	#8 The GUI is forbidden from unmanaging an entity (1) IF Disco has already set it to unmanaged (2, 3)
8	3		No	3	
9	Any (N)	Set MS=2	No	N	#9 The GUI is always forbidden from unmanaging an entity by setting the managed status to 2 or 3
10	Any (N)	Set MS=3	No	N	

Status Propagation

- In NCIM not every entity in containment hierarchy of a device has to have explicit entry in NCIM managedStatus table.
- If chassis managed status = 1, all subcomponents considered unmanaged by default - model managed status will be set to 1.
- If chassis managed status = 0, all subcomponents also considered managed by default.
- Any managed status entry in NCIM for subcomponents = 1,2 or 3 will be set in model.
- Status propagation from chassis to subcomponent occurs for managed status values of 1-3.

RCA

- Events correlated to unmanaged entities will not take part in RCA.
- These events from external sources(no poller events) are ignored by RCA rules so NmosCauseType is not set to root cause or suppressed.
- Managed entities are identified by model's managed status value.
- Event Gateway populates NmosManagedStatus in alerts.status for each event correlated with an entity in topology. Sets it to 0,1,2 or 3 and RCA is calculated only for those events with managed status = 0.
- Unmanaged events can be processed by RCA by modifying 'HonourManagedStatus' attribute value from 1 to 0 under $NCHOME/etc/precision/RCASchema.cfg and restarting ITNM Gateway.

TagManagedEntities.stch

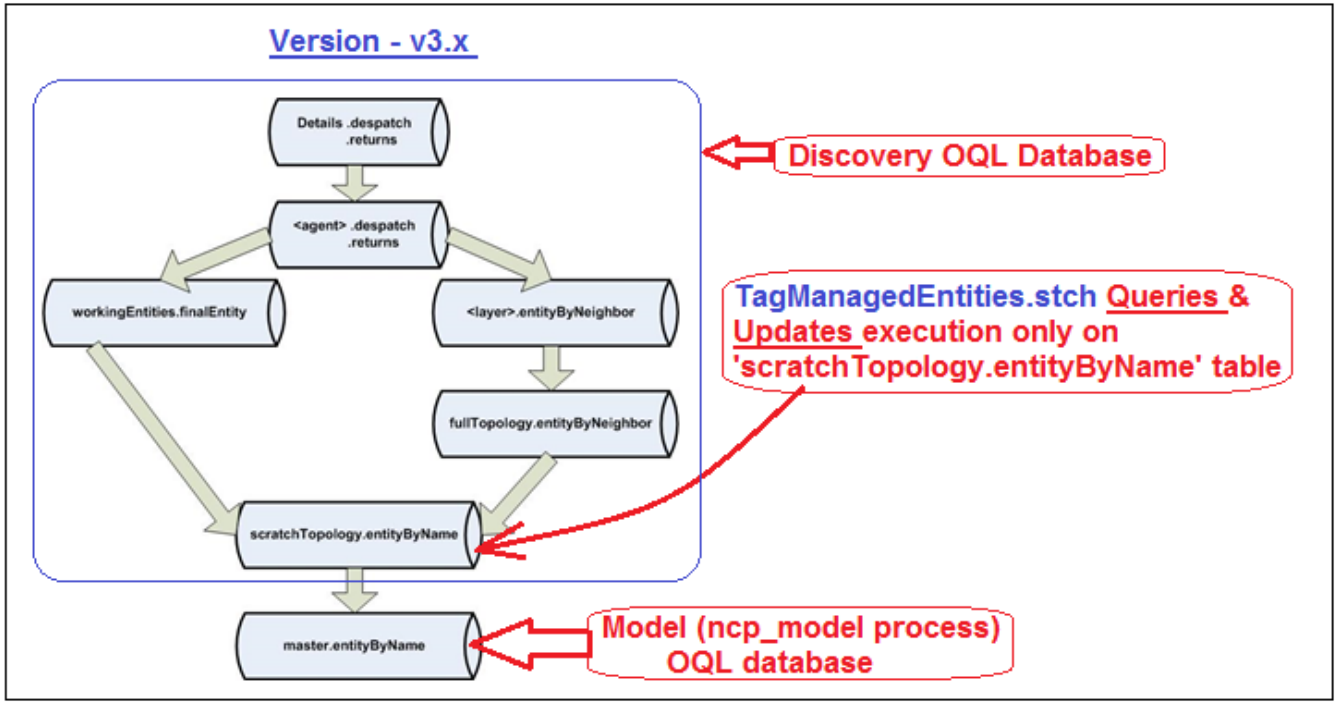
Background:

- TagManagedEntities.stch is used to use unmanage entities via Discovery Engine.
  - Certain type of entities you do not want/need to monitor (e.g. Dialer , Async) & Out of scope Entities etc.
- Location of the stitcher:
  - V3.9 - $NCHOME/precision/disco/stitchers/TagManagedEntities.stch
  - V4.2 - $NCHOME/precision/disco/stitchers/DNCIM/PopulateDNCIM_ManagedStatus.stch
- Logic excerpt from the Stitcher :

```
// These interfaces are set to "Permanently unmanaged". This is
// done to prevent them becoming managed again when a containing
// entity temporarily becomes unmanaged then managed.
text unmanagedFilter =
            "(
                m_LocalNbr->m_IfDescr like 'Dialer'
                OR
                m_LocalNbr->m_IfDescr like 'Async'
                OR
                m_LocalNbr->m_IfDescr like 'Virtual'
                OR
                m_LocalNbr->m_IfDescr like 'Null'
                OR
                m_LocalNbr->m_IfDescr like 'NULL'
                OR
                (m_LocalNbr->m_IfDescr like 'Vlan' AND m_LocalNbr->m_IpAddress = NULL)
                OR
                (m_LocalNbr->m_IfDescr like 'VLAN' AND m_LocalNbr->m_IpAddress = NULL)
                OR
                m_LocalNbr->m_IfAlias like 'NoMon'
            );";
```

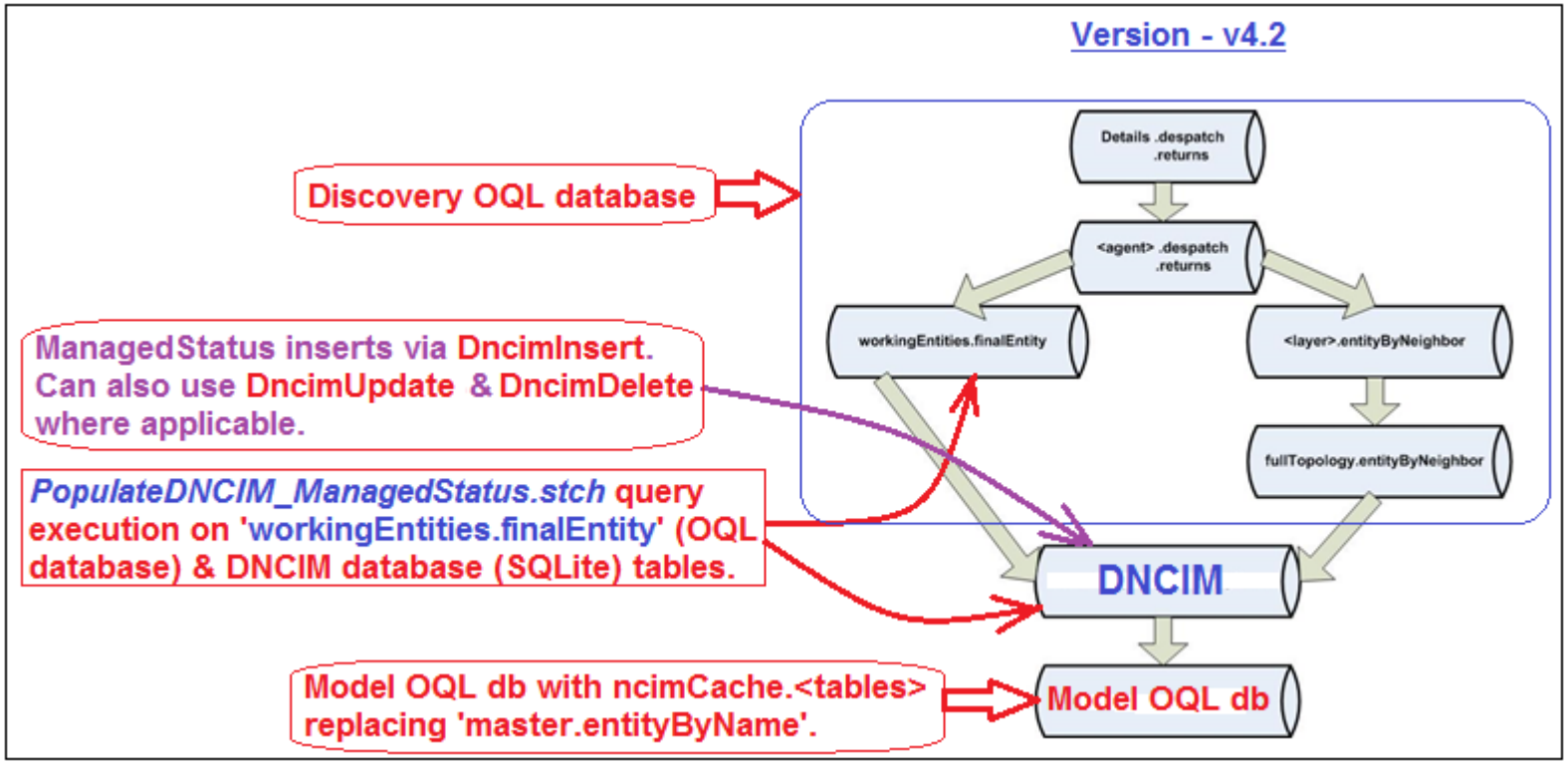
TagManagedEntities.stch

- Data flow and Stitcher execution in v3.9:
  - Direct OQL updates on '*scratchTopology.entityByName*'



PopulateDNCIM_ManagedStatus.stch

- Data flow and stitcher execution in v4.2
  - Query for list of entities either from '*workingEntities.finalEntity*' and/or DNCIM tables
  - Update entity ManagedStatus via DncimInsert into *dncim.managedStatus* table



TagManagedEntities.stch ‘managedStatus’ table definition

ManagedStatus (0, 1, 2 & 3) propagation:

From Discovery -> Model -> NCIM

- What’s honored in v3.9 Vs. v4.2 ?
  - Ideally only values 2 & 3 should be passed from Disco to Model.
  - In v3.9 – you can feed any value from Discovery to Model.
  - In v4.x only 2 and 3 are passed onto Model via Disco.

```
-- ============================================================
-- managedStatus
-- ============================================================

CREATE TABLE managedStatus
(
        entityId                INTEGER NOT NULL,
        status                  SMALLINT DEFAULT 0 NOT NULL,
        username                VARCHAR(255),
        changeTime              TIMESTAMP NOT NULL,
        maintenanceStartTime    TIMESTAMP,
        maintenanceEndTime      TIMESTAMP,
```

From NCIM -> Model:

- Values 0 and 1 either via direct insert/update(s) to ncim.managedStatus table
- Unmanage/Manage tool execution via GUI (Structure Browser , Hop / Network Views, AEL etc.)

TagManagedEntities.stch – Attributes usage

Key changes between v3.9 & v4.2:

- v3.9 stitcher logic against 'scratchTopology.entityByName' mainly on '**ExtraInfo**' vblist.
- V4.x stitcher logic against 'workingEntities.finalEntity' mainly on '**m_LocalNbr**' vblist

Version - 3.9

```
{
        EntityName='vegas-cs3750[ Fa1/0/17 ]';
        BaseName='vegas-cs3750';
        Address=['','00:0F:F7:E2:E9:13',''];
        EntityType=2;
        EntityOID='1.3.6.1.4.1.9.1.516';
        Status=1;
        IsActive=0;
        ExtraInfo={
                m_IfIndex=10017;
                m_IfType=6;
                m_IfDescr='FastEthernet1/0/17';
                m_LocalNbrPhysAddr='00:0F:F7:E2:E9:13';
                m_IfAdminStatus=1;
                m_IfOperStatus=2;
                m_IfName='Fa1/0/17';
                m_IfSpeed=10000000;
                m_IfAlias='';
                m_IfConnectorPresent=1;
                m_IfMTU=1500;
                m_IfHighSpeed=10;
                m_IfPromiscuousMode=2;
                m_CdmAdminState=2;
                m_DisplayLabel='[ Fa1/0/17 ]';
                m_BaseName='vegas-cs3750';
                m_AccessProtocol=NULL;
                m_AddressSpace=NULL;
        };
        UpwardConnections=['vegas-cs3750'];
```

Any attribute from 'scratchTopology.entityByName' table for e.g. ExtraInfo->m_IfName like 'Fa1' etc.

Version-4.2

```
{
        m_UniqueAddress='172.30.29.1';
        m_Name='vegas-cs3750[ Fa1/0/17 ]';
        m_ObjectId='1.3.6.1.4.1.9.1.516';
        m_HaveAccess=1;
        m_Protocol=1;
        m_ManagerId='';
        m_LocalNbr={
                m_IfIndex=10017;
                m_IfType=6;
                m_IfDescr='FastEthernet1/0/17';
                m_LocalNbrPhysAddr='00:0F:F7:E2:E9:13';
                m_IfAdminStatus=1;
                m_IfOperStatus=2;
                m_IfName='Fa1/0/17';
                m_IfSpeed=10000000;
                m_IfAlias='';
                m_IfConnectorPresent=1;
                m_IfMTU=1500;
                m_IfHighSpeed=10;
                m_IfPromiscuousMode=2;
                m_CdmAdminState=2;
                m_DisplayLabel='[ Fa1/0/17 ]';
        };
        m_Creator='Interface';
        m_PhysAddr='00:0F:F7:E2:E9:13';
        m_IsActive=0;
        m_BaseName='vegas-cs3750';
        m_EntityType=2;
}
```

Any attribute from 'workingEntities.finalEntity' table for e.g. m_LocalNbr->m_IfName / m_EnttiyType etc.

PopulateDNCIM_ManagedStatus.stch – Scenarios

Scenario #1 – ‘Unmanage all 'proprietary serial' links i.e. ifType = 22’ with DeviceName containing ‘NewYork’

– Best choice would be – extend existing stitcher logic to include ‘ifType = 22’

- V3.9:
  - ExtraInfo->m_Type =22 and EntityName like ‘NewYork’
- V4.2:
  - m_LocalNbr->m_IfType = 22 and m_EntityName like ‘NewYork’

- But if you would like to have your own stitcher, without touching default stitcher, you can use this reference

```
[OMNI-NM-netcool]#cat PopulateDNCIM_K2.stch
UserDefinedStitcher
{
    StitcherTrigger
    {}
    StitcherRules
    {
        int domainId = eval(int,'$ARG_1');
        text oql = NULL;
        text select = "select m_Name from workingEntities.finalEntity where ";
        text interfaceFilter = " m_EntityType = 2 AND ";
        text unmanagedFilter =
                    "(
                        m_LocalNbr->m_IfDescr like 'Serial'
                    );";
        oql = select + interfaceFilter + unmanagedFilter;
        RecordList unmanagedList = RetrieveOQL(oql);
        text entityName = NULL;
        int entityId = -1;
        int createTime = -1;
        StartSQLTransaction("DNCIM");
        foreach (unmanagedList)
        {
            entityName = eval(text, '&m_Name');
            entityId = ExecuteStitcher('GetEntityId', domainId, entityName, 1);
            if(entityId > 0)
            {
                createTime = eval(int,'$TIME');
                Record unmanagedInsert;
                @unmanagedInsert.entityId = entityId;
                @unmanagedInsert.status = 2;
                @unmanagedInsert.changeTime = eval(time, '$createTime');
                DncimInsert( entityId, "managedStatus", unmanagedInsert );
                delete(unmanagedInsert);
            }
            else
            {
                Print("Warning: Unable to set managedStatus to 2 for entity ", entityName
                    , " as the entity does not exist within DNCIM entityData.");
            }
        }
        CommitSQLTransaction("DNCIM");
    }
}
```

TagManagedEntities.stch – Scenarios cont'd

Scenario#2 – 'Unmanage all Ethernet interfaces that aren't connected' to any devices

– In this case, Ethernet Interfaces are 'ifType = 6' , more about 'ifType' values can be read @ https://ibm.biz/BdrVbD

- V3.9 - ExtraInfo->m_IfType = 6 and (ExtraInfo->ReverseRelatedTo is NULL and RelatedTo is NULL)
- V4.2 – Query DNCIM & use **DncimInsert**.

```
[OMNI-NM-netcool]#cat PopulateDNCIM_Unmanage_unconnected_interfaces.stch
UserDefinedStitcher
{
    StitcherTrigger
    {
    }
    StitcherRules
    {
        int domainId = eval(int,'$ARG_1');
        int entityId = 0;
        SQLData myData = NULL;
        int createTime = -1;
        myData = PrepareSQL(
        "select ed.entityId, ed.entityName from dncim.entityData ed
        inner join dncim.interface i on ed.entityId=i.entityId where i.ifType=6 and ed.entityType = 2
        and
        ( ed.entityId NOT IN ( select aEndEntityId from connects c) AND ed.entityId NOT IN ( select zEndEntityId from connects c) );"
        , "DNCIM" ,
        eval(int,'$domainId')
        );
        ExecuteSQL(myData);
        foreach(myData)
        {
                entityId = eval(int, '&ENTITYID');
                createTime = eval(int,'$TIME');
                Record unmanagedInsert;
                @unmanagedInsert.entityId = entityId;
                @unmanagedInsert.status = 2;
                @unmanagedInsert.changeTime = eval(time, '$createTime');
                DncimInsert( entityId, "managedStatus", unmanagedInsert );
                delete(unmanagedInsert);
        }
        delete(myData);
        CommitSQLTransaction("DNCIM");
    }
}
```

TagManagedEntities.stch – Scenarios cont'd

Scenario#2 – 'Unmanage all Ethernet interfaces that aren't connected' to any devices – cont'd

– Call new stitcher either at the end of 'PopulateDNCIM_ManagedStatus.stch' or call it under 'PopulateDNCIM.stch'

– See below excerpt from PopulateDNCIM.stch:

```
ExecuteStitcher('PopulateDNCIM_ManagedStatus', domainId, isRediscovery, dynamicDiscoNode);
ExecuteStitcher('PopulateDNCIM_Unmanage_unconnected_interfaces', domainId, isRediscovery, dynamicDiscoNode);

StitcherTimeCheck('DNCIM managed status', 'Sending DNCIM to model', 97);

ExecuteStitcher('SendDNCIMChangesToModel');

StitcherTimeCheck('Sending DNCIM to model' , 'Next Discovery', 100);

// Ensure all transactions are complete at the end of the processing
CommitSQLTransaction( "DNCIM" );

// Now that all of the data has been inserted analyze the DB to gather the final stats
AnalyzeSQLStats('DNCIM','dncim');
```

TagManagedEntities.stch

Scenario # 3 – Combine multiple scenarios under one stitcher – part#1

```
[OMNI-NM-netcool]#cat TwoExamples.stch
UserDefinedStitcher
{
    StitcherTrigger
    {
    }
    StitcherRules
    {
        int domainId = eval(int,'$ARG_1');
        int entityId = 0;
        SQLData myData = NULL;
        SQLData myData1 = NULL;
        int createTime = -1;
        createTime = eval(int,'$TIME');

//------ First stanza to unmanage all Interfaces that aren't connected to any devices------//
           myData = PrepareSQL(
        "select ed.entityId, ed.entityName from dncim.entityData ed
        inner join dncim.interface i on ed.entityId=i.entityId where i.ifType=22 and ed.entityType = 2
        and
        ( ed.entityId NOT IN ( select aEndEntityId from connects c) AND ed.entityId NOT IN ( select zEndEntityId from connects c) );"
        , "DNCIM" ,
        eval(int,'$domainId')
        );
        ExecuteSQL(myData);
        foreach(myData)
        {
                entityId = eval(int, '&ENTITYID');
                Record unmanagedInsert;
                @unmanagedInsert.entityId = entityId;
                @unmanagedInsert.status = 3;
                @unmanagedInsert.changeTime = eval(time, '$createTime');
                DncimInsert( entityId, "managedStatus", unmanagedInsert );
                delete(unmanagedInsert);
        }
        delete(myData);
// ------ End of First stanza------------------//
```

TagManagedEntities.stch

Scenario # 3 – Combine multiple scenarios under one stitcher –part#2

Note – if the required values are available under ‘workingEntities.finalEntity’, its best to extend ‘PopulateDNCIM_ManagedStatus.stch’ logic instead of adding overhead on DNCIM.

```
// Second Stanza to unmanage all Voice Gateway 64Kbps Interfaces ----------//
        myData1 = PrepareSQL(
        "select ed.entityId, ed.entityName from dncim.entityData ed
        inner join dncim.interface i on ed.entityId=i.entityId where i.ifType=77 and i.ifSpeed=64000 ;"
        , "DNCIM" ,
        eval(int,'$domainId')
        );
        ExecuteSQL(myData1);
        foreach(myData1)
        {
                entityId = eval(int, '&ENTITYID');
                Record unmanagedInsert;
                @unmanagedInsert.entityId = entityId;
                @unmanagedInsert.status = 77;
                @unmanagedInsert.changeTime = eval(time, '$createTime');
                DncimInsert( entityId, "managedStatus", unmanagedInsert );
                delete(unmanagedInsert);
        }
        delete(myData1);
// ------ End of Second stanza--------------//
        CommitSQLTransaction("DNCIM");
    }
}
[OMNI-NM-netcool]#
```

TagManagedEntities.stch

Stitcher validation & other notes:

– Stitcher changes (including DNCIM ones) are dynamic and no need to restart any processes.

V3.9 :

- You can check in-memory stitcher definition using ncp_oql against service ‘Disco’

#ncp_oql –domain <domainName> -service Disco –query “select * from stitchers.definitons where m_Name=‘TagManagedEntities’;”

- You won’t know if the execution is successful until the stitcher is called.

V4.2:

- You can’t retrieve stitcher definition via ncp_oql
- Best is review ncp_disco.<domainname>.log for errors or updates. You can only review log/trace to see if any errors after saving and/or execution.

➢ OQL functions to retrieve data: ‘ExecuteOQL’ goes with ‘RetrieveOQL’

➢ SQL (DNCIM & NCIM): ‘ExecuteSQL’ is only works with a ‘PrepareSQL’ object

➢ OQL is case sensitive for table and column names unlike SQL (DNCIM & NCIM)

Quick Demo

&

Questions?

Text version of the stitchers:

```
#cat PopulateDNCIM_Unmanage_unconnected_interfaces.stch
UserDefinedStitcher
{
    StitcherTrigger
    {
    }
    StitcherRules
    {
        int domainId = eval(int,'$ARG_1');
        int entityId = 0;
        SQLData myData = NULL;
        int createTime = -1;
        myData = PrepareSQL(
        "select ed.entityId, ed.entityName from dncim.entityData ed
        inner join dncim.interface i on ed.entityId=i.entityId where i.ifType=6 and ed.entityType = 2
        and
        ( ed.entityId NOT IN ( select aEndEntityId from connects c) AND ed.entityId NOT IN ( select zEndEntityId from connects c) );"
        , "DNCIM" ,
        eval(int,'$domainId')
        );
        ExecuteSQL(myData);
        foreach(myData)
        {
                entityId = eval(int, '&ENTITYID');
                createTime = eval(int,'$TIME');
                Record unmanagedInsert;
                @unmanagedInsert.entityId = entityId;
                @unmanagedInsert.status = 2;
                @unmanagedInsert.changeTime = eval(time, '$createTime');
                DncimInsert( entityId, "managedStatus", unmanagedInsert );
                delete(unmanagedInsert);
        }
        delete(myData);
        CommitSQLTransaction("DNCIM");
    }
}
```

Text version of the stitchers: Two scenarios in one stitcher

```
#cat TwoExamples.stch
UserDefinedStitcher
{
    StitcherTrigger
    {
    }
    StitcherRules
    {
        int domainId = eval(int,'$ARG_1');
        int entityId = 0;
        SQLData myData = NULL;
        SQLData myData1 = NULL;
        int createTime = -1;
        createTime = eval(int,'$TIME');
//------ First stanza to unmanage all Interfaces that aren't connected to any devices------//
            myData = PrepareSQL(
        "select ed.entityId, ed.entityName from dncim.entityData ed
        inner join dncim.interface i on ed.entityId=i.entityId where i.ifType=22 and ed.entityType = 2
        and
        ( ed.entityId NOT IN ( select aEndEntityId from connects c) AND ed.entityId NOT IN ( select zEndEntityId from connects c) );"
        , "DNCIM" ,
        eval(int,'$domainId')
        );
        ExecuteSQL(myData);
        foreach(myData)
        {
                entityId = eval(int, '&ENTITYID');
                Record unmanagedInsert;
                @unmanagedInsert.entityId = entityId;
                @unmanagedInsert.status = 3;
                @unmanagedInsert.changeTime = eval(time, '$createTime');
                DncimInsert( entityId, "managedStatus", unmanagedInsert );
                delete(unmanagedInsert);
        }
        delete(myData);
// ------ End of First stanza------------------//
```

Text version of the stitchers: Two scenarios in one stitcher – cont'd

```
// Second Stanza to unmanage all Voice Gateway 64Kbps Interfaces ----------//
        myData1 = PrepareSQL(
        "select ed.entityId, ed.entityName from dncim.entityData ed
        inner join dncim.interface i on ed.entityId=i.entityId where i.ifType=77 and i.ifSpeed=64000 ;"
        , "DNCIM" ,
        eval(int,'$domainId')
        );
        ExecuteSQL(myData1);
        foreach(myData1)
        {
                entityId = eval(int, '&ENTITYID');
                Record unmanagedInsert;
                @unmanagedInsert.entityId = entityId;
                @unmanagedInsert.status = 77;
                @unmanagedInsert.changeTime = eval(time, '$createTime');
                DncimInsert( entityId, "managedStatus", unmanagedInsert );
                delete(unmanagedInsert);
        }
        delete(myData1);
// ------ End of Second stanza--------------//
        CommitSQLTransaction("DNCIM");
    }
}
```