

# **MDB, Hardware Assets and CORA**



## LEGAL NOTICE

This publication is based on current information and resource allocations as of its date of publication and is subject to change or withdrawal by CA at any time without notice. The information in this publication could include typographical errors or technical inaccuracies. CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice.

Any reference in this publication to non-CA products and non-CA websites are provided for convenience only and shall not serve as CA's endorsement of such products or websites. Your use of such products, websites, and any information regarding such products or any materials provided with such products or at such websites shall be at your own risk.

Notwithstanding anything in this publication to the contrary, this publication shall not (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this publication remain at CA's sole discretion.

The information in this publication is based upon CA's experiences with the referenced software products in a variety of development and customer environments. Past performance of the software products in such development and customer environments is not indicative of the future performance of such software products in identical, similar or different environments. CA does not warrant that the software products will operate as specifically set forth in this publication. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product.

Certain information in this publication may outline CA's general product direction. All information in this publication is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "AS IS" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if CA is expressly advised of the possibility of such damages.

### COPYRIGHT LICENSE AND NOTICE:

This publication may contain sample application programming code and/or language which illustrate programming techniques on various operating systems. Notwithstanding anything to the contrary contained in this publication, such sample code does not constitute licensed products or software under any CA license or services agreement. You may copy, modify and use this sample code for the purposes of performing the installation methods and routines described in this document. These samples have not been tested. CA does not make, and you may not rely on, any promise, express or implied, of reliability, serviceability or function of the sample code.

Copyright © 2008 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. Microsoft product screen shots reprinted with permission from Microsoft Corporation.

# MDB, Hardware Assets and CORA

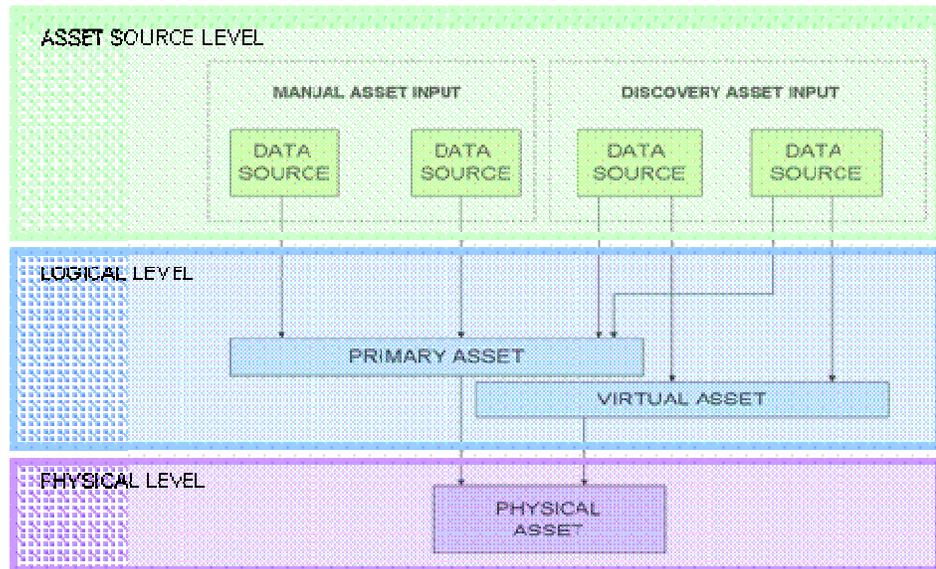
## Table of Contents

<b>MDB, Hardware Assets and CORA</b> .....	<b>1</b>
<b>Table of Contents</b> .....	<b>1</b>
<b>MDB, Hardware Assets and CORA</b> .....	<b>1</b>
Asset Matching Logic.....	2
Discovered vs. owned assets.....	5
A Note about Asset Classes.....	10
Methods for Populating the MDB.....	12
<b>CORA and Unicenter Desktop and Server Management</b> .....	<b>14</b>
Managed vs. Unmanaged.....	15
CORA Parameters for DSM.....	17
DSM and Asset Classes.....	18
<b>CORA and Unicenter Asset Portfolio Management</b> .....	<b>20</b>
Asset Naming Conventions.....	21
CORA Parameters for UAPM .....	23
UAPM and Asset Classes .....	23
<b>CORA and Unicenter NSM</b> .....	<b>27</b>
CORA Parameters for NSM .....	28
<b>CORA and Unicenter Service Desk\CA CMDB</b> .....	<b>30</b>
Migrating from an Earlier USD Release .....	31
CORA Parameters for USD .....	31
USD Asset Classes .....	33
<b>Universal Federation Adapters and the CA CMDB</b> .....	<b>35</b>
SMS Adapter for CA CMDB .....	38
NSM Adapter for CA CMDB.....	38
<b>Useful Links and References</b> .....	<b>40</b>



## MDB, Hardware Assets and CORA

All Unicenter r11 products utilize the common MDB schema to store and manage their data. As the interface through which these assets are registered and as the only source for updating these tables, the Common Registration API (CORA) ensures that asset data flows consistently, thereby supporting the data and referential integrity of the MDB's **master asset** data model.



The master asset data model consists of the following 3 levels of asset references:

- The **asset source level**, which consists of the `ca_asset_source` table, and is used to track assets as they enter the system from different data sources, whether input manually or through discovery.
- The **logical asset level**, which consists of the `ca_logical_asset` and `ca_logical_asset_property` tables, and is used to store virtual assets. The logical asset level acts as a middle layer that exists between the data source and the physical level to accommodate assets embedded in other assets such as VMWare sessions or dual-boot scenarios.
- Finally, the **physical asset level**, which consists of the `ca_asset` table, stores the identifiers that define the object as a distinct, physical asset.

Once CORA is given a set of registration identifiers from the calling r11 application, it will perform one of the following actions:

- **Return the asset source reference** if the registration identifiers match an existing asset, thus preventing duplicate assets from being registered.

- **Insert** a new physical, logical, logical property, or asset source record into the database depending on where the mismatch occurs. This step also prevents duplication of data by inserting records only at the appropriate levels. For instance, if there are no physical assets that can be identified by the registration identifiers, a new physical asset is created. However, if a physical asset can be identified by the registration identifiers, but not a logical asset, then a new logical asset is created and linked to the existing physical asset.
- **Update** an existing identifier(s) in the database with one of the registration identifiers. In this scenario, a single physical asset can be identified by the registration identifiers and one or more identifiers need to be updated.
- **Merge** two physical or logical assets together. In this scenario, CORA received information indicating that two or more physical assets are, in fact, the same asset. The existing physical assets are merged together to form one asset and information for each asset is stored in ca\_logical\_asset\_property table.

For r11.1, when a product registers an asset and CORA generates a UUID that matches an existing asset, it also automatically links (reconciles) Owned and Discovered information for that asset.

To determine which CORA version is being used by the product, execute the following command:

```
CORAVER
```

## Asset Matching Logic - Updated

When an asset is registered, CORA generates the asset uuid (ca\_asset) by applying a black-box logic to the following six properties:

- Serial Number
- Asset Tag (appearing as Alt Asset ID)
- Host Name
- Mac Address
- DNS Name
- Asset Label (Name)

CORA applies the following weighting system to these properties to determine if a match exists. Since certain properties are considered "more important" than others, CORA will recognize a duplicate based on those values alone.

- **Serial Number** is the most highly weighted field. Two assets with the same serial number will **always** be matched by CORA *unless* Asset Tag or Host Name are different.

- **Alt Asset ID** is the second most highly weighted field. Serial Number and Alt Asset ID appear at the highest level of the Asset Registration schema in ca\_asset).
- **Host Name** appears in the middle level (ca\_logical\_asset). If Serial Number and Alt Asset ID are blank the Host Name takes precedence over DNS and MAC Address values. Although more than one DNS/MAC pair can be specified for the same Host Name, it will still be considered the same Asset.
- **DNS Name** and **MAC Address** are weighted the same. CORA will recognize the same asset if DNS or MAC address match and will create a new asset when they do not.
- Finally, although **Asset Label (Name)** is required to create an asset, you can have multiple assets with the same name as long all the other CORA fields are empty.

The following table shows how CORA determines uniqueness of an Asset. The intent here is not to show every single combination but to show enough of the behavior so that one could determine what would occur based on the properties they choose to include when registering an Asset in release r11.

Serial Number	Asset Tag	Host Name	DNS Name	MAC Address	Asset Label	Results
Unique	Unique	Unique	Unique	Unique	Unique, Duplicate or Null	New Asset
Unique	Null	Null	Null	Null	Unique, Duplicate or Null	New Asset
Null	Unique	Null	Null	Null	Unique, Duplicate or Null	New Asset
Null	Null	Unique	Null	Null	Unique, Duplicate or Null	New Asset
Null	Null	Null	Unique	Null	Unique, Duplicate or Null	New Asset
Null	Null	Null	Null	Unique	Unique, Duplicate or Null	New Asset
Null	Null	Null	Null	Null	Unique	New Asset
Null	Null	Null	Null	Null	Duplicate	Duplicate
Null	Null	Null	Unique	Duplicate	Unique, Duplicate or Null	Duplicate
Null	Null	Null	Duplicate	Unique	Unique, Duplicate or Null	Duplicate
Null	Null	Null	Unique	Unique	Unique, Duplicate or Null	New Asset
Unique	Duplicate	Duplicate	Duplicate	Duplicate	Unique, Duplicate or Null	New Asset
Duplicate	Unique	Duplicate	Duplicate	Duplicate	Unique, Duplicate or Null	New Asset
Duplicate	Duplicate	Unique	Duplicate	Duplicate	Unique, Duplicate or Null	New Asset
Duplicate	Duplicate	Duplicate	Unique	Duplicate	Unique, Duplicate or Null	Duplicate
Duplicate	Duplicate	Duplicate	Duplicate	Unique	Unique, Duplicate or Null	Duplicate
Duplicate	Duplicate	Duplicate	Unique	Unique	Unique, Duplicate or Null	Duplicate
Duplicate	Duplicate	Duplicate	Duplicate	Duplicate	Unique, Duplicate or Null	Duplicate

Another behavior of CORA which is represented below shows other ways duplicates can occur.

Serial Number	Asset Tag	Host Name	DNS Name	MAC Address	Asset Label	Results
Null	ABC	Null	Null	Null	Unique or Null	New Asset
123	ABC	Null	Null	Null	Unique, Duplicate or Null	Duplicate of ABC
789	XYZ	Null	Null	Null	Unique or Null	New Asset
Null	XYZ	Null	Null	Null	Unique, Duplicate or Null	Duplicate of XYZ

The following table is provided for our existing Service Desk customers. It shows the names of the CORA properties as they are displayed on the Service Desk Asset form as well as the names of the 6 properties from a Database Schema perspective.

Serial Number	Asset Tag	Host Name	DNS Name	MAC Address	Asset Label	Results
Null	ABC	Null	Null	Null	Unique or Null	New Asset
123	ABC	Null	Null	Null	Unique, Duplicate or Null	Duplicate of ABC
789	XYZ	Null	Null	Null	Unique or Null	New Asset
Null	XYZ	Null	Null	Null	Unique, Duplicate or Null	Duplicate of XYZ

### Discovered vs. Owned Assets

Attributes for each asset are divided into "Discovered" and "Owned" in order to facilitate reconciliation and verification capabilities. Unicenter NSM and Unicenter Desktop and Server Management comprise the "discovered" assets whereas "owned" assets are associated with Unicenter Service Desk, CA-CMDB (CI) and UAPM. When multiple products are in use, there are many ways to reconcile owned and discovered assets:

- Use of pdm\_nsmimp to import NSM discovered assets as owned assets
- Use of pdm\_discimp to import DSM discovered assets as owned assets
- CA CMDB GRLoader

Unicenter NSM and Unicenter DSM can also create "owned" assets from discovered assets when a USD ticket is opened.

**Note:** If the asset already exists as "owned", it is only linked to the "discovered" asset thereby avoiding duplication.

More details regarding a specific product's owned or discovered assets can be found later in this document.

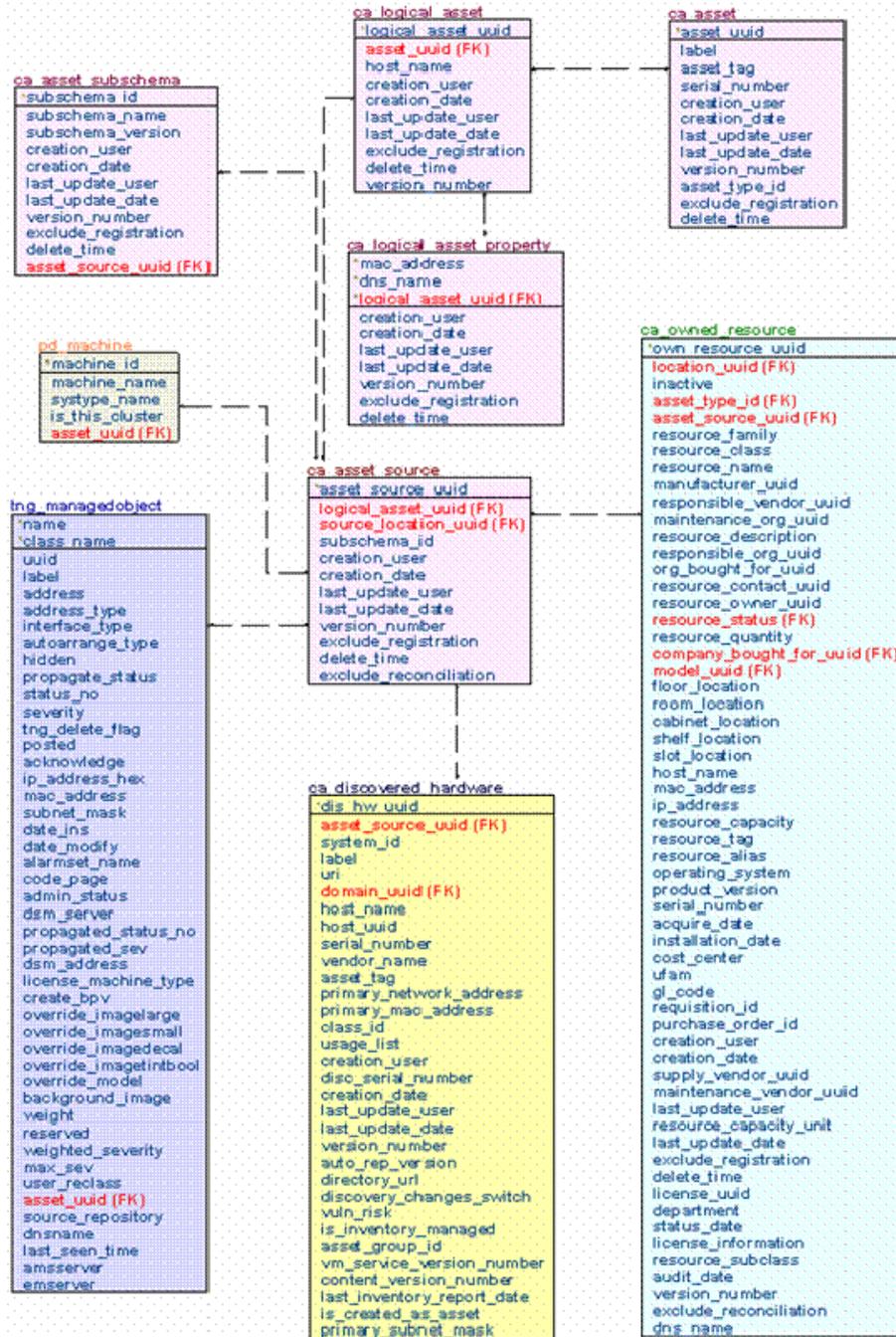
### Asset Tables and Registration

The primary tables used to identify data sources are:

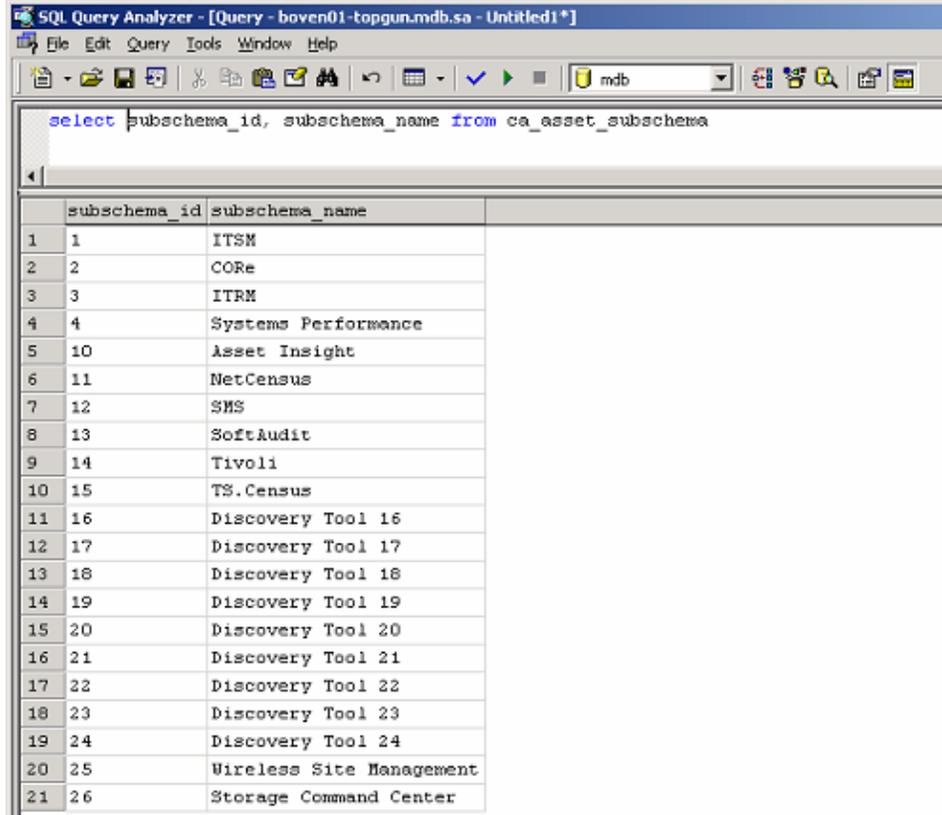
- CA\_DISCOVERED\_HARDWARE
- TNG\_MANAGEDOBJECTS
- PD\_MACHINE

■ CA\_OWNED\_RESOURCE

To understand how these tables relate to one another, consider the following graphic.



The **ca\_asset\_source** table contains the subschema\_id column which identifies the origin of the asset. The subschema\_id values are maintained in ca\_asset\_subschema as shown with the following query:



As you can see:

- "ITSM" objects, which includes "Owned" sources such as Unicenter Asset Portfolio Management, Unicenter Service Desk and CA CMDB (ca\_owned\_resources), have a subschema\_id of "1".
- Unicenter NSM (tng\_managedobject) objects have a subschema\_id of "2."
- Unicenter Desktop and Server Management objects (ca\_discovered\_hardware) have the subschema\_id of "3."
- System Performance Management objects (pd\_machine) have a subschema\_id of "4."

If an asset is registered in the MDB by *different products*, CORA only registers that asset once then links the information from the different data sources. This ensures that the ca\_asset table will only have a single unique entry for each asset.

The following screenshots provide a walk through of the queries executed after a sample asset is registered by Unicenter Asset Portfolio Management, Unicenter Service Desk, Unicenter NSM and Unicenter Desktop and Server Management. Note that the order in which the products register the asset is **not** relevant to the process.

First, from Machine name into **ca\_asset**:

Query - boven01-topgun.mdb.sa - Untitled2\*

```
select * from ca_asset where label like '%boven01%'
```

	asset_uuid	label	asset_tag	serial_number
1	0xE7FC0779EE22424389C28A30FD23D607	boven01-topgun	NULL	56dd501

Then, from ca\_asset into **ca\_logical\_asset** (using asset\_uuid):

Query - boven01-topgun.mdb.sa - Untitled2\*

```
select * from ca_logical_asset where asset_uuid in (
select asset_uuid from ca_asset where label like '%boven01%')
```

	logical_asset_uuid	asset_uuid	host_name
1	0xECE60606A217C74D9E8D7C0D836901D8	0xE7FC0779EE22424389C28A30FD23D607	boven01-topgun

The **ca\_logical\_asset\_property** will show the logical instances of the same asset. For instance, if the same asset is registered by CORA with different DNS and/or MAC address but same Host name, CORA recognizes it is the same assets and stores 2 logical instances in this table:

Query - boven01-topgun.mdb.sa - Untitled2\*

```
select * from ca_logical_asset_property where logical_asset_uuid in(
select logical_asset_uuid from ca_logical_asset where asset_uuid in (
select asset_uuid from ca_asset where label like '%boven01%'))
w
```

	dns_name	mac address	logical_asset_uuid	creation_date
1	BOVEN01-TOPGUN	000C295ACD6A	0xECE60606A217C74D9E8D7C0D836901D8	1160141977
2	boven01-topgun.ca.com	000C295ACD6A	0xECE60606A217C74D9E8D7C0D836901D8	1160141840

**NOTE:** In this example the DNS name input by UAPM (USD, CMDDB) did not use the fully qualified name as it was discovered by NSM (DSM). It is only an example.

Then, from ca\_logical\_asset to **ca\_asset\_source** (using logical\_asset\_uuid):

```
Query - boven01-topgun.mdb.sa - Untitled2*
select * from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%boven01%'))
```

	asset_source_uid	logical_asset_uid	source_location_uid	subschema_id
1	0xD93E456B08FE604C9D8461E13E94DC55	0xECE60606A217C74D9E8D7C0D836901D8	NULL	1
2	0xED4DABA4834CDD4FBAB89D8BB31FDE01	0xECE60606A217C74D9E8D7C0D836901D8	NULL	2
3	0xEE19F52014604C4D8ED5C211C10CFCCA	0xECE60606A217C74D9E8D7C0D836901D8	NULL	3

**Note:** Here you can see the different data sources from the subschema\_id value

Then, from ca\_asset\_source into DSM **ca\_discovered\_hardware** (using asset\_source\_uid):

```
Query - boven01-topgun.mdb.sa - Untitled2*
select * from ca_discovered_hardware where asset_source_uid in (
select asset_source_uid from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%boven01%')))
```

	dis_hw_uid	host_name	domain_uid	label	serial_number
1	0xEE19F52014604C4D8ED5C211C10CFCCA	boven01-topgun	0x041D21D1F5607047819DA5EAD0162A67	boven01-topgun	56dd501

Then, from ca\_asset\_source into UAPM/USD **ca\_owned\_resource** (using asset\_source\_uid):

```
Query - boven01-topgun.mdb.sa - Untitled2*
select * from ca_owned_resource where asset_source_uid in (
select asset_source_uid from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%boven01%')))
```

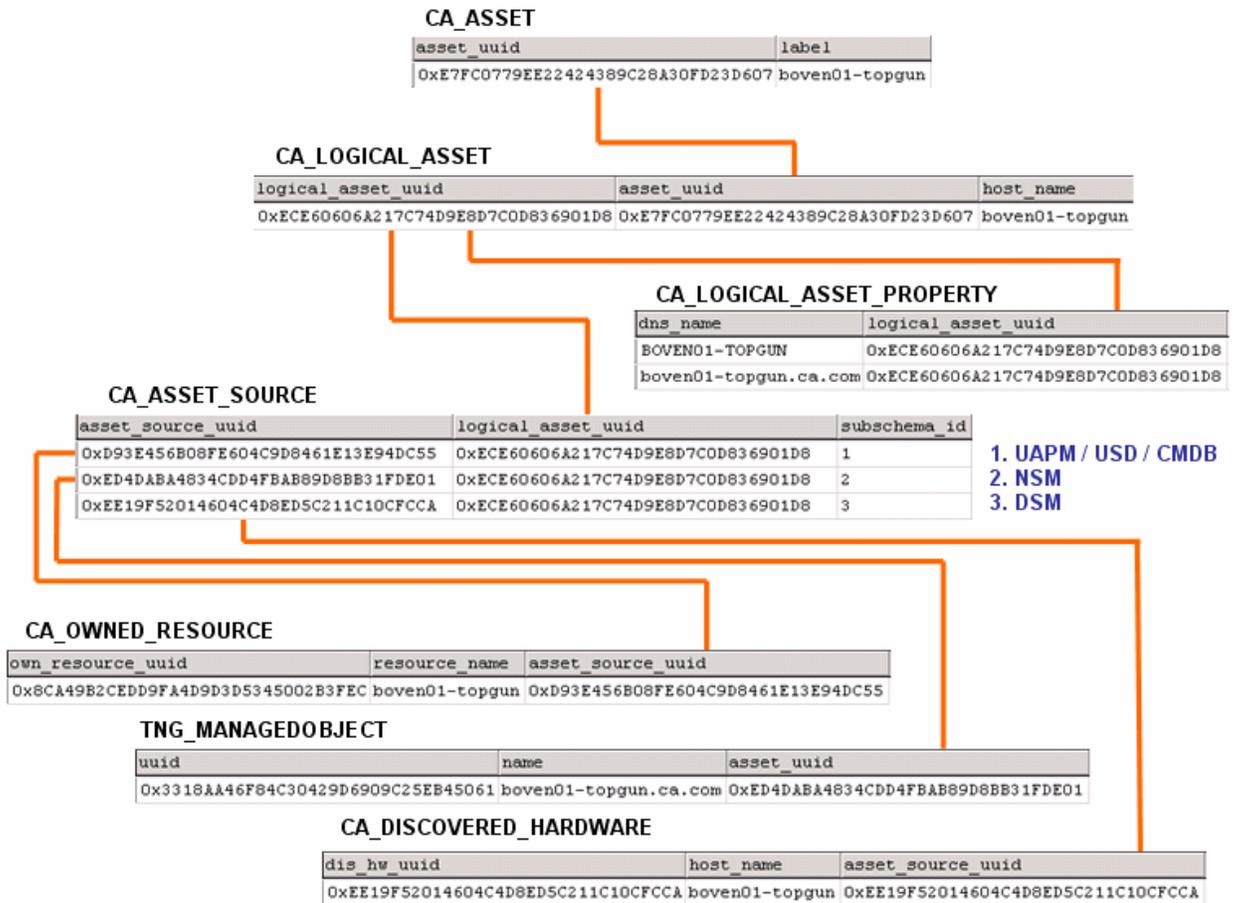
	own_resource_uid	inactive	asset_type_id	resource_name	resource_description
1	0x8CA49B2CEDD9FA4D9D3D5345002B3FEC0	0	1	boven01-topgun	NULL

Then, from ca\_asset\_source into Unicenter NSM **tng\_manageobject** (using asset\_source\_uid):

```
Query - boven01-topgun.mdb.sa - Untitled2*
select uuid, label, address, asset_uid from tng_managedobject where asset_uid in (
select asset_source_uid from ca_asset_source where logical_asset_uid in (
select logical_asset_uid from ca_logical_asset where asset_uid in (
select asset_uid from ca_asset where label like '%boven01%')))
```

	uuid	label	address	asset_uid
1	0x3318AA46F84C30429D6909C25EB45061	boven01-topgun.ca.com	130.119.20.221	0xED4DABA4834CDD4FBAB89D8BB31FDE01

These links can be summarized in the following picture:



## A Note about Asset Classes

Although "class" is a mandatory field to for asset registration, it is not a field that is used by CORA. The concept of an asset "class" is, in fact, interpreted differently by different products. For example:

- In **USD**, the concept of "family" is used to identify the highest level of definition for a CI and each family can consist of one or more "classes" to allow for a more granular categorization of CIs. Further, each family has an extension table that defines the attributes that will be visible in the CI Detail page. When CA CMDB is implemented, it includes over 50 families and over 140 classes that are each stored in the MDB and shared between USD and UAPM.
- When the MDB used by the CA CMDB is shared with **UAPM**, those CMDB families are shared and are known to UAPM as "asset types" for "models" and "assets." In other words, for UAPM:

CMDB Families = UAPM Asset Types

CDMBClasses = UAPM Classes

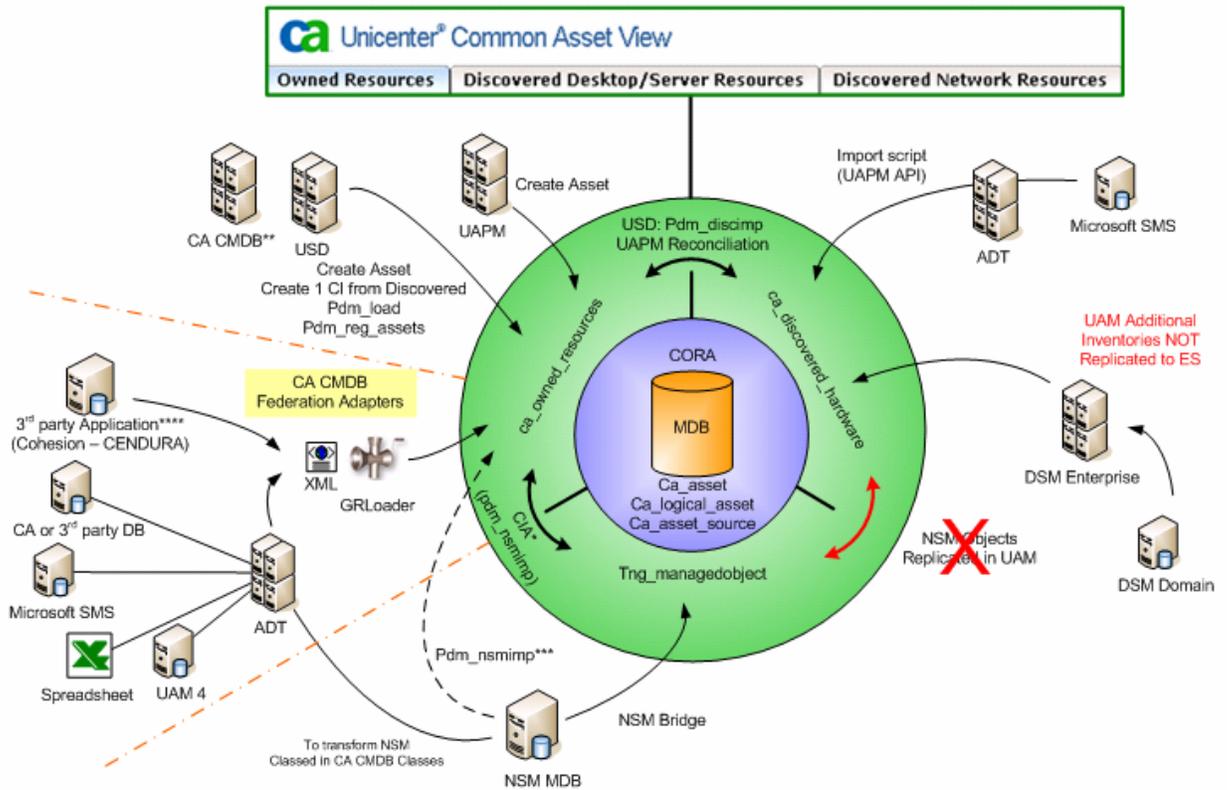
In UAPM, the Asset Type is a family\_id field for ca\_model\_def and a resource\_family for ca\_owned resource.

- Unicenter **DSM**, on the other hand, does not use families and classes to register discovered assets. However, if USD is also installed and integrated with DSM, when DSM initiates the creation of a USD ticket and the registration of a discovered asset as "owned" it uses the default family of "Hardware and Class: Discovered Hardware."
- Only a small number of **NSM** classes match the CA CMDB classes, however, procedures are provided in the USD CA CMDB *Administrator Guide* for mapping USD\CA CMDB classes to NSM classes

**Note:** Since multiple USD/CMDB classes can be mapped to the same NSM Class, the pdm\_nsmimp can't use the NSM class to determine which Class to use when creating the asset because it doesn't know how to pick the correct one if multiple ones are mapped.

## Methods for Populating the MDB

The following graphic depicts the different methods various r11.x products use to register assets in the MDB as well as the tables used to store “owned” or “discovered” information.



**NOTES:**

\*CIA cannot “translate” NSM Classes into USD/CA-CMDB Classes: CIs created with this method will maintain the NSM Class. To transform NSM Class we need to use ADT

\*\*CA CMDB can be installed stand-alone or with USD

\*\*\*available but not recommended. NSM info must be in the same MDB for Asset Viewer to work

\*\*\*\*3<sup>rd</sup> party application able to export in CA XML Format (required by GRLoader)

Regardless of how they are registered, these assets are maintained and reconciled through CORA and a complete view of this information is available through the AMS Common Asset Viewer as long as the data are in the same MDB.

More information on these processes is provided in the following product-specific sections. Note, however, that Unicenter Service Delivery Suite, which is comprised of Service Catalog, Accounting, Assure and SMA, is not included in this document because these products do not register assets in the MDB through CORA. In addition, a discussion of how SMS data can be imported into the MDB via ADT, using the UAPM API, is provided in a separate document titled "Advantage Data Transformer and Microsoft SMS Data" which is available on the R11 implementation cd.

Information on importing SMS data through the GRLoader utility, however, can be found in the CA CMDB section.

## CORA and Unicenter Desktop and Server Management

Unicenter Desktop and Server Management assets are registered with the MDB in the following manner:

- When the Unicenter Desktop and Server Management (DSM) agent is installed it registers with the Scalability Server and an asset (computer) is created.

**Note:** External Asset types can be created from the Unicenter DSM Explorer. These assets are registered when inventory is collected using the associated MIF file.

- “Discovered” Assets are created by the Replication Engine and managed by CORA at the Unicenter DSM Enterprise level CORA. They can also be manually registered using Unicenter DSM Explorer (*unmanaged* = Unicenter DSM Agent not installed)

**Note:** The asset\_uuid can be different in the Enterprise and Domain level MDB.

- Unicenter DSM creates “owned” assets from the managed “discovered” asset when a Unicenter Service Desk ticket is opened for a policy violation or a failed software distribution job.

**Note:** If the asset already exists as “owned”, Unicenter DSM only *links* it to the “discovered” asset thereby avoiding duplication.

It is also important to note that, in an architecture in which USD shares an MDB with the DSM Enterprise, there won't be duplication of assets if the incidents are coming directly from the DSM domain.

The following screenshot shows 2 incidents for the same asset: one generated using the DSM Enterprise Console and 1 using the DSM Console:

The screenshot shows the Unicenter Service Desk / CA CMDB interface. The main content area displays the Configuration Item Detail for 'bradu01-dsmr11'. Below this, there is a navigation menu with various options like 'Attributes', 'CMDB Relationships', 'Inventory', 'Log', 'Service', etc. At the bottom, an 'Incident List' table shows two incidents:

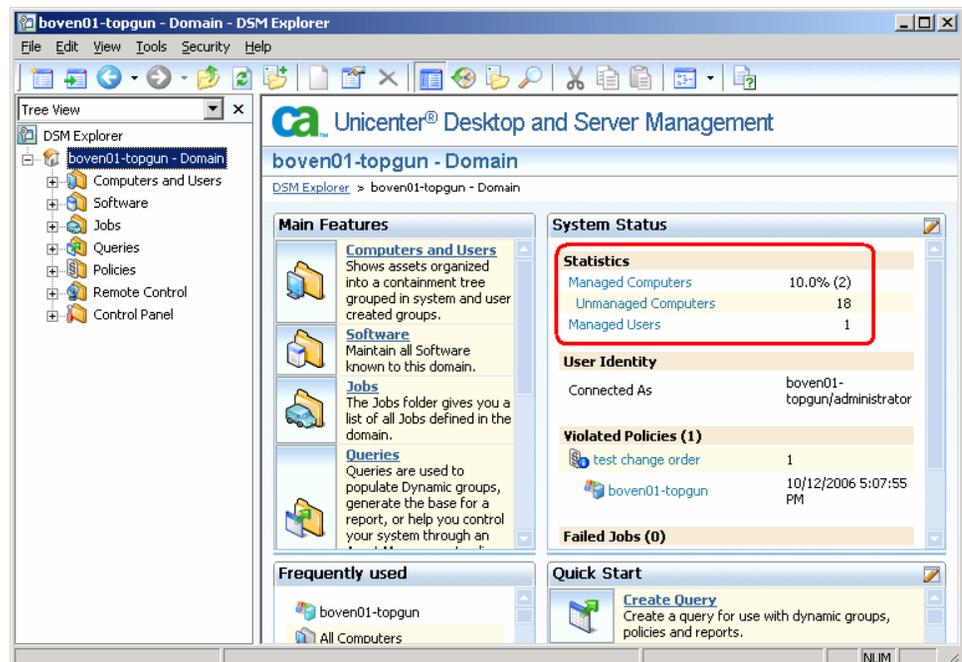
Incident # / Summary	Date Opened	Status
63 Test incident raised from the DSM Enterprise GUI	10/12/2006 10:38 am	Open
62 Test incident raised from the DSM Domain GUI	10/12/2006 10:37 am	Open

## Managed vs. Unmanaged

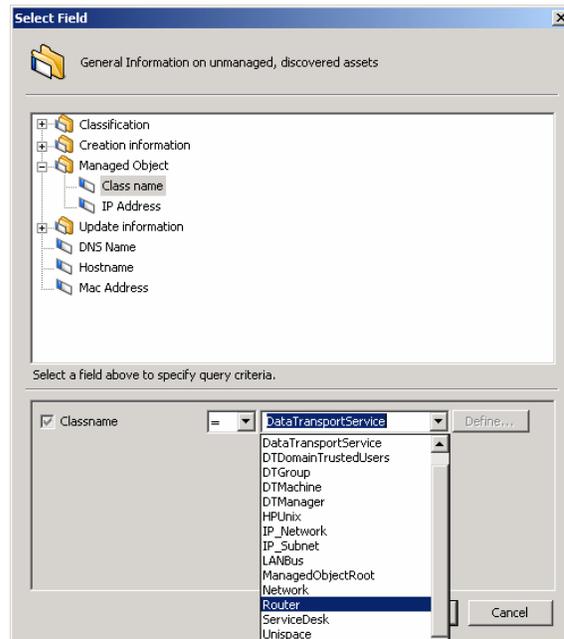
In the 11.2 release of DSM (which was in beta at the time this document was originally written) integration with NSM is re-introduced. The Worldview Synchronization engine job, which was used in the previous releases to import NSM objects to the UAM database, however, is no longer available.

DSM does provide statistics and queries about those objects that are discovered by NSM but not managed by DSM Agents. DSM identifies these assets as **Unmanaged**.

The key point of this integration is the capability of scheduling DSM agent deployment using the NSM Continuous Discovery, “transforming” the unmanaged assets into “managed” assets.

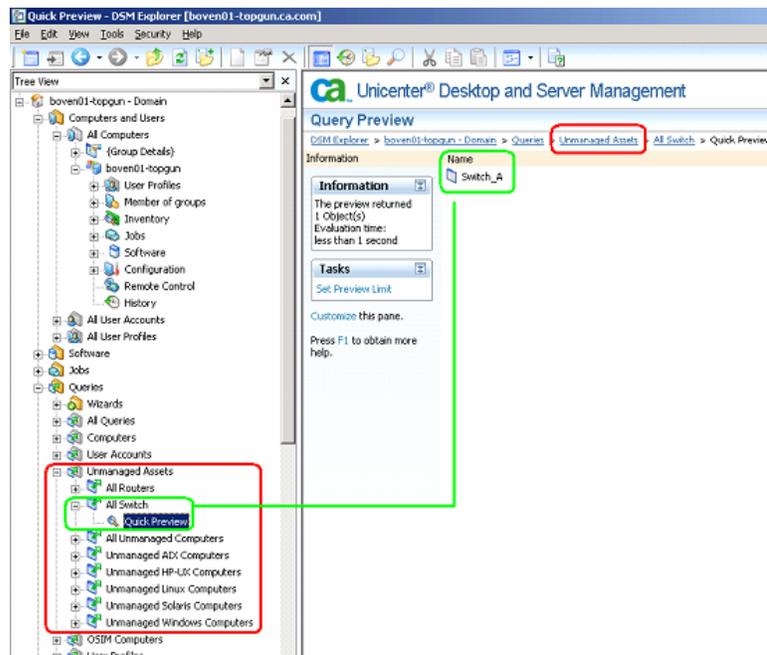


For those assets that are not classified as “Computers” you can execute DSM queries to view and report NSM objects in the DSM Explorer. When NSM discovers objects, the NSM class becomes available in the DSM Query Wizard drop down list and the NSM object is displayed in the results:



**For DSM Explorer to display the NSM object it is essential that NSM registers them using CORA.**

The following screen shot shows the DSM out-of-the-box unmanaged asset queries and a query to display NSM Discovered switches:



The queries in the example identify:

- The UNMANAGED objects (discovered by Unicenter NSM but, without an installed Unicenter DSM agent) in the MDB:

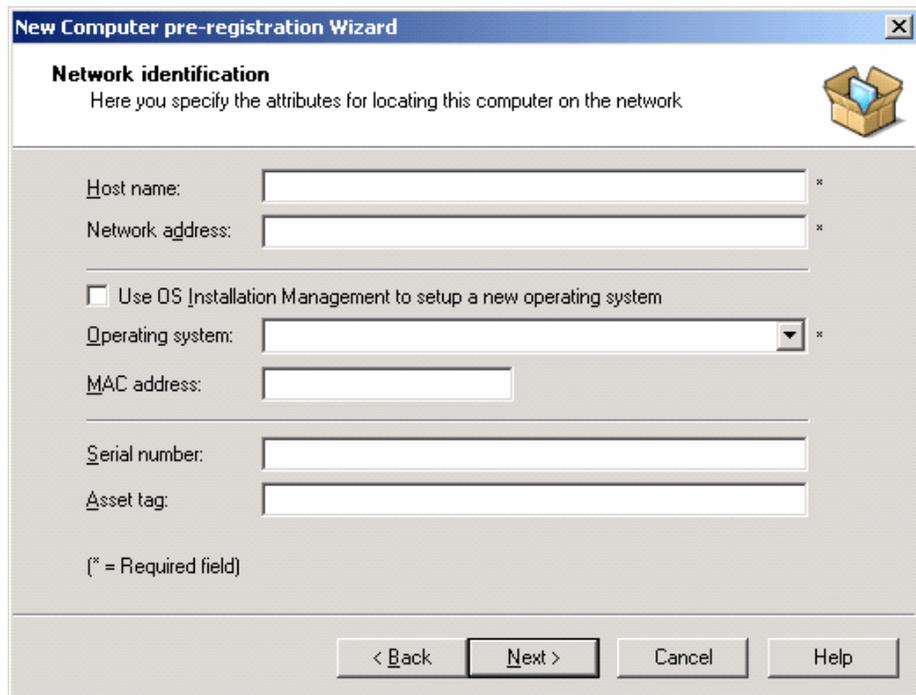
```
select 'Unmanaged Hosts' , uuid_to_char(a.asset_uuid), p.dns_name from
ca_asset a, ca_asset_source s, ca_logical_asset l, ca_logical_asset_property
p where ( (a.asset_uuid = l.asset_uuid) and ((l.logical_asset_uuid =
s.logical_asset_uuid) and (s.subschema_id =2)) and (l.logical_asset_uuid =
p.logical_asset_uuid) and not exists (select * from ca_asset_source s2
where (s2.logical_asset_uuid = l.logical_asset_uuid) and (s2.subschema_id
=3)))
```

- The MANAGED objects (Unicenter NSM discovered and Unicenter DSM agent installed):

```
select 'Managed Hosts' , uuid_to_char(a.asset_uuid), l.host_name from
ca_asset a, ca_asset_source s, ca_logical_asset l where ( (a.asset_uuid =
l.asset_uuid) and ((l.logical_asset_uuid = s.logical_asset_uuid) and
(s.subschema_id =3)) and exists (select * from ca_asset_source s2 where
(s2.logical_asset_uuid = l.logical_asset_uuid) and (s2.subschema_id =2)))
```

## CORA Parameters for DSM

DSM allows assets to be manually pre-registered. To do this, launch the “New Computer pre-registration Wizard” from the DSM Explorer.



To register an unmanaged asset, you need to identify the Host Name, Network Address and Operating System (from the DSM prepopulated list). The Asset Label will be the same as the Host Name.

## DSM and Asset Classes

As noted earlier, DSM does not use Families and Classes to register discovered assets although, if DSM initiate the creation of a USD ticket and the registration of a discovered asset as "Owned" it will use the default Family (Hardware) and Class (Discovered Hardware).

If the CA CMDB is also installed and if the CI Attributes are vital for the implementation, you can, optionally modify the Extension table for the "Hardware" family so that, for example, the attributes of the CMDB Family: Hardware. Server will be displayed.

The screenshots show the 'Hardware Configuration Item Family Detail' page in the Unicenter Service Desk / CA CMDB interface. The top screenshot shows the default state where the 'Name' is 'Hardware' and the 'Record Status' is 'Active'. The bottom screenshot shows the 'Extension Name' field populated with 'ci\_hardware\_server', which is circled in red. The 'Description' for both is 'Hardware Device family'.

Name	Extension Name	Record Status
Hardware		Active

Name	Extension Name	Record Status
Hardware	ci_hardware_server	Active

After ca\_hardware\_server is added to the Extension table, CIs belonging to Hardware Family will display the CMDB attributes:

Unicenter® Service Desk / CA CMDB

Logged in as: Administrator (Log Out) (Close)

File View Search Reports Window Help

bradu01-redhes Configuration Item Detail Edit Asset Viewer CMDB Visual

Name	Class	Family	Active?	
bradu01-redhes	Discovered Hardware	Hardware	Active	
Serial Number	Alt CI ID	Host Name	DNS Name	MAC Address
NRZKV		bradu01-redhes	wsp031320wss.ca.com	00A0C96F7E22
Notes				

3. Inventory		4. Log		5. Service	
6. Location	7. Contacts	8. Organizations	9. Service Contracts	10. Incidents	
11. Problems	12. Requests	13. Change Orders	14. Issues	15. Impact Analyzer	
1. Attributes			2. CMDB Relationships		

Attributes

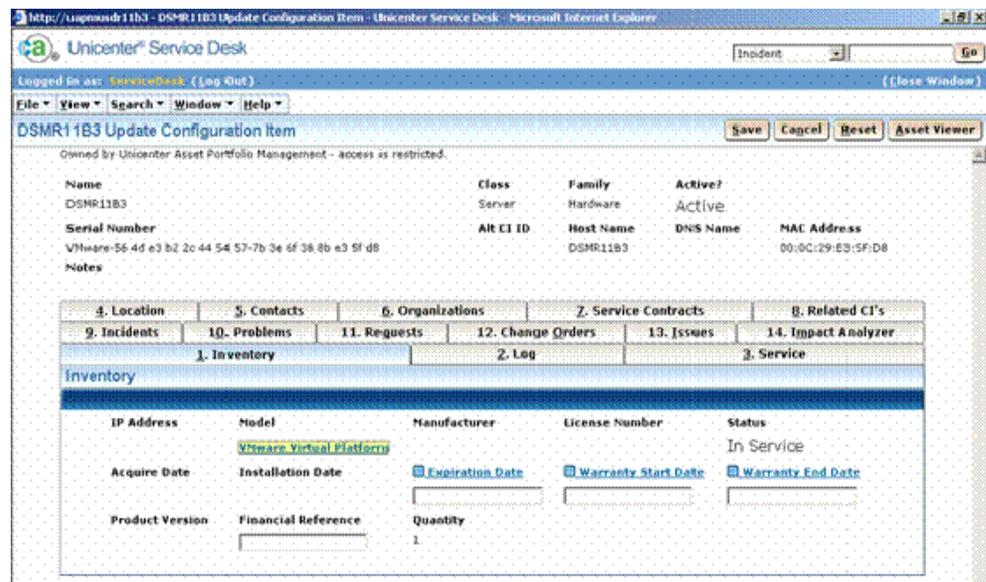
Memory Installed	Memory Capacity	Disk Capacity	Processor Type	
Processor Speed	Disk Type	CD Rom Type	Network Card	Monitor Model
Printer	Technology	Processor Capacity	Number of Processors Installed	Processor Cache
Number of Memory Slots	Number of Memory Slots Used	Type of Network Connection	Number of Network Cards	Number of Network Ports

## CORA and Unicenter Asset Portfolio Management

Under Unicenter Asset Portfolio Management, “owned” assets (and Models) are created automatically by the Reconciliation Engine using information from “discovered” asset and according to the following rules:

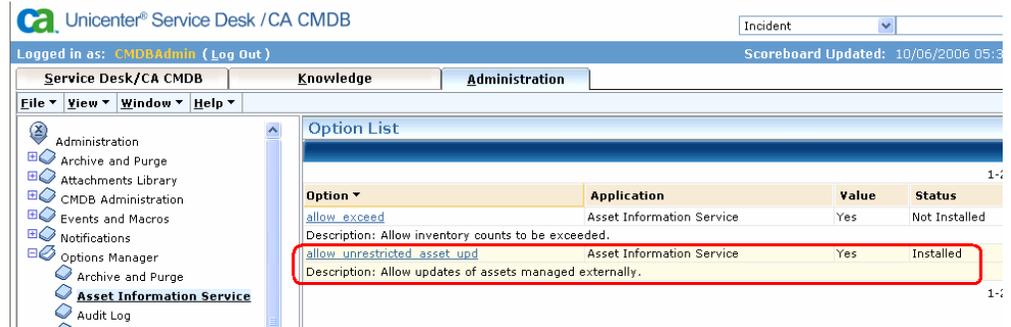
- If the “owned” asset already exists, the Reconciliation Engine will link it to the “discovered” asset based on the defined Matching Criteria Rules.
- Once reconciled, both owned and discovered information are linked from the Common View (also from Unicenter Service Desk.)
- Owned assets can also be manually created from UAPM models through the Web Interface

Owned assets are **shared** with Unicenter Service Desk but the “**Managed by UAPM**” flag is used to identify which product actually manages the asset. If an asset is “Managed by Unicenter Asset Portfolio Management,” then a majority of its attributes can only be updated from the UAPM interface. The following screen shot demonstrates how a “Managed by UAPM” asset is displayed by Unicenter Service Desk in Edit mode:



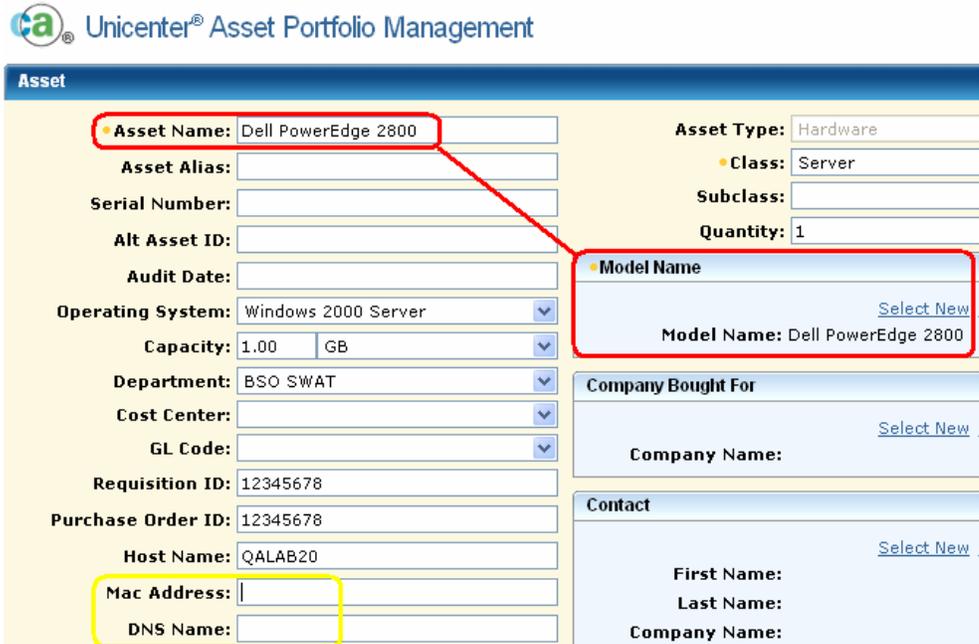
When UAPM generates tickets in Unicenter Service Desk it will not create a duplicate Owned Asset.

In r11.2, Unicenter Service Desk introduced an option, installed by default, to override the UAPM flag and allow updates of the assets.



### Asset Naming Conventions

Historically, when a UAPM asset was created the "Model Name" was used as the "asset name."



With the introduction of the MDB and with "owned" tables shared across products, however, this convention would prevent Unicenter Service Desk from correctly displaying the asset by default. To counter this, you have the option to use the "Host Name" as the asset name. In r11, this option defaults to **TRUE**; in r11.1, this option defaults to **FALSE**.

Consider the following UAPM API configuration options:

**Asset Portfolio Management Configurations**

**1. Choose Your Database**

Ingres Database      Server Name:

Microsoft SQL Server Database      Database Name:

TableOwner:

**2. Choose Your Settings**

State Data:

Attachment Path (UNC):

Maximum List Size:

Session Timeout (minutes):

System Reminder Start:

Script Timeout (seconds):

User Query Timeout (seconds):

Retries:

Retry Interval (seconds):

Log File Name:

Log Level:

Cache Service Port:

Load Asset Name From Host Name:

Enable Audits     Enable Scripts     Enable Events

APM Inventory Web Service:

Maximum Records Returned:

**3. Choose Your Security Type**

Windows Integrated Security with Single DB Login

Windows Basic Security with Single DB Login

EIAM with Single DB Login

External

EIAM Host Name:

APM DB Login ID:

APM DB Password:

Confirm APM Password:

**4. Choose Your Asset Viewer (AMS) Settings**

Asset Viewer Path:

Tomcat Startup Port:

Tomcat Shutdown Port:

Following are the configuration options for the Reconciliation Engine:

**Reconciliation Engine Configurations**

**1. Choose Your Database**

Ingres Database      Server Name:

Microsoft SQL Server Database      Database Name:

TableOwner:

**2. Choose Your Settings**

Script Timeout (seconds):

User Query Timeout (seconds):

Retries:

Retry Interval (seconds):

Log File Name:

Log Level:

Cache Service Port:

Enable Audits     Enable Scripts     Enable Events

Engine Debug Level:

Event Log Override:

Refresh Lock Every Number of Records:

Days Until Delete of Rec Modification:

Email Server:

Email From Address:

Web Server:

Load Asset Name From Host Name:

**3. Choose Your Security Type**

APM Database Level Security

APM DB Login ID:

APM DB Password:

## CORA Parameters for UAPM

In UAPM, assets can be created only based on Models and the model determines the Asset Type (=CMDB family). Asset Class is a mandatory field and it is retrieved based on the Asset Type.

Asset Name and Acquired Date are also mandatory fields for saving the new asset. CORA reconciliation is done only when the Save button is clicked and the mandatory information provided.

The screenshot shows the 'Asset (New)' form in Unicenter Asset Portfolio Management. The form is divided into several sections:

- Asset Menu:** A sidebar on the left with expandable options like 'Activity summary for asset', 'Asset as primary object in re...', 'Attachments for asset', etc.
- Asset Fields:** A central area with fields for 'Asset Name', 'Asset Alias', 'Serial Number', 'Alt Asset ID', 'Audit Date', 'Operating System', 'Capacity', 'Department', 'Cost Center', 'GL Code', 'Requisition ID', 'Purchase Order ID', 'Host Name', 'Mac Address', 'DNS Name', 'Floor Location', 'Room Location', 'Shelf Location', and 'Slot Location'.
- Model Information:** A section for 'Model Name' with a dropdown menu and 'Select New Details Clear' links.
- Company Bought For:** A section for 'Company Name' with a dropdown menu.
- Contact:** A section for 'First Name', 'Last Name', and 'Company Name' with dropdown menus.
- Location:** A section for 'Location Name' with a dropdown menu.
- Seller Company:** A section for 'Company Name' with a dropdown menu.
- Current Status:** A section with dropdown menus for 'Version Name', 'Portfolio Status', and 'Date Acquired'.
- Managed by UAPM:** A section with checkboxes for 'Managed by UAPM' (checked), 'Exclude Reconciliation', and 'Inactive'.

## UAPM and Asset Classes

When the CA CMDB is deployed and when the MDB used by CA CMDB is shared with UAPM, the CMDB families are also shared and they are known to UAPM as Asset types for Models and Assets:

CMDB Families = UAPM Asset Types

CMDB Classes = UAPM Classes

In UAPM an Asset Type is a family\_id field for ca\_model\_def and resource\_family for ca\_owned\_resource:

The screenshot shows two panels from the UAPM configuration interface:

- Field Information:**
  - Object Label: Asset
  - Database Table Name: ca\_owned\_resource
  - Label: Asset Type
  - Database Field Name: resource\_family
  - Description: foreign key to the id field of the
  - Store Audit Trail Data:
  - Has Dropdown List:
  - Required:
  - Add index to this field:
- Dropdown List Customization:**
  - Name: Asset Type
  - Buttons: Insert Row, Delete Row
  - Table:

	Reconcile	Value
1	<input type="checkbox"/>	Cluster
2	<input type="checkbox"/>	Cluster.Resource
3	<input type="checkbox"/>	Cluster.Resource Group
4	<input checked="" type="checkbox"/>	Computer
5	<input type="checkbox"/>	Contact
6	<input type="checkbox"/>	Contract
7	<input type="checkbox"/>	Document
8	<input type="checkbox"/>	Facilities.Air Conditioning

In UAPM we can view the Asset Type (CMDB Family) and the Classes created by CA CMDB, and vice versa.

For example let's look at the Hardware.Server Family which has many Classes associated to it:

The screenshot shows the UAPM Customization interface for the 'Class' asset type. The 'Field Information' panel is configured as follows:

- Object Label: Model
- Database Table Name: ca\_model\_def
- Label: Class
- Database Field Name: class\_id
- Description: foreign key to the id field of the ca\_resource\_class table for
- Store Audit Trail Data:
- Has Dropdown List:
- Required:
- Add index to this field:

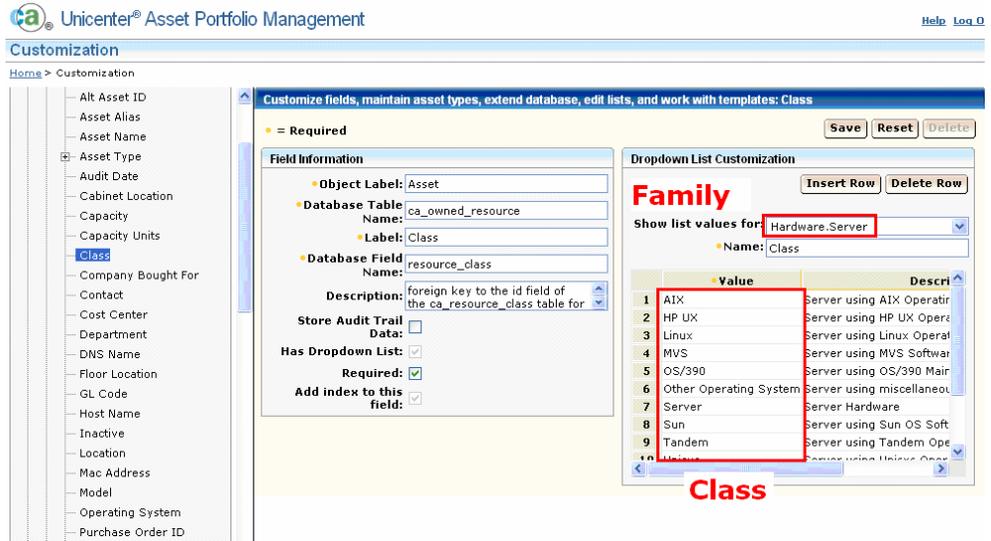
The 'Dropdown List Customization' panel shows the following configuration:

- Show list values for: Cluster
- Name: Cluster
- Table:

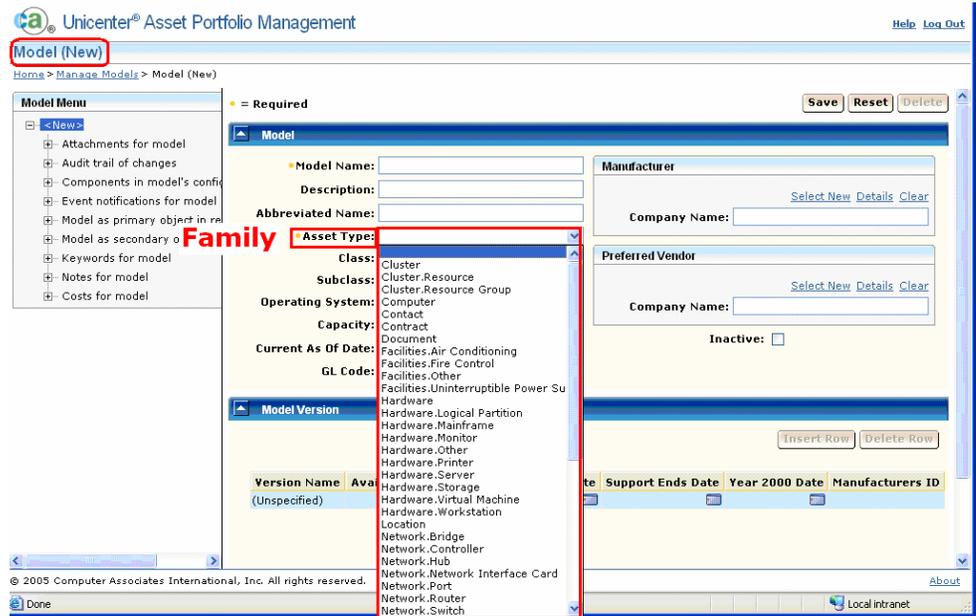
	Value
1	Cluster

A red box highlights the dropdown list of values, and a red arrow points to it with the label 'Family'. The list includes various asset types such as Cluster, Cluster.Resource, Computer, Contact, Contract, Document, Facilities.Air Conditioning, Facilities.Fire Control, Facilities.Other, Facilities.Uninterruptible Power S, Hardware, Hardware.Logical Partition, Hardware.Mainframe, Hardware.Monitor, Hardware.Other, Hardware.Printer, Hardware.Server, Hardware.Storage, Hardware.Virtual Machine, Hardware.Workstation, Location, Network.Bridge, Network.Controller, Network.Hub, Network.Network Interface Card, Network.Port, Network.Router, Network.Switch, and Organization.

These values are provided by CACMDB content out-of-the-box.



When registering asset with UAPM the class values are determined by the Asset Type (CMDB Family) used when the UAPM Model was created:



They are displayed when the asset is created:

The screenshot shows the 'Asset (New)' form in the Unicenter Asset Portfolio Management system. The form is titled 'Asset (New)' and includes a navigation breadcrumb 'Home > Manage Assets > Asset (New)'. On the left, there is an 'Asset Menu' with various options like 'Activity summary for asset', 'Asset as primary object in rel', etc. The main form area is divided into sections: 'Asset' and 'Family'. The 'Asset' section contains fields for 'Asset Name' (Dell Precision), 'Asset Alias', 'Serial Number', 'Alt Asset ID', 'Audit Date', 'Operating System', 'Capacity', 'Department', 'Cost Center', 'GL Code', and 'Requisition ID'. The 'Family' section contains 'Asset Type' (Hardware.Server), 'Class' (dropdown), 'Subclass' (dropdown), 'Quantity', 'Model Name', 'Company Bought For', and 'Company Name'. A red box highlights the 'Class' dropdown menu, which is open to show a list of operating systems: AIX, HP\_UX, Linux, MVS, OS/390, Other Operating System, Server, Sun, Tandem, Unisys, Unix, Vax, VM, and Windows. A red arrow points to the word 'Class' in the form.

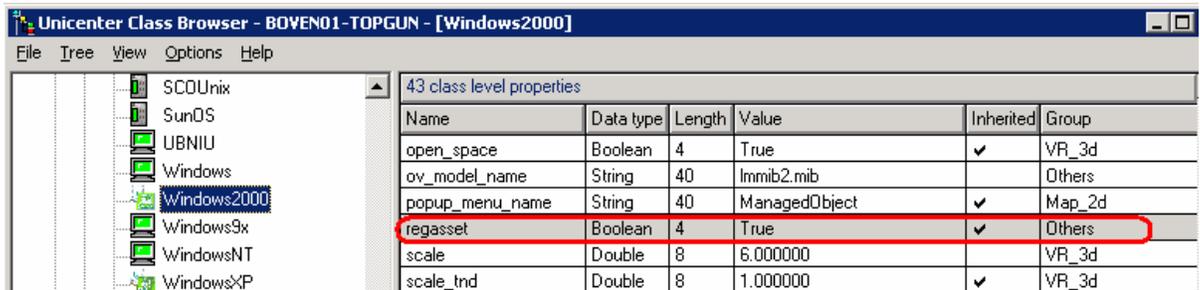
## CORA and Unicenter NSM

Unicenter NSM “discovered” assets are added to the tng\_managedobject table through :

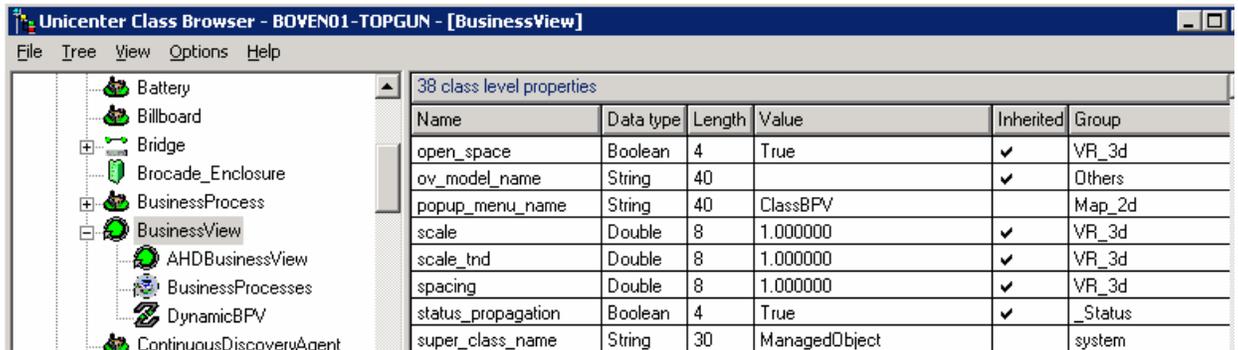
- AutoDiscovery
- Continuous Discovery
- dscvrbe command
- manually in the 2Dmap in Design Mode
- with NSM Bridge

In the r11.x MDB, device classes (such as Host, Hub, Printer, Router, Switch, and Workstation) of the ManagedObject class have a Class Level Property called **regasset** that determines if CORA will be used to register the discovered object or not.

If the **regasset = True** the NSMClass will be registered through CORA, and can be reconciled when other products register the same asset through CORA.



If the regasset property is either **missing** or set to **False**, NSM will discover the object and store all the information without CORA. As a result, any other products sharing the MDB will not be aware of that object.



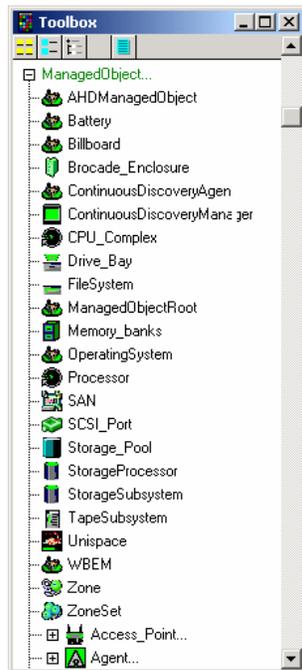
The NSM Bridge can be used to “federate” NSM MDBs. Because it registers NSM Discovered objects using CORA, this guarantees consistency across the MDBs.

Unicenter NSM also creates “owned” assets from the “discovered” asset when a Unicenter Service Desk ticket is opened. If the asset already exists as “owned,” Unicenter NSM merely links it to the “discovered” asset to avoid duplication.

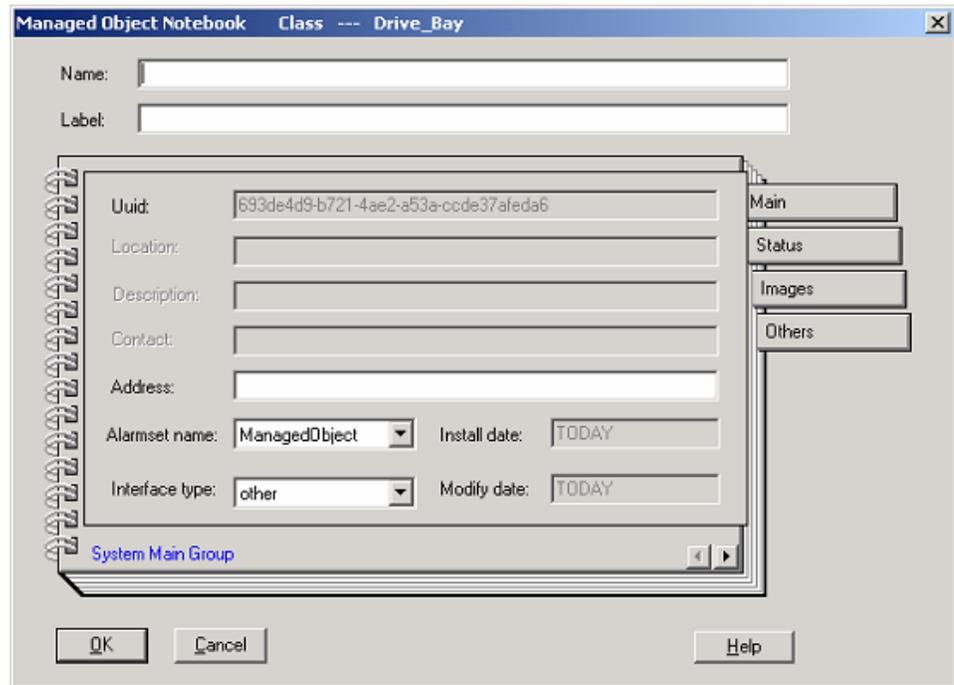
Owned Assets (and relationships) are created and linked to discovered assets through the Unicenter NSM Management Command Center (MCC) or through the WorldView 2D Map with the “Read Change Impact Analyzer” command (which uses the PDM\_NSMIPM.EXE command).

## CORA Parameters for NSM

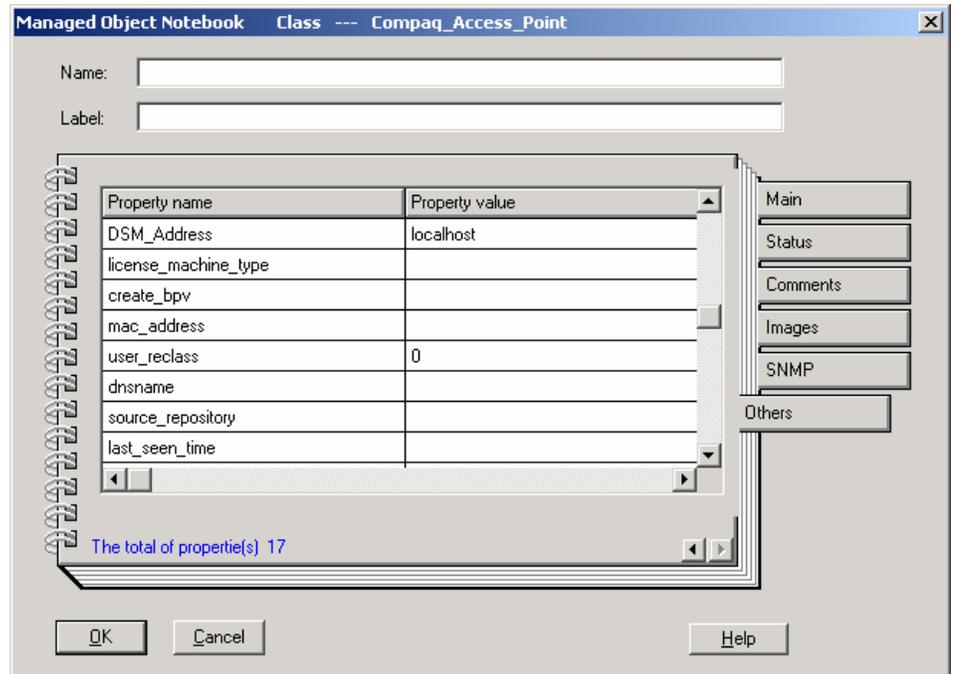
NSM allows the manual creation of new Discovered objects using the 2D Map in Design Mode. The choice of the NSM Class is done upfront from the NSM Class list:



Asset details can then filled in. Note that the “Name” field is mandatory.



The other CORA fields are displayed by clicking the Others tab:



## CORA and Unicenter Service Desk\CA CMDB

In Unicenter Service Desk, “owned” assets are registered through CORA according to the following rules:

- Owned assets are **shared** with Unicenter Asset Portfolio Management
- Owned assets (Hardware, Computer) can be manually created using the Web Interface
- Owned assets can be created and linked to “discovered” computers using the Discover Assets page on the Unicenter Service Desk Web Interface

Registered Asset List - Unicenter Service Desk - Microsoft Internet Explorer

Unicenter® Service Desk

Incident

Logged in as: ServiceDesk (Log Out) (Close Window)

File View Window Help

Registered Asset List

1-9 of 9

Asset Label	Serial Number	Asset Tag	Host Name	DNS Name	MAC Address
		tagF	hostH		
	snD		hostE		
pickMeA	snA	tagA	hostA	dnsA	macA
pickMeB	snB	tagB	hostB		
pickMeC	snC	tagC	hostC	dnsC	macC
pickMeC	snC	tagC	hostD	dnsD	macD
pickMeG	snG				macG
pickMeI		tagI	hostI	dnsI	

1-9 of 9

**Note:** The Discovered Asset page searches both ca\_discovered\_hardware (DSM) and tng\_managedobjects (NSM/CCS) tables

- Owned assets can be created and linked using either the PDM\_DISIMP.EXE or PDM\_NSMIMP.EXE bulk import utilities. PDM\_DISIMP.EXE uses the **ca\_discovered\_hardware (DSM)** table as its source while PDM\_NSMIMP.EXE uses the **tng\_managedobjects (NSM/CCS)** table.
- Discovered assets for Unicenter NSM/CCS (**tng\_managedobjects**) are created and linked to Owned assets through the Unicenter Service Desk Web Interface using CIA Export CI
- Owned Assets are created using PDM\_LOAD.EXE.

## Migrating from an Earlier USD Release

Although the migration process will register existing assets using CORA, some considerations had to be made to handle potential duplicates. Prior to r11 there were very few restrictions regarding asset creation. As a result, a pre-migrated system could contain many assets that CORA would consider duplicates.

To counter this, when CORA determines that two or more assets are duplicates, one of them is registered as-is. The remaining duplicate assets are then given the same `asset_source_uuid` as the first and saved to the `ca_owned_resource` table – but with the following additions:

- Each extra asset is given a DNS Name value. Prior to r11, Unicenter Service Desk assets did not have a DNS Name field, so there is nothing in the migrated data to overwrite. The value of the DNS Name is, `"asset_source_uuid = U'asset_source_uuid"`, where `asset_source_uuid` is the value shared by this set of duplicates.
- The extra assets will also have `"DUP[old_id]"` appended to their Host Name value, where `old_id` is the pre-migration id value of the asset.

Note that the modified DNS Name and Host Names will only exist in the `ca_owned_resource` table. The `ca_owned_resource` values are those displayed in the Unicenter Service Desk user interfaces. They are not registered with CORA; instead, all of the duplicates share the same `asset_source_uuid`.

For example, if a pre-migrated system has three assets with the same Host Name, "QATestBox", after migration one of these records will still have the original values, and will be registered as such. The DNS names for the other two assets will be renamed to something similar to the following:

```
"asset_source_uuid=U'2AB66EE1443BF14293007A11B2E6A5F6"
```

In addition, their Host Names will be modified similar to the following:

```
"QATestBoxDUP[400053]"
```

```
"QATestBoxDUP[400075]"
```

After migration the administrator can sort through the duplicate assets and modify as needed. If one of the duplicates is modified, it will, of course, be re-registered and unlinked from the others in that set of duplicates.

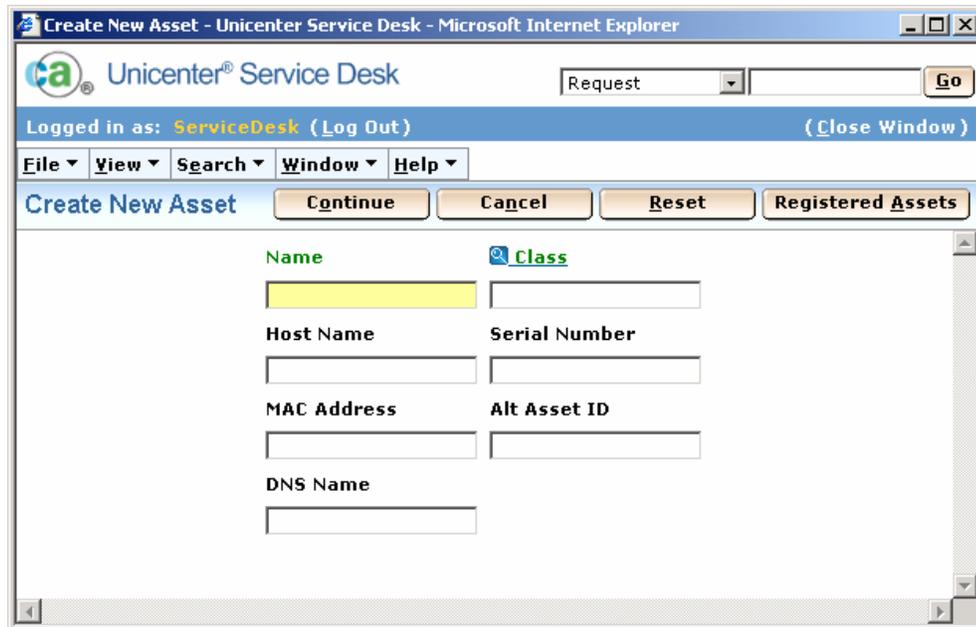
## CORA Parameters for USD

USD assets are created through the New Asset Form that allows the input of the asset class and the following registration properties:

- Name

- Host Name
- Serial Number
- MAC Address
- Alt Asset ID
- DNS Name

Only Asset name and class are required fields:



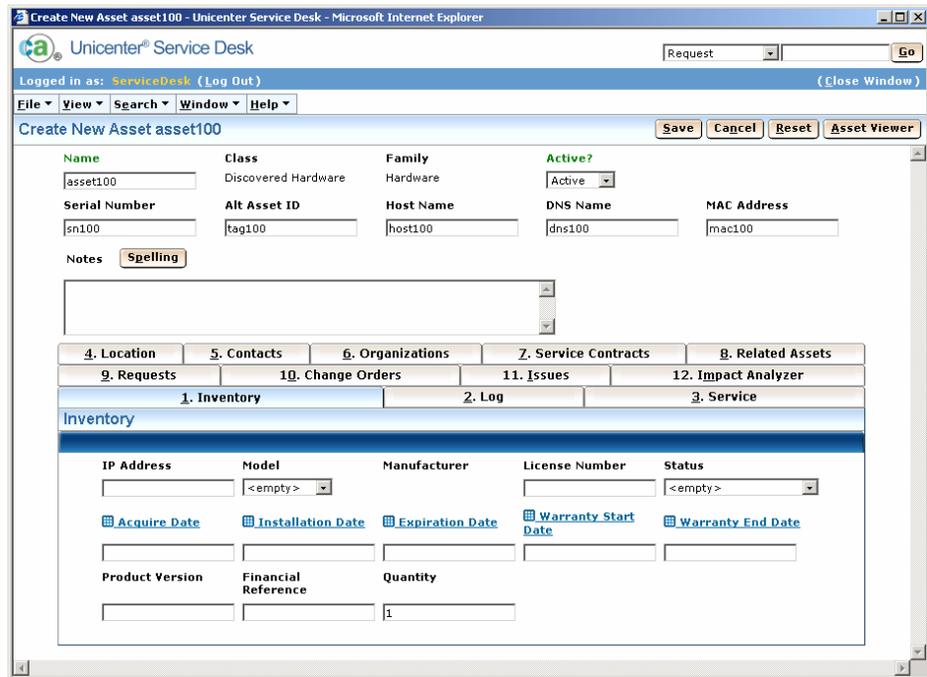
The screenshot shows a web browser window titled "Create New Asset - Unicenter Service Desk - Microsoft Internet Explorer". The page header includes the Unicenter Service Desk logo and a search bar. Below the header, there is a navigation bar with "File", "View", "Search", "Window", and "Help" menus. The main content area is titled "Create New Asset" and contains several buttons: "Continue", "Cancel", "Reset", and "Registered Assets". The form fields are arranged in two columns:

<b>Name</b>	<b>Class</b>
<input type="text"/>	<input type="text"/>
<b>Host Name</b>	<b>Serial Number</b>
<input type="text"/>	<input type="text"/>
<b>MAC Address</b>	<b>Alt Asset ID</b>
<input type="text"/>	<input type="text"/>
<b>DNS Name</b>	
<input type="text"/>	

Once this information is provided and the Continue button is clicked, CORA executes a query to determine if a matching asset already exists.

If a match is found, Unicenter Service Desk issues a message stating that an existing asset was found. The existing asset is then displayed in Update mode. If no matching asset is found, Unicenter Service Desk will display a full Create New Asset form.

**Note:** At this point in the process nothing has been saved.



Once the asset definition is saved, CORA is queried again before the data is written to the database. If the registration properties conflict with an existing asset, the save is aborted and the following message is issued:

“Your changes to Registration parameters match an existing Asset”

If there is no conflict, the new asset is saved and registered.

**Note:** Refer to the [CORA Matrix table](#) provided earlier in this guide for more information on how duplicates are resolved – and under which circumstances CORA identifies a new asset versus a duplicate asset.

## USD Asset Classes

USD uses the concept of “Family” to identify the highest level of definition of a CI. Each Family can have one or more Classes to allow a granular categorization of the CIs.

Each Family has an Extension Table that defines the attributes visible in the CI Detail page.

The CA CMDB Content includes 50+ Families and 140+ Classes that are stored in the mdb and shared with Service Desk and UAPM.

Unicenter® Service Desk / CA CMDB

Logged in as: CMDBAdmin (Log Out)      Incident: [dropdown]      Go

Scoreboard Updated: 09/06/2006 04:10 pm

Service Desk/CA CMDB      Knowledge      Administration

File View Reports Window Help

Configuration Item Family List      Search      Show Filter      Clear Filter      Create New

1-25 of 50 >> List All

Family	Description	Status
Cluster	Multiple servers linked together to handle variable workloads or to provide continued operation if one or more devices fail.	Active
Cluster_Resource	A member of a cluster resource group.	Active
Cluster_Resource_Group	A group of devices in a cluster.	Active
Computer	Computer family	Active
Contact	A person or role active in the IT infrastructure.	Active
Contract	A legally binding business document signed between two parties.	Active
Document	Printed or electronically stored text which is human readable.	Active
Facilities_Air_Conditioning	Heating, ventilation, humidity control, air conditioning or general environment management systems.	Active
Facilities_Fire_Control	Equipment for fire suppression	Active
Facilities_Other	Miscellaneous facilities equipment or supplies.	Active
Facilities_Uninterruptible_Power_Supply	Uninterruptible Power Supplies, and other power conditioning and regulation systems	Active
Hardware	Hardware Device family	Active
Hardware_Logical_Partition	Logical Partitions (LPAR) are a mainframe architecture that segments a single system into several independent logical systems.	Active
Hardware_Mainframe	Large central computing devices, traditionally manufactured by IBM and running zOS, OS390 etc.	Active
Hardware_Monitor	Computer, video, and surveillance displays. Includes CRT's, LCD's, and plasma monitors.	Active
Hardware_Other	Miscellaneous IT hardware.	Active

Copyright © 2005 CA. All rights reserved.

Class is required to register a CI and the referenced Family determines the attributes of the CI Details page.

Unicenter® Service Desk / CA CMDB

Logged in as: ServiceDesk (Log Out)      CI by Name: [dropdown]      (Close Window)

File View Search Reports Window Help

boven01-topgun Configuration Item Detail      Edit      Asset Viewer      CMDB Visualiz

Name	Class	Family	Active?
boven01-topgun	Server	Hardware.Server	Active

Serial Number	Alt CI ID	Host Name	DNS Name	MAC Address
56dd501		boven01-topgun		

Notes

3. Inventory		4. Log		5. Service	
6. Location	7. Contacts	8. Organizations	9. Service Contracts	10. Incidents	
11. Problems	12. Requests	13. Change Orders	14. Issues	15. Impact Analyzer	
1. Attributes			2. CMDB Relationships		

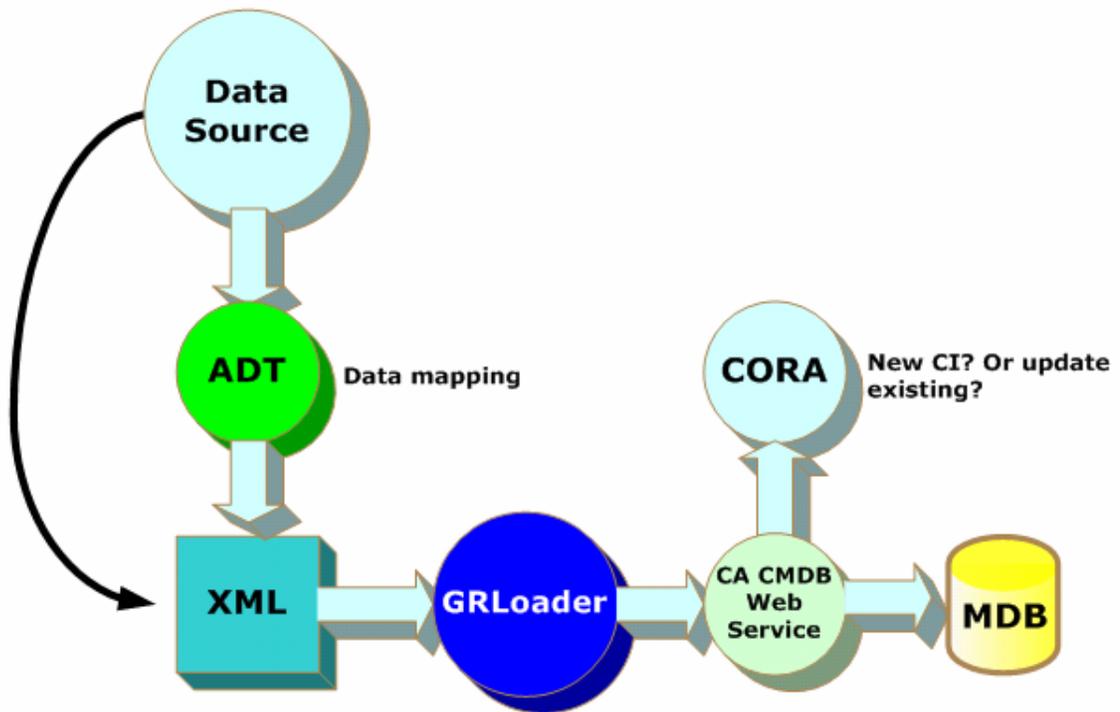
Attributes

Memory Installed	Memory Capacity	Disk Capacity	Processor Type	
Processor Speed	Disk Type	CD Rom Type	Network Card	Monitor Model
Printer	Technology	Processor Capacity	Number of Processors Installed	Processor Cache
Number of Memory Slots	Number of Memory Slots Used	Type of Network Connection	Number of Network Cards	Number of Network Ports

## Universal Federation Adapters and the CA CMDB

The Universal Federation Adapters are a set of utilities provided with the CA CMDB to enable "federation" of data from multiple sources. Through the Universal Federation Adapters, owned resources (i.e., "CI") can be loaded into the MDB and registered in the ca\_owned\_resource table. The subschema\_id will be "1" (like USD and UAPM).

The following diagram describes the flow of data from the source to the MDB:



GRLoader is a CA CMDB executable that allows the creation or update of a CI and its relationships in the MDB through CORA. It is invoked through the CA CMDB (USD) Web Services UI. Input to the GRLoader is done through an XML file which uses a CA proprietary format.

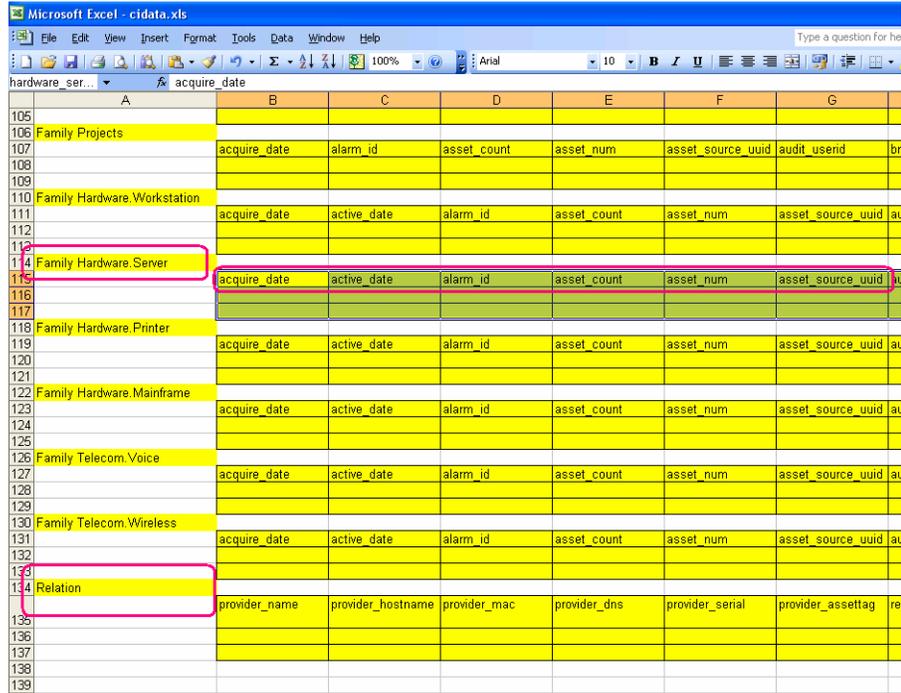
Applications, such as Cohesions (CENDURA), provide a mechanism to create the XML file in the format needed therefore they can execute GRLoader directly. Otherwise ADT (shipped with the CA CDMB) can be used to transform the data into the desired XML file.

**Note:** CA CMDB provides ADT interfaces to create the XML File from any source.

The following can be used as input for the ADT program:

■ **CA CMDB supplied spreadsheet (Cidata.xls)**

This file, which can be found in the \Program Files\CA\Service Desk\cldb\data\federationAdapters folder, lists all out of the box CMDB families, along with their attributes. It also includes a section to define relationships:

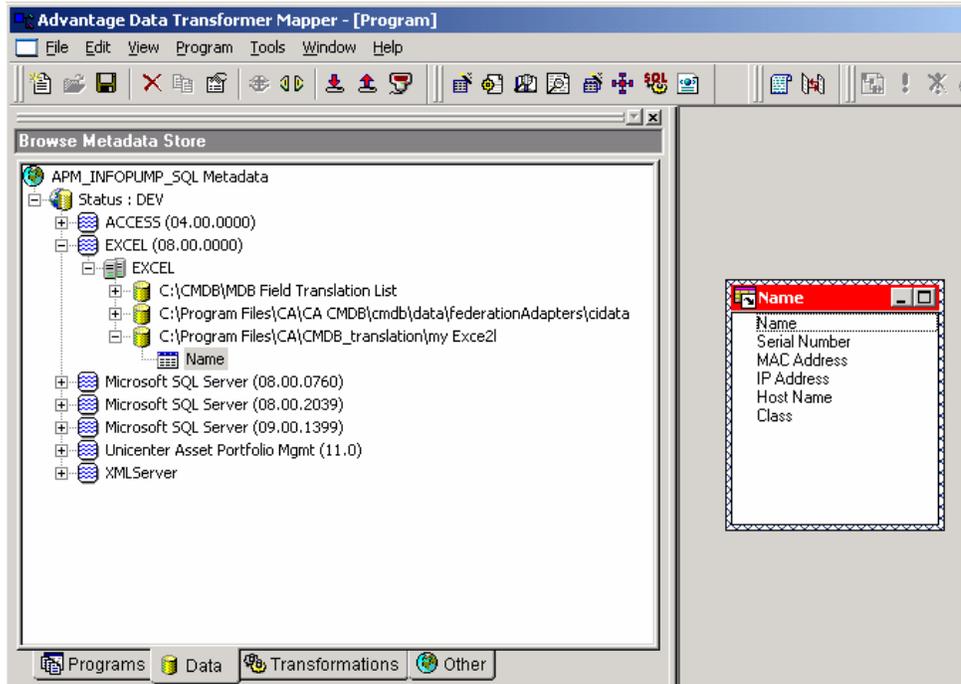


■ **Custom Spreadsheet.**

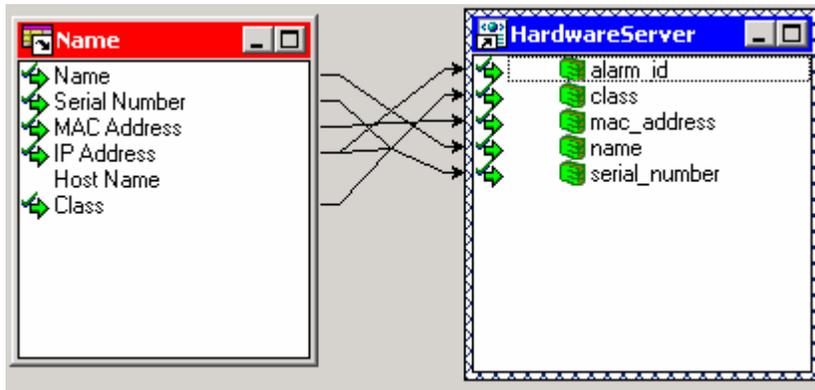
In a situation where it is not possible to populate the supplied spreadsheet, you can use any spreadsheet, provided you perform some additional ADT steps to instruct the ADT Mapper on how to read it. For example, consider the following simple file:

B	C	D	E	F
Name	Serial Number	MAC Address	IP Address	Host Name

This file can be seen in ADT as the "Source"....



And mapped to the appropriate output to generate the required XML file:



- **Use ADT provided interface:**

Using Database views to the source database, ADT provides out of the box scripts for Microsoft SMS and CA UAM 11.0

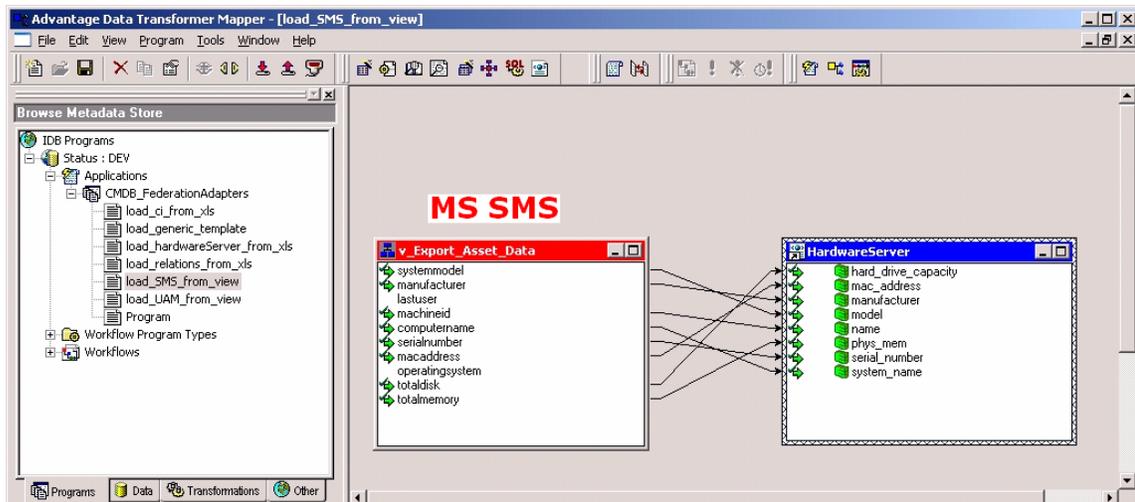
- **Create our own source:**

For example you can easily create ADT Programs for NSM and UAM 11.0

## SMS Adapter for CA CMDB

Out-of-the-box CA CMDB installs Microsoft SMS ADT programs to import SMS data as CI. Using as source a Database View into the SMS database, you can create an output XML file using the XML output for the CI of the Family Hardware.Server.

The following picture shows the ADT script for SMS:



Notice that this script is creating CI from SMS in the ca\_owned\_resource and not importing SMS data as discovered data in the MDB (which you can do with the SMS importer available using the UAPM API).

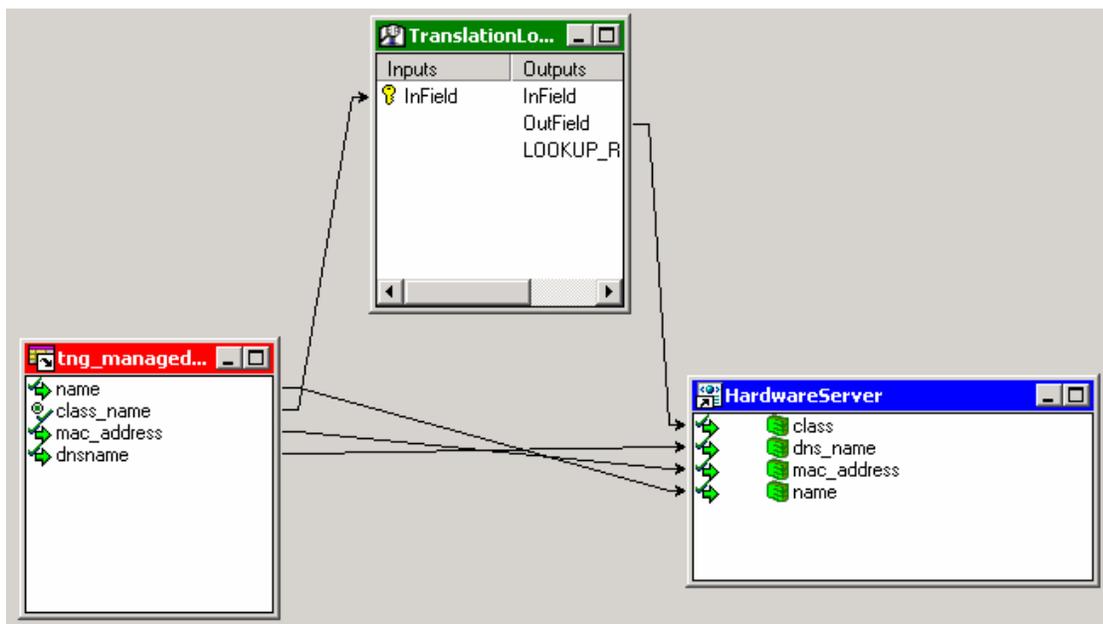
## NSM Adapter for CA CMDB

When CA CMDB and NSM are implemented together, the CA CMDB GRLoader can be used to reconcile NSM discovered objects with CA CMDB CIs.

CA CMDB Classes through the CMDB Families determine the CA CMDB Attribute page. With CA CMDB R11.0 it is extremely important to have the CI created with the correct Class because it cannot be changed afterwards.

We can use ADT to look into NSM tables and generate an XML file to be used by the GRLoader. ADT Program for NSM provides a view into the NSM discovered values and a Class transformation list that maps and transforms the NSM Classes into the desired CA-CMDB Classes.

In this example we show how to map data from tng\_managedobject to the provided CMDB Hardware.Server Family XML output, transforming the NSM Class into the desired CA CMDB Class.



Name, MAC Address and DNS Name are the basic values used by CORA to register and reconcile the assets.

Notice how the NSM Class field goes to the InField of the TranslationLookup to be transformed in the Outfield before going to the CMDB class field. The translation list is maintained in a spreadsheet used by ADT while processing the script:

Microsoft Excel - HPUnixNSM.xls

File Edit View Insert Format Tools Data Window Help

Verdana 10

	A	B
1	NSM_Class	CMDB_Class
2	HPUnix	HP UX

## Useful Links and References

The full schema for the MDB is viewable through the Implementation Best Practices page (formerly the “r11 Implementation CD”) which is available on SupportConnect at the following link:

<http://supportconnectw.ca.com/public/impcd/r11/MDBMain/schema/viewer/index.htm>