White Paper: Best Practices for Recovering Trusted Storage

Version 1.0
November 2011

1 Abbreviations

The following abbreviations are used in this document:

Acronym	*Description*
TS	Trusted storage
FR	Fulfillment record
FNP	FlexNet Publisher
FNO	FlexNet Operations

2 Introduction

Situations arise in which you might need to recover a trusted storage (TS). The purpose of this document is to identify those situations that require a TS recovery and to provide Flexera Software support and consulting engineers with appropriate methods to accomplish this recovery. The customers involved use both FlexNet Publisher (FNP) and FlexNet Operations (FNO).

2.1 Trusted Storage Security

TS is an encrypted and protected store for licenses. These licenses are installed in TS in the form of fulfillment records (FRs). TS is designed to protect against common license-leakage events, such as following:

License Leakage Event	Protection Method
Copying the license store to another machine	Trusted-storage binding
Backing up and restoring the license store	Trusted-storage anchoring
Reusing trials on a machine	Trial anchoring
Clock -windback to achieve extended duration on expiring licenses	Windback monitoring

3 Common Solutions for Fixing Trusted Storage

TS becomes untrusted when one of the following occurs:

- Binding break
- Anchoring break
- Clock-windback break

3.1 Determining That Storage Is Untrusted

To determine whether TS has become untrusted, run the following command:

```
appactutil -view
```

As shown in this sample, if TS is currently in an untrusted state, the first line in the output shows a message that indicates this state:

```
--------------------------------------------------------------------
Trust Flags: **BROKEN** RESTORE
Fulfillment Type: TRIAL
Status: ENABLED
Fulfillment ID: FIL-EZCALC-S1
Entitlement ID: ENTL-EZCALC-S1
Product ID: EZCALC-S
Suite ID: NONE
Expiration date: 5-feb-2012
Feature line(s):
INCREMENT ADD demo 1.0 31-dec-2020 1 SIGN="0056 F246 9A45 48C2 3CFF \
        9A5A 96F3 4800 AA60 BA9C 2886 5DD0 6696 9CAA 36F5"
INCREMENT SUBTRACT demo 1.0 31-dec-2020 1 SIGN="002E 84DB 3158 4FF0 \
        235C 19F3 65BB 6B00 FF04 6E95 E798 7591 508E E419 8FBC"
INCREMENT MULTIPLY demo 1.0 31-dec-2020 1 SIGN="0082 53B4 A5CB 3A39 \
        961B BF9B 481A F300 9584 0DDA B362 1B07 E522 9111 26A1"
INCREMENT DIVIDE demo 1.0 31-dec-2020 1 SIGN="0014 8624 084B 0E1F \
        90FD A06A B40D 7600 1E19 F361 C162 758E 2E38 6CF3 73AA"

--------------------------------------------------------------------
```

The following table summarizes the different messages that can display to indicate TS has become untrusted:

Message	Storage Break Type
`Trust Flags: **BROKEN** TIME`	Clock windback
`Trust Flags: **BROKEN** HOST`	Binding break
`Trust Flags: **BROKEN** RESTORE`	Anchor break

3.2 Resolving Clock-Windback Breaks

A clock-windback break can occur when the system clock is turned back to a time previous to the last TS update. As long as no updates to TS have occurred since it became untrusted, you can reinstate trust by simply resetting the customer’s system clock to a time later than the last known TS update. (This fix is one example of how you can *locally* revert TS back to a trusted state.)

3.3 Resolving Binding Breaks

The following lists events that cause binding breaks and provides methods for fixing these breaks:

Binding Break Event	Action to Restore Trust
TS has been copied to a different machine.	Use the TS on the original machine.
Significant upgrade of the original machine has occurred.	Issue a repair request.

3.4 Resolving Anchoring Breaks

An anchor is information stored in a location (on the TS machine) that is difficult to observe, modify, or delete and that is synchronized with a sequence number stored separately in TS. An anchoring break occurs when one or more anchors[1] and the value in TS become unsynchronized. The following events can lead to this type of break:

Anchoring Break Event	Action to Restore Trust
Customer backs up TS, updates the now live TS, and then attempts to restore the backed-up TS.	Restore the synchronized TS.
Third-party software (especially antivirus or disk-encryption software) overwrites anchors or blocks access to anchor locations. (For example, some disk-encryption products also use the `track0` to store boot-time data.)	Enhance anchor stability by selecting to use all available anchor types (the default) in trusted configurations. For example, Windows and Mac both have two types of anchors. If one anchor type becomes out-of-sync with TS, the other anchor automatically repairs the break.

3.5 Performing Local Repairs of Trusted Storage

The activation utility (using **flxActCommonRepairLocalTrustedStorage**) implicitly performs a local repair whenever a request is created at a time that TS is in an untrusted state. A local repair makes it possible to perform new updates to TS. However, a local repair causes all broken FRs to remain permanently in their untrusted state until they are restored by a repair transaction.

[1] The sequence number in both anchors and TS is updated each time TS is updated.

3.6 Options When Trusted Storage Requires Repeated Repairs

If TS repeatedly becomes untrusted, even after multiple repairs, consider these options:

- Do not use TS on the system concerned.
- If the repeated breaks involve anchoring, consider allowing that particular activation client to use a customized trusted configuration with all anchoring (or one of the anchors) disabled. To set this up in FNO, one would have to associate a different product with each trusted configuration.
- Raise a support request to investigate the recurring issue on the system concerned.

4 Trusted Storage Corruption

Although an infrequent occurrence, TS can become unusable. An indicator of this state is the following event code listed in the TS event log:

```
EventCode: 3000001e
```

Note • *With the introduction of the TS backup file in FNP 11.6.1 and later, reported instances of TS corruptions are rare.*

5 Resetting Versus Deleting Trusted Storage

If TS becomes unusable in its existing state, you must either delete or reset it. (The FNP kit comes with a `tsreset` utility.)

Resetting TS differs from deleting TS in the following ways:

- When you delete the TS file, you wipe out TS completely. However, `tsreset` leaves the TS file, with FRs removed, but with trusted configuration still intact.
- `tsreset` can locate a publisher's TS file, whereas FNP currently offers no utility or API for allowing a publisher to locate and delete its TS file.
- `tsreset` is intended for use as a publisher testing tool only.

6 Locating the Publisher's Trusted Storage File

The location and name of a publisher's TS file is needed for troubleshooting the broken TS, especially for deleting TS or sending Flexera Software Technical Support required files.

6.1 Trusted Storage and Event Log Names

The name of the TS file has the following format:

```
<publisher name>_[8|0]<7-digit hex publisher id>_tsf.data
```

The name consists of these components:

- The publisher name and publisher id are obtained from `publisher.xml`.

  Note • *The publisher ID is supplied in its decimal value format in* `publisher.xml`*.*

- The value `8` indicates server-side TS; `0` indicates client-side TS.

The TS event log is found in the same directory as the TS file and has the following name format:

```
<publisher name>_[8|0]<7-digit hex publisher id>_event.log
```

6.2 Trusted Storage File Location

The location of TS depends on the operating system.

Caution • *All publishers share the same folder for storing their specific TS on the customer machine. Therefore, never delete the folder; delete only the publisher-specific files within the folder.*

6.2.1 Windows

On a Windows machine, TS is found in this location:

```
<Common Application Data>\FLEXnet
```

Typically, this translates to the following:

- On Windows XP, `C:\Documents and Settings\All Users\Application Data\FLEXnet`
- On Windows 7, `C:\ProgramData\FLEXnet`

Tip • *To locate this folder, call* `SHGetFolderPath` *with the* `CSIDL_COMMON_APPDATA CSIDL` *value; then append* `FLEXnet` *to the resulting string.*

6.2.2 Mac

On a Mac machine, TS is found in this location:

```
/Library/Preferences/FLEXnet Publisher/FLEXnet
```

6.2.3 All Other Unix Platforms

On all other Unix machines, TS is found in this location:

```
/usr/local/share/macrovision/storage/FLEXnet
```

7 Information to Provide Flexera Software Support

If TS becomes repeatedly broken, even after deleting and reinstalling, or after multiple repairs, the situation should be escalated to Flexera Technical Support. Provide Support with the following information:

- Copy of the TS file
- Copy of the TS event log
- Version of FNP activation library (`<name>_libFNP.dll [`or `so` or `dylib]`)
- Version of the FlexNet Publisher Licensing Service (which might be different from the activation library version)
- Error output from **flxActCommonHandleGetError**
- Platform details, including:
  - Operating system name and version
  - Computer model
  - Name of any anti-virus software
  - Name of any disk-encryption software
  - Details of the system-disk RAID system, if appropriate
- Copy of any repair requests generated and the results of processing the corresponding responses
- Copy of the original activation request
- Copy of the trusted configuration used

8 Troubleshooting When Trusted Storage Becomes Unusable

The following sections offer suggestions for troubleshooting a broken TS. The sections include processes for troubleshooting on both the publisher's test site and the customer's production site.

8.1 Testing Trusted Storage at the Publisher's Site

Use the following processes for troubleshooting at the publisher's site.

8.1.1 Using a Virtual Machine for Testing

A publisher might choose to use virtual machines to test TS licensing. An advantage of using a virtual environment for testing is that the publisher can readily return the virtual machine to a pre-TS or pre-anchor snapshot. However, the publisher should be aware of the following:

- The publisher should use only a virtualization stack[2] supported by FNP.
- The behavior of FNP on virtual machines is different from its behavior on physical machines. For example, in a virtual environment, the following can happen:
  - UMN1 values are often unavailable inside a virtual machine

[2] A virtualization stack is the combination of the hypervisor type and version, virtual machine operating system, and the FlexNet Publisher platform and version hosted inside the virtual machine.

  - Starting with FNP 11.10.0, a request generated from a virtual machine is likely to contain both a `<Virtualization>` section and an UMN3 value. These components are absent in requests from physical machines.
- If the publisher's customers are using physical machines, final testing at the publisher's site should occur on a physical machine.

8.1.2 Initial Troubleshooting

Use the following troubleshooting steps to determine the TS issue:

1. Determine whether TS is untrusted because of a clock windback. (See Determining That Storage Is Untrusted.) If a windback is the issue, try setting system time to the correct time.
2. If that fails, run `tsreset`, and re-install the licenses.
3. If that fails, delete TS, and reinstall TS and the licenses.

8.2 Testing Trials at the Publisher Site

Publishers might want to test trial ASRs before distributing them. This testing requires a means to reset trial anchors between tests. Deleting or resetting TS does *not* remove trial anchors. A test machine effectively becomes unusable for testing a trial after a trial has been loaded once, unless a mechanism to remove the trial anchor is provided.

Therefore, FNP provides a mechanism that allows publishers to remove trial anchors for testing purposes. This mechanism consists of the following set of calls and settings:

- Call to **flxActCommonResetTrialASRs** API
- Call to **flxActCommonResetTrialASRFromBuffer** API
- Non-zero value for the `PUBLISHER_TRIAL_RESET` property in the ASR

For details on the APIs, see FlexNet Publisher's *C/C++ Function Reference Guide*. For details on the `PUBLISHER_TRIAL_RESET` property, see the *Programming Reference for Trusted Storage-Based Licensing*.

Caution • *The trial reset functionality is not intended for use in a generally available product. You incur security risks if you make resettable ASRs[3] generally available or if you leave calls to **flxActCommonResetTrialASRs** or **flxActCommonResetTrialASRFromBuffer** in production code. Best practice is to have a separate utility – available only to QA – that embeds the resettable ASR and makes a call to **flxActCommonResetTrialASRFromBuffer**.*

[3] A resettable ASR is one in which the PUBLISHER_TRIAL_RESET property is set to a value other than zero.

8.3 Testing in the Production Environment at the Customer Site

To perform the troubleshooting steps at the customer's site, enhance the publisher's activation utility to perform the following steps automatically.

Step 1. Determine whether a clock-windback error is present. (See Determining That Storage Is Untrusted.)

- If not, proceed to Step 2.
- If so, verify that the system clock on customer's machine is set to the correct time. If their system time is not correct, offer to synchronize their system time with a valid time server.

  Note • *The publisher can synchronize the customer clock without consulting the user. However, the customer's clock might be out of sync for a valid reason, such as for testing. Therefore, the customer should agree to the synchronization.*

  If resetting the customer's system clock clears the error, TS is repaired.

  If the error is not cleared, proceed to Step 2.

Step 2. Attempt a standard repair of TS:

1. Generate a repair request.
2. After processing the repair response, close down the activation utility and any other application that contains handles to TS.[4]
3. Restart the activation utility to determine whether TS is trusted.

   If TS remains untrusted, proceed to Step 3.

Step 3. Perform these tasks:

1. Retain the previous repair request and corresponding response. (See Retaining Request and Response xml.)

2. Run the activation utility to determine whether anchoring is broken. (This step requires a call to **flxActCommonProdLicSpcTrustFlagsGet**.) If anchoring is not broken, proceed to Step 4.
3. If anchors are broken, generate a repair request that uses an anchor-disabled trusted configuration.[5]
4. After processing the repair response, close down the activation utility and any other application that contains handles to TS.
5. Restart the activation utility to determine whether TS is trusted. If TS is still untrusted, proceed to Step 4.
6. If TS is trusted, consider asking the customer to reboot[6]; then confirm whether TS is still trusted.

   If TS is untrusted again, proceed to Step 4.

[4] When the repair response is processed, trust is usually re-established for that session. However, if a persistent problem exists, trust is lost again when TS is next loaded, or possibly after a system reboot.
[5] A more granular approach on Windows is to first disable only track0 anchors, and then try the repair. If the repair fails to restore trust, disable all anchors.
[6] Anchor interference can occur at boot-time. Confirming that TS remains trusted after a boot cycle ensures that no boot-time issues are occurring with anchors.

Step 4. Perform these tasks:

1. Retain the previous repair request and corresponding response. (See Retaining Request and Response xml.)

2. Delete both TS and the backup TS file.

3. Generate an initial activation request to configure TS and reinstall the licenses.
4. After processing the response, close down the activation utility and any other application that contains handles to TS.
5. Restart the activation utility to determine whether TS is trusted.

   If TS is still untrusted, proceed to Step 5.

Step 5. Perform these tasks:

1. Retain the previous request and corresponding response. (See Retaining Request and Response xml.)
2. Archive the TS file, its event log, and all the other components and information listed in Information to Provide Flexera Software Support.
3. Send the archive to Flexera Software Technical Support.

8.3.1 Retaining Request and Response xml

To troubleshoot TS issues, save each activation request issued and its corresponding response.

8.3.1.1 Saving Requests From Online Transactions

Online activations can be carried out using the example activation utilities in a single command. In order to save a request, separate the activation transaction into two separate steps:

1. Generate a request, write it to file, and save it in TS.
2. Send the request to the activation server. (Once you receive the response, you must process it.)

The following are examples of commands that you can use *for single-action transactions* to perform the above two steps.

`appactutil -served -entitlementID myRightsId -gen request.xml` (to generate and save the request)
`appactutil -served -entitlementID myRightsId -comm soap -commServer myServerUri` (to send the request)

The following are examples of commands that you can use *for composite transactions* to perform the above two steps.

`appcomptranutil -new request.xml -activate myRightsId` (to generate and save the request)
`appcomptranutil -stored myRightsId -transaction myServerURI` (to send the request)

8.3.1.2 Saving Responses

In FNP 11.10 and earlier, the activation utility uses the **flxCtuContextFileLoad** API to provide the response `xml`. This feature is available for offline activations only.

FNP 11.10.1 adds the **flxActTransactionGetResponseXML** API, which returns the entire response `xml` from online responses.

8.3.2 UMN Values in Repair Requests

When a repair request is processed by FNO, the UMN values in the request are compared to the UMN values from the original activation request. FNO uses an evolving algorithm for comparing the original UMN values from the first activation request and the current UMN values in the repair request. However, in general, if the subsystem generates UMNs that are all different to the original UMNs, then FNO rejects the repair request. In this case, the only option for re-establishing an FR is to activate and consume a new license instead of generating repair requests. Although a change in UMN values is rare, it can happen, typically after a system upgrade – such as an operating-system or hardware upgrade. For more information about UMNs, refer to the document *White Paper: Understanding the UMN in FlexNet Publisher*.