WHITE PAPER

FlexNet Publisher Trusted Storage Unique Identifier – Usage Guidelines

FLEXERA SOFTWARE

FlexNet Publisher Trusted Storage Unique Identifier – Usage Guidelines

Executive Summary

This White Paper presents an evolution of the various methods that FlexNet Publisher Trusted Storage has utilized to identify a client system to its server with an emphasis on best practice recommendations of using the Unique Machine Number (UMN).

Requirement for Client System Identification

Various methods of identifying a client system have been introduced for different purposes at different stages in the evolution of FlexNet Publisher Trusted Storage. All of these methods have been retained for backward compatibility. In addition, some legacy identifying methods may be necessary for some operational modes that are described below.

Machine ID

The Machine ID (MID) is generated from machine binding items as configured by the user. Binding items are individual physical characteristics of the machine that are used individually or in combination to create a signature of the system hardware. It is for this reason that Machine ID is also known as the "Binding Hash."

When implementing Activation in FlexNet Publisher, license rights are held in a secure "vault" known as Trusted Storage (TS). The Trusted Storage is unique to each machine and this uniqueness is maintained by the publisher who may wish to apply different licensing strategies to individual features or to complete software packages. To allow this granularity, Trusted Storage can be divided into sections to match the features or applications of those entitlements that are being managed.

The MID is generated in the client at the request of the server. In the case where multiple trusted storage sections exist on the machine's hard drive, there will be more than one MID on that system.

The publisher can use these multiple sections of TS and apply stronger or lighter binding to that section, depending on the binding configuration applied to that section.

FlexNet Publisher's flexible binding features mean that some binding change can occur without the fulfillment records stored on the machine losing trust. However, since the Machine Identifier includes all binding items, it will change if any item changes. If more than half of the weighted binding items simultaneous change, this is known as a binding break.

This allows a publisher to trade off strong binding, with high protection at the risk of a support call if some element of the binding changes, against weak binding, where there is basic protection against fraud but much less chance of the end user being inconvenienced if a physical characteristic of the system is changed. By default in FlexNet Publisher strong binding is enabled. Weak binding can be configured by explicitly disabling binding items in the trusted configuration.

MIDs have their shortcomings. They are subject to change since they are based on multiple binding items that are user selectable. Furthermore, some binding items are more likely to change than others (e.g., IP address).

Since this value can change when Trusted Storage is repaired, the server must keep an audit trail of previous MIDs to track the client. Unfortunately, when system changes occur which are within security tolerance, they do not necessarily trigger repairs. The result is the activation server may not have a record of the most current MID on the client machine. Therefore, it is inadvisable to base a reinstall business decision solely on MID. This susceptibility to change ("brittleness") may be reduced by removing the HID_SYSTEM and HID_INTERNET binding items from the trusted configuration.

Previously, the Machine ID was recommended as the best means to authenticate a system but an improved approach, based upon lessons learned, has superseded this method. This method is known as UMN and is described below.

Unique Machine Number

The Unique Machine Number (UMN) was introduced in FlexNet Publisher v11.3 in an attempt to overcome the shortcomings of the techniques described above.

UMN is designed to be unique and constant. Not dependent on being a stored value, it is intended that it can be faithfully regenerated on a given piece of hardware even after a complete system wipe and be used to positively identify itself as the client system that was originally issued license rights.

UMN is now the best practice method of establishing and verifying client identity and is described in greater detail in the next section.

TSSN

The TSSN is a randomly-generated, unique number stored in a trusted storage file when it is first created. However, this is not linked to any physical attribute of a machine's hardware.

The TSSN can be used to identify a particular instance of trusted storage (i.e., the file but not the machine it is on).

UMN in Detail

Introduction

UMN was introduced in FlexNet v11.3 to overcome the consistency problems with MID described above.

The intent was to derive a means to uniquely identify an end user's computer system for subsequent licensing transactions. UMN identifies the client system to which license rights apply.

UMN is used when initially granting activation rights, in returning license rights, and when repairing an activation (which requires confirmation of the client being the same one that originally received the allowed activation).

In an ideal world, all systems would have a globally-unique serial number burned into the main circuit board. Indeed, some of the high-end workstations from HP and Sun Microsystems provide this. However, for all other systems the next best identifying technique is UMN.

Even with an unchanging, unique identifier, a software producer's business logic would still have to handle the case where the main hardware failed and had to be replaced, causing the unique identifier to change.

UMN, therefore, is derived according to the type of client system or "platform." Mindful of the possibility that it may not be possible to generate a UMN in certain circumstances on certain machines, a second mechanism was defined for each platform, to be used as a backup approach.

This is the reason why there is a primary UMN1 and a fallback UMN2. UMN1 is the first-choice on each platform. It has been chosen to give the best chance that each machine will have a different UMN, but may not be available on all machines.

UMN2 is the second-choice on each platform. It has been chosen as being available on a very high percentage of machines but with a greater chance of duplication.

Sources of UMN

FlexNet Publisher clients are supported on a wide range of platforms, and so a number of different means of generating UMN values exist depending on the underlying hardware.

Since UMN was introduced in FlexNet Publisher v11.3, experience has been gained which has prompted a change in strategy regarding the source of information used to generate the UMN values. This change is manifested in FlexNet Publisher v11.6.1 and beyond.

UMN2 on Windows systems is now derived from the MAC address of the primary network interface. This is determined by ignoring any class of network interface that is not on a physical bus. This means that removable and virtual network interfaces are not considered, ensuring the reliability of UMN2 on Windows.

Diagram 1 lists UMN1 and UMN2 derivations for each supported platform.

Platform	UMN1	v11.3 – 11.6.0 UMN2	v11.6.1 onward UMN2
Windows	Boot disk serial number.	Boot disk geometry.	Primary Ethernet MAC address.
Mac	MAC Unique System ID. On newer systems, this ID is burned-in to the motherboard, so the number cannot be changed. For older systems, this information is on the disk and can only be overwritten by low-level formatting.	Primary Ethernet MAC address.	Primary Ethernet MAC address.
Linux	Hard disk serial number. Only available if FlexNet PublisherLicensingService is run.	Primary Ethernet MAC address.	Primary Ethernet MAC address.
AIX	Unique hardware serial number. Available on all PCI based AIX hardware.	Primary Ethernet MAC address.	Primary Ethernet MAC address.
HP/UX	Unique hardware security key	Primary Ethernet MAC address.	Primary Ethernet MAC address.
Solaris	Serial number generated during manufacturing and written to the EEPROM. Please note that this value changes on OS reinstallation on Sun or PC hardware but not on SPARC.	Primary Ethernet MAC address.	Primary Ethernet MAC address.

Diagram 1. FlexNet Publisher clients are supported on a wide range of platforms, and so a number of different means of generating UMN values exist depending on the underlying hardware.

Potential Shortcomings

As can be seen, the preferred UMN1 value is a hardware serial number when it's available. In the case of x86-compatible systems (the Windows and Linux platforms), this is not available, and so the next best identifier is the serial number of the hard disk retrieved programmatically.

Unfortunately, for FlexNet Publisher client systems based on Windows, there is not a consistent and reliable means of interrogating the hard disk serial number. It can also be seen that all systems derive the fallback UMN2 value from the system's primary Ethernet MAC address, except Windows systems. This can lead to problems explained in more details in the section *Things that can go wrong with UMN*.

UMN Post v11.6.1

The shortcomings alluded to above (and described in more detail in a later section) prompted a move to derive UMN2 from the Ethernet MAC address of the system's primary network interface. This was not done previously, in part, because network interfaces on Windows PC systems historically were usually add-in cards. The presence or lack of a card, or indeed the possibility of exchanging cards, did not suggest a reliable source of system identifier.

However, today network interface hardware on Windows platforms is typically a feature of the system's main board, especially for notebook computers. It is now a more obvious choice.

Therefore, for FlexNet Publisher v11.6.1 and beyond, UMN2 for Windows platforms has been brought into line with the other clients to use the primary Ethernet MAC address.

Best Practice

The following operations are presented as Best Practice:

Anchoring

On Windows systems, both track 0 and registry anchoring should be enabled (i.e. stick with the default).

Binding

Initial trusted configurations should use the default settings for binding with everything enabled. However, to overcome the brittleness of MID, it is recommended that the system (HID_SYSTEM) and IP address (HID_INTERNET) binding items be excluded.

If Track 0 anchoring is turned off by default, as is the case in FlexNet Operations versions until 11.5, then it is recommended that track 0 is turned back on. If track 0 is not turned on,
then the binding item HID_MSN should also be excluded from binding. This is because HID_MSN is stored in anchors, and could be subject to change in the case of a bona fide system restore.

Use UMN

Use UMN1 and UMN2 to assist in deciding whether to approve a repair transaction.

Recommendations for the use of UMN on Windows platforms and suitable fallback strategies now differ between the pre and post 11.6.1 versions of FlexNet Publisher. Accordingly, these recommendations are presented separately.

Best Practice versions 11.3 to 11.6.0

For activations, repairs, and re-installations, it is recommended that UMN1 be used as the first choice.

Software producers who have previously used Machine ID or TSSN to identify machines should update their licensing database to record UMNs as a machine's identity. The Machine ID is still available post v11.6.1 in all standard XML requests and the TSSN in repair requests. This means they can be used to find out if a machine already exists on the database and then to update the record to include UMN.

Producers who use Machine ID or TSSN as the input to their business logic (e.g., in deciding whether to allow repeated repairs) should now use UMN as the recommended form of client identifier.

Things that can go wrong with UMN pre- v11.6.1

While UMN is the best method to identify a client to its server, no technique is perfect. Several variables can contribute to problems with UMN.

In decreasing order of severity:

1. Non-consistent or random UMN
   – UMN changes when the query is refreshed or sometimes after reboot (common with RAID)
2. Non-unique UMN in some rare situations
3. Brittleness
   – Value changes after a hardware upgrade/repair.
4. Not obtainable
   – UMN not supported by the underlying driver or device.

UMN 1

There is the possibility on Windows platforms using SSD, RAID, or virtual machines (e.g., using VirtualPC or VMWare) that the UMN1 value may be random resulting in multiple re-install requests. For these same reasons, the UMN value can also be non-unique, resulting in giving away entitlements to multiple clients since they all have the same UMN identifier.

In these cases, including the condition where UMN1 is reported as being “not available”, it is advisable to use MID instead.

It is recommended to upgrade to the latest version of the FlexNet Windows Licensing Service (available in the FlexNet Publisher v11.6.1 and above) in order to remove random or non-unique UMN1 values from these types of systems. This version of the service will return the UMN1 value as being “not available.”

The Windows FlexNet Licensing Service is designed to always be backward compatible with earlier versions of FlexNet Publisher (back to v11.3). The licensing service may be installed from the latest FlexNet Publisher kit.

UMN2

In FlexNet Publisher client versions before 11.6.1, the fallback UMN2 value for Windows systems is derived from the geometry of the system's primary hard disk. Note that two hard disk drives with the same model number from the same manufacturer will have the same disk geometry. This approach has been found to be problematic in certain situations where a cluster of client systems are of identical specification (e.g., having been specified and bought en masse for a particular project). In this scenario, all the UMN2 values are identical since the hard drives are identical and have identical geometries. This results in non-unique UMN2 values being presented to the server with each client appearing to be the same.

Also, in RAID or virtual machine configurations, if there were problems obtaining a serial number from the primary drive (as will often be the case), there may also be problems in obtaining the disk geometry.

In these pre- FlexNet Publisher v11.6.1 Windows client scenarios, it is recommended that UMN2 not be used. Instead, the fallback is to use Machine ID with the HID_SYSTEM and HID_INTERNET binding items excluded. An example trusted configuration file that shows how to disable binding items can be found in Appendix A of the Trusted Storage-based Licensing Programming Reference guide.

In summary, UMN1 on Windows systems generally works well, but some operational issues concerning UMN1 have arisen due to the lack of a reliable, hardware-based identifier.

Unfortunately, when difficulties do arise in deriving UMN1, the fact that UMN2 on Windows is not guaranteed to be unique tends to compound the problem, especially in workgroups where identical hardware has been sourced or if RAID or virtual machines are used.

Best Practice for v11.6.1 and Beyond

For activations, repairs, and re-installations, it is suggested that UMN1 be used as the first choice.

The behavior of UMN1 generation on Windows clients has been improved in v11.6.1 with respect to handling SSD, RAID, and Virtual Machine disk configurations.

In cases where FlexNet Publisher versions pre- 11.6.1 may have become confused and produced changing UMN1 values, the enhancements described in this document will now yield a consistent UMN1 value or none at all. This behavior is more predictable and robust and will remove spurious re-install requests that were previously experienced in these particular configurations.

When a UMN1 value is not available, UMN2 should be used instead because it is much improved in v11.6.1 and beyond since it is disconnected from the primary disk configuration details.

Producers who have previously used Machine ID or TSSN to identify machines should update their licensing database to record UMNs as a machine's identify. The Machine ID is still available in all standard XML requests (and the TSSN in repair requests) in v11.6.1 so these requests can be used to find out if a machine already exists on the database and then update the record to include UMN.

Producers who use Machine ID or TSSN as the input to their business logic (e.g., in deciding whether to allow repeated repairs) should now use UMN as the recommended form of client identifier.

Side Effects of Changing the Source of UMN2 on Windows

In the unlikely event that a producer has been using UMN2 on Windows clients (only in the small percentage of cases when UMN1 is not available) and then upgrades to FlexNet Publisher v11.6.1, a re-install will be required since the value of UMN2 will be different due to the change in source used for it generation.

It should be noted that FlexNet Operations has never used UMN2 for Windows activations pre-v11.6.1 so this will only affect LGT users.

Solid State Drive (SSD)

A number of Windows machines have been evaluated for use with FlexNet Publisher that feature an SSD as the primary drive. Those devices whose software drivers present them as an IDE drive appear to work as any normal disk drive and yield a serial number for UMN1.

As previously stated, some SCSI devices present a problem in obtaining the disk serial number in FlexNet Publisher versions prior to 11.6.1. Since some SSDs appear as SCSI devices, this same condition may occur. In scenarios that do not yield a consistent UMN1 value, the fallback should be used. This condition has been fixed in the FlexNet Licensing Service for Windows available in FlexNet Publisher v11.6.1.

RAID

Another issue has been found which allows non-unique and/or random values to be generated for UMN1 when the primary hard disk in an FlexNet Publisher client is a RAID configuration.

This has been fixed in the FlexNet Publisher v11.6.1 client code, so updating to the latest FlexNet Licensing Service will not resolve this issue. Therefore, UMN1 should not be used in such systems in versions of FlexNet Publisher prior to 11.6.1.

The behavior for UMN1 in v11.6.1 is for UMN1 to be denoted as "not available", and UMN2 should be used.

Virtual Machines

The same issue mentioned above for RAID systems also causes non-unique values to be generated for UMN1 in virtualized environments. This has been fixed in the FlexNet Publisher v11.6.1 client code, so it is not possible to pick up this fix by updating just the FlexNet Licensing Service. UMN1 should not be used in such systems in versions of FlexNet Publisher prior to 11.6.1.

The behavior for UMN1 in v11.6.1 is for UMN1 to be denoted as "not available", and the fallback strategy is for UMN2 to be used, as described above.

Note that while MAC addresses are unique in virtual machine instances, it is possible for a user to modify them which is a possible area for license circumvention.

FlexNet Operations

In versions of FlexNet Operations prior to 11.5, only MID was used.

In versions of FlexNet Operations from 11.5, the following fallback is in place:

All platforms:

- UMN1, then
- MID

In transactions between FlexNet Publisher client and FlexNet Operations, the first choice for client identification made by the server is UMN1. Should UMN1 be unavailable, noted by the lack of a UMN1 entry in the activation request, the fallback is to Machine ID (MID).

In versions of FlexNet Operations starting with 12.7 (where UMN2 is used), the fallback will be:

All platforms:

- UMN1, then
- UMN2, then
- MID

FlexNet Operations can detect this since the version number is included in activation requests with the client and FlexNet Publisher v11.6.1 supports a robust UMN2.

Using UMN in the License Generator Toolkit (LGT)

The recommendation is to adopt the same mechanism as in FlexNet Operations, described above.

The activation request contains the FlexNet Publisher version number, platform type, UMN1, and UMN2 as shown in the file snippet below:

```
 <OriginData>
  <ClientVersion>11.6.1</ClientVersion>
  <ConfigData>
  <IsServer>0</IsServer>
  </ConfigData>
  <Platform>
  <PlatformTypeName>i86_n3</PlatformTypeName>
  <OperatingSystem>
   <Info>none</Info>
  </OperatingSystem>
  </Platform>
  <UniqueMachineNumbers>
  <UniqueMachineNumber>
   <Type>1</Type>
   <Value>024277395D31F263C350A0255CE3C282C7D3AEBF</Value>
  </UniqueMachineNumber>
  <UniqueMachineNumber>
   <Type>2</Type>
   <Value>BA4E766C2FC6387A5F31B03683E445E2B93044A6</Value>
  </UniqueMachineNumber>
  </UniqueMachineNumbers>
  <SequenceNumber>7</SequenceNumber>
 </OriginData>
```

Repair and reinstallation policies

By allowing one or two reinstall requests, the publishers can gracefully accommodate the UMN brittleness problem.

Conclusions

A number of mechanisms are employed in FlexNet Publisher to identify clients to their servers. The merits of these were presented with the recommendation to use the UMN values.

In Windows systems, problems in acquiring consistent and unique values have been experienced in a small number of cases. However, enhancements made to the v11.6.1 FlexNet Publisher client have greatly reduced the likelihood of these problems occurring in the future.

The final recommendation is for users of Windows systems to upgrade to the latest version of FlexNet Publisher to benefit from

- Enable all anchors
- Exclude the two "brittle" host IDs from binding (HID_SYSTEM and HID_INTERNET)
- For versions of FlexNet Publisher pre-11.6.1,
  - Use UMN1 as primary client identifier
  - Use Machine ID (MID) as the fallback
- For versions of FlexNet Publisher post-11.6.1,
  - Use UMN1 as primary client identifier
  - Use UMN2 as the first fallback
  - Use Machine ID (MID) as the second fallback
- Always use the latest version of the FlexNet Windows

For more information on FlexNet Publisher and FlexNet Suite, please visit:
www.flexerasoftware.com/fnp

Flexera Software, Inc.
1000 East Woodfield Road,
Suite 400
Schaumburg, IL 60173 USA

Schaumburg (Global Headquarters),
Santa Clara:
+1 800-809-5659

United Kingdom (Europe,
Middle East Headquarters):
+44 870-871-1111
+44 870-873-6300

Japan (Asia,
Pacific Headquarters):
+81 3-4360-8291

Australia:
+61 2-99-8-22-178

www.flexerasoftware.com