RvtLock3r

Table of Contents

RvtLock3r

Overview

What is RvtLock3r?

RvtLock3r is a Revit .NET C# custom add-in designed to keep track of any modifications or alteration done on a BIM element.

It is purposely meant to validate that certain BIM element properties have not been modified.

In a real-world scenario, two parties are involved:

Vendor:

Vendor owns the model.

Model elements are associated with certain specific read-only properties.

The vendor wants to protect the shared properties from getting modified or altered with to maintain quality and authenticity.

The vendor has no control of how his / her end users interact with the model.

Consumer:

Consumer interacts with model, shared by the vendor.

Has whole power to use the model whichever they wish.

May or may not want to modify any parameters / properties.

RvtLock3r

Motivation

Revit does not provide any functionality to ensure that parameter values are not modified.

Although there is not a clear functionality to ensure that end users do not modify the models, RvtLocker provides the journey process as follows:

1. Computes checksum for the original parameter values of the BIM.
2. Implements validation method to ensure that intended values are not modified either during transmission, interaction, or handling.
    I. Manually

    For testing purposes, the validation function is implemented as an external command, launched manually by the user.

    II. Automatically

    Triggered automatically on opening or saving a document to notify the user that undesired tampering has taken place.

    III. Real-time

    Use Dynamic Model Updater framework DMU to detect and prevent any such tampering immediately in real-time.

Change Analysis Caveat

BIM360 and ACC design collaboration provide a change visualization interface that enables you to find model differences by Model Properties API. It is based on the Forge Model Properties API. Another alternative approach to this task.

To sum up, it is impossible to prevent the user from corrupting models if they really try. A large level of trust and following best practices is required.

RvtLock3r

Problem Solving Design Overview

Preparation

The vendor prepares a Ground Truth data.

Ground Truth data is the real and the original information of the BIM elements with the associated properties that need protection from modification.

There are various possible approaches to prepare the ground truth, and they can be completely automated, programmatically assisted, or fully manual.

In all three cases, the vendor must determine up front what elements and which of their parameters are to be checked. Retrieve the corresponding parameter values, compute their checksums, and save the above-mentioned triples.

The validation is then done against the saved ground truth triples.

Storage

The ground truth data triples containing the data required for integrity validation needs to be stored somewhere.

There are several options to storing Ground Truth data:

1. Hard-wired directly into the add-in code for a specific BIM, stored in an external text file.
2. Two options are available for storing custom data directly within the RVT project file:
    i. shared parameters and,
    ii. Extensible storage.

Extensible storage is more modern and explicitly tailored for use by applications and data that is not accessible to the end user or even Revit itself. Seems most suitable for our purpose here.

Extensible storage can be added to any database element. However, it interferes least with Revit operation when placed on a dedicated DataStorage element, especially in a work-sharing environment.

Creation and population of a DataStorage element is demonstrated by the named GUID storage for project identification sample.

RvtLock3r

Extensible Storage Options

Two obvious choices for storing the ground truth in extensible storage:

- Store a separate Entity containing ground truth for each BIM Element on the Element itself. In that case, the ground truth no longer consists of triples since the element id is already known.
- Store one single global collection of ground truth triples in a custom DataStorage element.

References:

- Extensible Storage TBC topic group
- Add-ins in a work-sharing environment
- Extensible storage in a worksharing environment
- Named Guid storage for project identification
- Storing a dictionary – use DataStorage, not ProjectInfo

RvtLock3r

Validation

The customer add-in reads a set of ground truth data from some storage location. It contains a list of triples:

- ElementId
- shared parameter GUID
- Checksum

The add-in iterates over all elements and parameters specified by these triples.

Reads the corresponding parameter value.

Calculates its checksum and validates it by comparing with the ground truth value.

Discrepancies are logged and a report is presented to the user.

The add-in does not care what kind of elements or parameters are being examined. That worry is left up to whoever creates the ground truth file.

RvtLock3r

Implementation

Preparation

The add-in implements two commands: CmdGroundTruth and CmdValidation.

CmdGroundTruth
Initialize the ground truth data for a given model.

The command is fully manual and designed for the vendor.

The ground truth data contains a list of triples:

- ElementId
- shared parameter GUID
- Checksum

The vendor can manipulate the command and have the power to decide what data they want to store in the ground truth.

CmdValidation

Performs the validation against the stored ground truth data stored above.

The command reads a set of ground truth data from some storage.

The add-in iterates over all elements and parameters specified by these triples.

Reads the corresponding parameter value.

Calculates its checksum and validates it by comparing with the ground truth value.

Discrepancies are logged and a report is presented to the user.

The add-in does not care what kind of elements or parameters are being examined. That worry is left up to whoever creates the ground truth file.

RvtLock3r

Storage

Extensible Storage

The ground truth data is stored in the Extensible storage using Named Ground Truth Storage.

References for the using Named Ground Truth Storage is Named Guid Storage for Project Identification.

Basically, it allows to Simply create own Ground Truth for the current Revit project and use that to identify it globally forever after.

I define an extensible storage schema named NamedGroundTruthStorage that just stores one single Ground Truth object.

To create a new project identifier, I create a Ground Truth and store it in an extensible storage entity with the above schema on a Revit DataStorage element with a specific element name, e.g., ground_truth_identifier_v0.

To search for an existing project identifier, I can filter for all data storage elements with extensible storage entities containing data matching our specific schema and with the given element name.

RvtLock3r

Validation

Two approaches have been provided to accomplish the validation of the ground truth.

Manually

Using CmdValidation: The vendor can initiate this command manually by clicking the push button Validate under the ribbon tab Lock3r.

Automatically

Using Event drive approach.

DocumentOpened

Using App application, OnDocumentOpened event is invoked. The event gets the active document right after its opened, runs the validation by cross reading the ground truth from extensible storage against the elements of the active document.

If a discrepancy is encountered, the application presents an informational message to the end-user that the document is corrupted and contains tampered information.

It is the choice of the user to react on the message whether to proceed working with the corrupted model or contact the vendor for an original file.

DocumentSaving

The application invokes, OnDocumentSaving event every time an end-user attempts to save a modified model during interaction whether intended or not.

A warning is presented to the end user that he /she is prohibited to modify a certain property and saving is cancelled.

If user decides to terminate / close Revit after a modification, OnDocumentSaving is again invoked providing user with option whether to save the changes or not.

If the user decides not to save the changes and clicks ‘No’ the document does not persist the modified elements. The originality is maintained.

Loophole

If the user decides to save and click ‘Yes’ the model is modified!

RvtLock3r

User Interface

The add-in provides a Ribbon Tab on Revit User Interface called Lock3r.

The ribbon panel contains two push buttons:

1. Ground Truth Button:

   Generates the ground Truth data.

2. Validate Button:
   Implements the validation method manually to check whether any modification was done.

RvtLock3r

Improvements

- Implement DMU dynamic model updater to prevent modification of the protected parameter values
- Implement end user settings to choose validation strategy: command / opening and closing events / DMU
- Migrate to Forge Design Automation for Revit