Endpoint Management and Mobility

OwnerNS guid now affects resource replication

Reasons for the change

Reason for the change is constant replication up of the same computer resources from. The same computer resources appear on several children NS-s because of switching client machines from one NS to another. Such switching won't delete a computer resource from previous owner NS, as a result the computer resource appears on both NS-s, previous and new one. This results in multiple "sources of truth" and degradation of the performance of hierarchy replication.

What was introduced

The following areas were changed:
1. Resource replication filtering added, so that "not owned" resources won't be replicated. In other words, the only NS allowed to replicate a resource is that which owns the resource (resource.OwnerNS == vThisNS).

2. Extended already existing logic of **owner conflict resolution**. Now in addition to resolving "managed" flag of a resource, we also ensure that only one resource owner assigned on all tiers of the hierarchy.

3. **Owner NS** guid now included into **hash** calculation for a resource.

Resources are filtered at the begging of replication (during the 'CreateManifest' stage). If some resources are filtered out then in the Altiris Log will be message "*ReplicationFilter JobID[{0}]: {1} resources were filtered out, because they are owned by another NS.*". Also in the replication report such resource will have StausCode==0 and StatusText=="*Excluded from replication because is owned by another NS*".

The previous logic of **owner conflict resolution**, despite to its name, actually didn't change owner of a resource - the logic has made the "Managed" property of the resource to be consistent across all tears of the hierarchy. The logic extended so that if destination NS (mostly the parent NS) finds out that resource (currently being imported in context of hierarchy replication) has different owner NS guid than already existing resource (destination's one) then special "Owner Conflict" hierarchy event is raised. The event is replicated to opposite direction (read to children), and contains resolved "Managed" flag as well as new owner NS guid. The way how these flag and owner NS guid are resolved hasn't been changed (see details in the 'Owner Conflict Resolution Details' section below), but what was changed is how listeners of the event react:

- **Previously** virtual method Item.**ResolveOwnerConflict**() has been called for the resource item. Its default behavior hasn't changed, and for a resource item is: *clear the "Managed" flag to 0*. The intention of such behavior was the following: as far as we can't decide who owns the resource (conflict detected) we just clear the "Managed" flag on all tiers, and later real owner of the resource will reset the flag back (inventory NSE from client machine does this).

- **Now**, in case the event contains a new resolved owner NS guid, the new virtual method Item.**ResolveNewOwnerConflict** (Guid newOwnerGuid) will be called. Default behavior of the method for a resource item is: *if new owner differs from current one then clear the "Managed" flag to 0 and setup the new owner NS guid*. If the owner conflict resolution event doesn't contain a new owner guid (read as : "we can't resolve the owner") then the "old" Item.ResolveOwnerConflict() will be called (as it was before the change).

How Solutions could be affected

Commonly, solutions shouldn’t be affected because they generally don’t modify or directly rely on specific value of owner NS guid. The value is set, persisted and maintained by SMP core behind the scene. So the only exceptions are **resource derived classes** which have some owner guid dependent logic. Also those classes who override the Item.ResolveOwnerConflict() method should be aware that now exists additional Item.ResolveNewOwnerConflict(Guid newOwnerGuid) method which is called during owner conflict resolution as well.

Effect on n-tier hierarchy

The new resource filtering mechanism does not allow resources to be replicated more than one level in hierarchies consisting of more than two levels.

For example, a resource managed by an SMP at the lowest level in a three tier hierarchy cannot be replicated to the SMP at the top level of the hierarchy, because the second tier SMP will filter out resources that it does not manage and, consequently, will not replicate such resources to the top level SMP.

While Symantec has always strongly recommended that customers not use hierarchies consisting of more than two levels, support for hierarchies of more than two level will be officially eliminated in the Altiris ITMS 7.5 release as a result of this change.

Changes availability

Changes are available since SMP build # 7.5.1309

Owner Conflict Resolution Details

Following sub-sections below describe how owner conflict resolution has worked on 7.1.SP2 and how it is working now in 7.5. The tables represent replication up of the same resource item from either child NS # 1 (C1) or child NS #2. Notation "x/Cy" represent state of the "Managed" flag (x) and "OwnerNS" guid (y) (for example 1/C2 means that the resource is managed and its owner NS is child # 2). "*" (asterisk) is used to represent the same resource but with different data (for instance name of the resource).

7.1.SP2 behavior

	Before replication (managed/owner)			Replication happened?	Owner Conflict Event	After replication and event (managed/owner)			
	Parent	Children				Parent	Children		
	P	C1	C2			P	C1	C2	
0	x	0/C1↑	x	yes	x	0/C1	0/C1	x	
1	x	1/C1↑	x	yes	x	1/C1	1/C1	x	
2	0/C1	1/C1↑	x	yes	x	1/C1	1/C1	x	
3	1/C1	0/C1↑	x	yes	x	0/C1	0/C1	x	
4	1/C1	1/C1↑	x	no (same hashes)	x	1/C1	1/C1	x	
5	1/C1	1/C1*↑	x	yes	x	1/C1*	1/C1*	x	
6	0/C1	0/C1↑	x	no (same hashes)	x	0/C1	0/C1	x	
7	0/C1	0/C1*↑	x	yes	x	0/C1*	0/C1*	x	
8	0/C1	0/C1	0/C2↑	no (same hashes)	x	0/C1	0/C1	0/C2	
9	0/C1	0/C1	0/C2*↑	yes	x	0/C2*	0/C1	0/C2*	
10	0/C2*	0/C1↑	0/C2*	yes	x	0/C1	0/C1	0/C2*	
11	0/C1	0/C1	1/C2↑	yes	x	1/C2	0/C1	1/C2	
12	1/C2	0/C1↑	1/C2	yes	x	1/C2	0/C1	1/C2	
12	1/C2	0/C1↑*	1/C2	yes	x	1/C2*	0/C1*	1/C2	
13	1/C2*	0/C1*	1/C2↑	yes	x	1/C2	0/C1*	1/C2	
14	1/C1	1/C1	1/C2↑	no (same hashes)	x	1/C1	1/C1	1/C2	
15	1/C1	1/C1	1/C2↑*	yes	↓	0/C2*	0/C1	0/C2*	- for hierarchy only
16	1/C1	1/C1	1/C2↑*	yes	x	1/C2*	1/C1	1/C2*	- for standalone replication only
16	1/C2*	1/C1↑	1/C2*	yes	x	1/C1	1/C1	1/C2*	- for standalone replication only
17	0/C1	1/C1↑	1/C2	yes	x	1/C1	1/C1	1/C2	
18	0/C1	1/C1	1/C2↑	yes	x	1/C2	1/C1	1/C2	
19	1/C2	0/C2↑	1/C2	yes	x	1/C2	0/C2	1/C2	

↑	replication up
↓	replication down
red	item save
🟦	change of owner
*	resource with different data (and so different hash)
🟧	potential constant replication alternately from C1 and C2 (diff hashes)
🟥	constant replication alternately from C1 and C2 (diff hashes in managed flag)
🟨	actual replication up

7.5 behavior

	Before replication (managed/owner)			Replication happened?	Owner Conflict Event	After replication and event (managed/owner)			
	Parent	Children				Parent	Children		
	P	C1	C2			P	C1	C2	
0	x	0/C1↑	x	yes	x	0/C1	0/C1	x	
1	x	1/C1↑	x	yes	x	1/C1	1/C1	x	
2	0/C1	1/C1↑	x	yes	x	1/C1	1/C1	x	
3	1/C1	0/C1↑	x	yes	x	0/C1	0/C1	x	
4	1/C1	1/C1↑	x	no (same hashes)	x	1/C1	1/C1	x	
5	1/C1	1/C1*↑	x	yes	x	1/C1*	1/C1*	x	
6	0/C1	0/C1↑	x	no (same hashes)	x	0/C1	0/C1	x	
7	0/C1	0/C1*↑	x	yes	x	0/C1*	0/C1*	x	
8	0/C1	0/C1	0/C2↑	yes	C2↓	0/C2	0/C2	0/C2	
9	0/C1	0/C1	0/C2*↑	yes	C2↓	0/C2*	0/C2	0/C2*	
10	0/C2*	0/C1↑	0/C2*	yes	C1↓	0/C1	0/C1	0/C1*	
11	0/C1	0/C1	1/C2↑	yes	C2↓	1/C2	0/C2	1/C2	
12	1/C2	0/C1↑	1/C2	yes	C2↓	1/C2	0/C2	1/C2	
12	1/C2	0/C1↑*	1/C2	yes	C2↓	1/C2*	0/C2*	1/C2	
13	1/C2*	0/C1*	1/C2↑	yes	x	1/C2	0/C1*	1/C2	
14	1/C1	1/C1	1/C2↑	yes	↓	0/C2	0/C1	0/C2	- for hierarchy only
15	1/C1	1/C1	1/C2↑*	yes	↓	0/C2*	0/C1	0/C2*	- for hierarchy only
16	1/C1	1/C1	1/C2↑*	yes	x	1/C2*	1/C1	1/C2*	- for standalone replication only
16	1/C2*	1/C1↑	1/C2*	yes	x	1/C1	1/C1	1/C2*	- for standalone replication only
17	0/C1	1/C1↑	1/C2	yes	x	1/C1	1/C1	1/C2	
18	0/C1	1/C1	1/C2↑	yes	C2↓	1/C2	0/C2	1/C2	
19	1/C2	0/C2↑	1/C2	no (owner is C2)	x	1/C2	0/C2	1/C2	

↑	replication up
↓	replication down
red	item save
🟦	change of owner
*	resource with different data (and so different hash)
🟨	actual replication up
□	area where behavior has been changed