



Structuring Software Models **in HP Asset Manager 9.30**

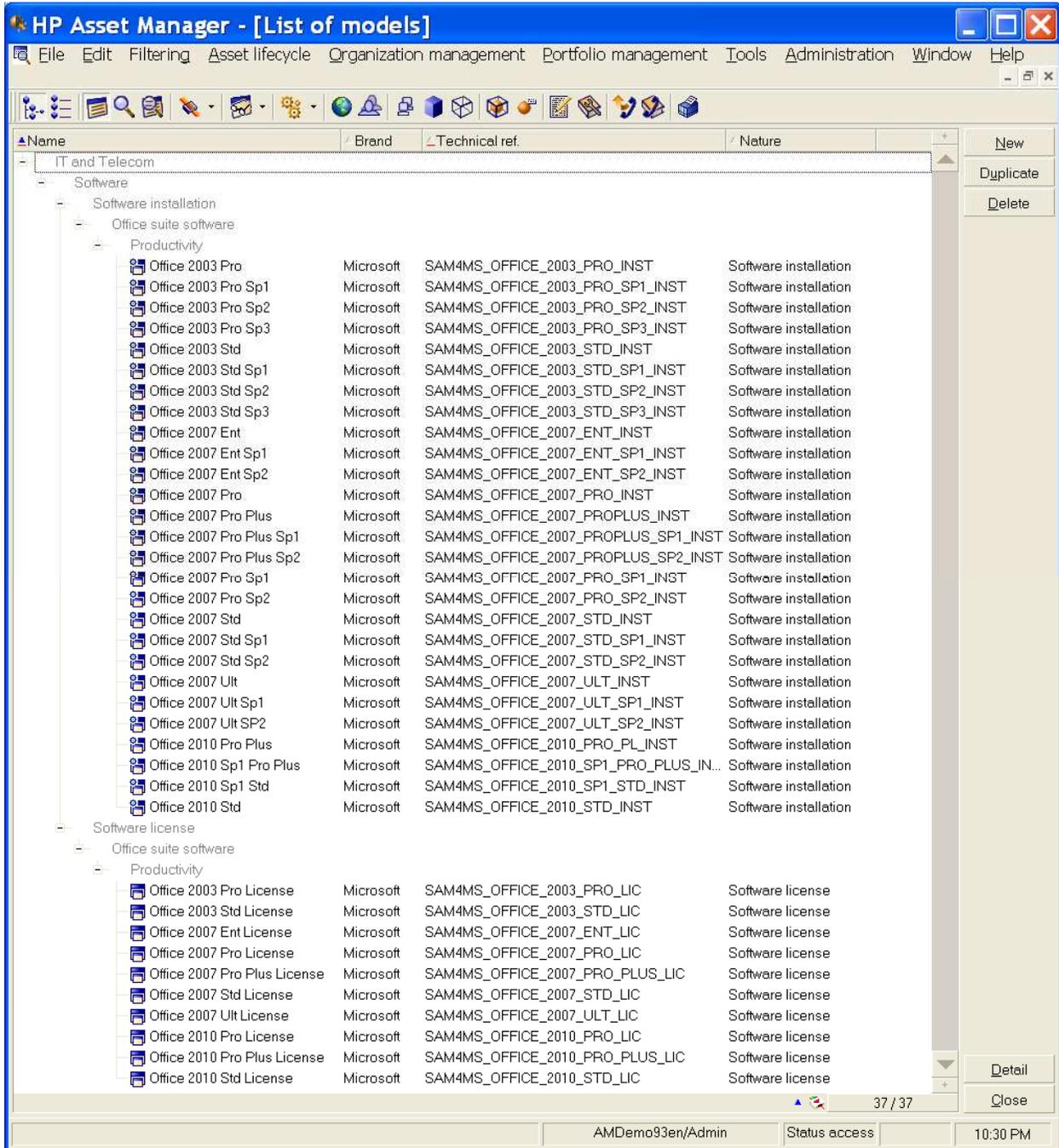
How it is done can greatly help or hinder your Software Asset Management efforts

One of the most important tables in Asset Manager is Models since it provides the underlying structure and organization for how you manage all objects stored in the AM database. Portfolio Items, Assets, Contracts, Work Orders, etc... everything we track must have a Model. Because of this the way you structure the data on the Models table has a tremendous effect on how you use Asset Manager. Doing it well is one of the keys to easily being able to work easily with the system in a variety of ways. This is not always immediately apparent however and it is an aspect of Asset Manager ignored in many implementations or optimized for one function or audience at the cost of others. The way Software Models are typically set up causes Software Asset Management a number of difficulties. Put another way, a number of the difficulties associated to SAM can be simplified by taking a new approach to how we structure the Models table with respect to Software.

The typical method commonly used to structure software models has a “software” node with a branch below it for software installations and another for software licenses. Each branch further subdivides by category and finally down to individual software titles. Contracts have their own hierarchy with separate branches for licenses and maintenance contracts. At first glance this basic approach looks good, however there are often cases where you need to relate and reconcile all the various models related to a particular piece of software to one another: contracts, licenses, installations, and work orders. The common approach has several major limitations built into it when it comes to working with these various pieces together, particularly with respect to building counters. The main advantage it offers is the easy separation of models by type and this can be easily done by filtering on the “Model.Nature” without constraining the model structure. The methodology developed by Golden Ratio addresses these issues and allows the structure of the Models table to be leveraged in a way which works with the Software Assets Module but allows for greater flexibility and less maintenance.

Visit www.GoldenRatioInc.com or contact me at Russ.Parker@GoldenRatioInc.com to find out more

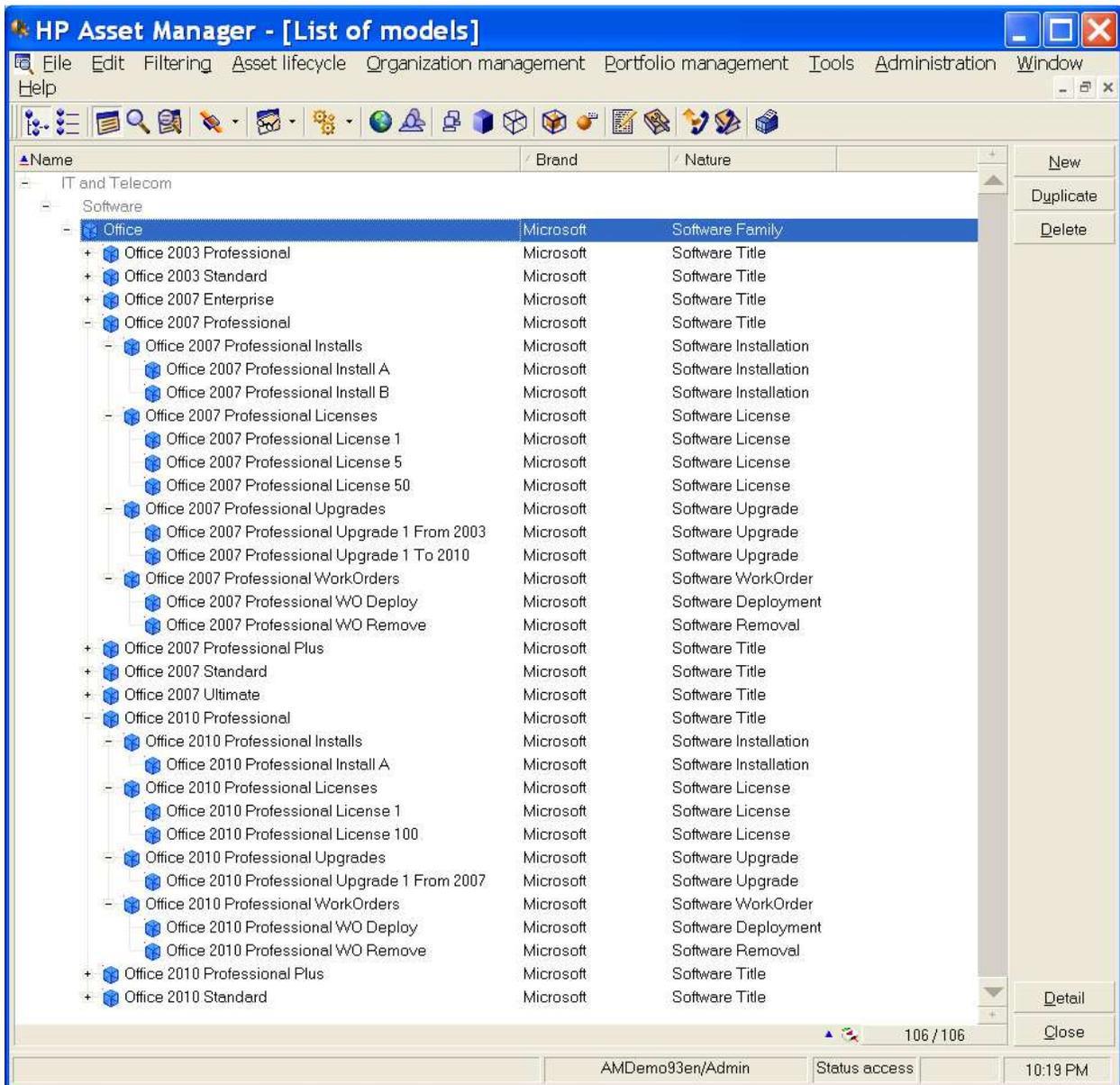
Here is an example of the structure commonly used in many Asset Manager Implementations.



The list above has been filtered to just show Microsoft Office, but all software classed under “Productivity” would be typically be at the same level making the list even longer. Even filtered the data reconciliation problem this presents can be fairly easily seen: there is no relationship between the license models for a given piece of software and the installation models for it. This results in the Queries and Counters created for this software needing to either use hardcoded values which make them static or or N:N relationship tables which need to be maintained.

Visit www.GoldenRatioInc.com or contact me at Russ.Parker@GoldenRatioInc.com to find out more

The methodology developed by Golden Ratio comes at the question of how to structure Models from a different direction. We wanted a structure which would both better fit how our clients naturally wanted to organize the data and which also minimized the amount of maintenance work they needed to do. We found that our clients rarely wanted to look at installations or licenses for many products at once. Instead what they most often needed to do was look at both of them at the same time for a particular piece of software or a related set of software. Based on that we came up with the concept of the Software Title and Software Family groupings and used these as the way to initially organize the our Models. All models relevant to a given Software Title are beneath it where they can be easily found and then related Software Titles are grouped together as a Software Family. The structure looks like this:



Visit www.GoldenRatioInc.com or contact me at Russ.Parker@GoldenRatioInc.com to find out more

In addition to making it easier to find everything related to a particular Software Title by having it all be in one place, this structure also provides the “missing link” between different object types we were lacking before. Instead of having to manually maintain the relationship between which licenses and installations tie to one another on each separate Counter the relationship is implicit in the Software Title structure. Licenses and Installations both have a common Parent, or as we ended up refining it, a common grandparent.

Below each Software Title model are nodes for the various types of objects relevant to that title. Contracts, licenses, upgrades, installations, and work orders each get their own tree, but they can all be tied together by their common Software Title. The beauty of this is that if you add a new model under one of those trees there is no maintenance needed. The query for Rights on a Counter gets tied to the Software Title and is going to count all Asset tied to a Model in the “Licenses” tree of that counter (with the correct License Type of course!). Just adding the new Model in the appropriate place is all the maintenance needed. No Queries need to be updated and no Wizards need to be run to add that Model to a Counter or Counters. Combined with some of the structural changes we have made to how Counters themselves function the operational effectiveness of using Asset Manager to support the SAM function is greatly enhanced.

Another difference between the standard Asset Manager SAM functionality and how Golden Ratio implements it has to do with Upgrade Counters. In our experience they have two major flaws: added calculation time and visibility limited to the counter results. Golden Ratio’s methodology for Upgrades does away with the Upgrade Counters completely and instead views an upgrade as a two part object: a “credit” record and a “debit” record. If you upgrade a license of “Microsoft Office 2007 Professional” to “Microsoft Office 2010 Professional” we create a debit license asset with a negative number of rights for the old version and a credit license asset with positive number of rights for the new version. In the example above this is shown with the models “Microsoft Office 2007 Professional Upgrade 1 To 2010” and “Microsoft Office 2010 Professional Upgrade 1 From 2007”. Counters which take upgrades into account sum up any assets tied to models in either the licenses or upgrades trees and thereby include the upgraded rights as part of the counter standard calculation. Totals for the old version are reduced and those for the new version increased automatically when calculated. This approach does have the downside of creating two asset records instead of one, but that is easily automatable and has the benefit of clearly showing the effect of the upgrade if viewed outside the context of the counters. We also tend to advocate using fewer multiple right licenses rather than individual licenses which results in a fairly small number of “extra” Asset records.

Golden Ratio offers both proven Software Asset Management methodologies and customizations for HP Asset Manager which increase its functionality and usability. The thoughts and ideas presented here are examples of those and we hope they prove helpful. If you have any questions about out them please feel free to contact us and we will be happy to provide more information about how we can help you and your organization.

Visit www.GoldenRatioInc.com or contact me at Russ.Parker@GoldenRatioInc.com to find out more