

# Make Solution/Project/Document checksum persist-able when its data is persist-able. #19179

Closed

heejaechang wants to merge 10 commits into dotnet:master from heejaechang:solutionChecksum

Conversation37

Commits10

Checks0

Files changed9

+170-27

heejaechang commented on 2 May 2017

Contributor

...

currently checksum contains volatile information which makes it not persist-able in some cases. this PR removes such data from checksum if possible.

for solution and project, add only persistable data when calculate ch...

efc2cd2

dnrfcl as added the cla-already-signed label on 2 May 2017

cla-already-signed

heejaechang added PR for Personal Review Only and Awsa-IDE labels on 2 May 2017

PR for Personal Review Only

Awsa-IDE

heejaechang added 9 commits on 2 May 2017

add HasAllInformation part of checksum in project info

857b3c4

made document checksum to be persistable

c3458ad

Fixed build break

35e5424

rearranged code a bit

7d9c64f

added unit test for persistable checksum

ea59771

made test to pass

0e771ad

fix stack overFlow

702e171

add comments

dbf01f1

made version not to be synched to OOP. any service that wants to pers...

46c96af

heejaechang removed the PR for Personal Review Only label on 4 May 2017

PR for Personal Review Only

Reviews

CyrusNajmabadi

Assignees

No one assigned

Labels

Awsa-IDEcla-already-signed

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues.

None yet

3 participants

heejaechang commented on 4 May 2017

Contributor

Author

...

@CyrusNajmabadi @jasonmalinowski @dotnet/roslyn-analysis can you take a look?

heejaechang reviewed on 4 May 2017

View changes

src/Workspaces/Core/Portable/Workspace/Solution/DocumentInfo.cs

... @@ -155,6 +160,11 @@ internal class DocumentAttributes : IChecksummedObject, IObjectWithTable

155 160 public string FilePath { get; }

156 161

157 162 /// <summary>

163 + /// The file path of the project

164 + /// </summary>

165 + public string ProjectFilePath { get; }

heejaechang on 4 May 2017

Author

Contributor

...

I might change name of this to some opaque one..

CyrusNajmabadi reviewed on 4 May 2017

View changes

src/VisualStudio/Core/Def/Implementation/ProjectSystem/DocumentProvider.StandardTextDocument.cs

160 + {

161 + // set project file path if possible. this will let us have persistable checksum

162 + // for the document

163 + Info = Info.WithProjectFilePath(abstractProject.ProjectFilePath);

CyrusNajmabadi on 4 May 2017

Contributor

...

why can't we just have ProjectFilePath in DocumentInfo?

heejaechang on 4 May 2017

Author

Contributor

...

you mean in constructor? sure we can add it to constructor.

CyrusNajmabadi on 4 May 2017

Contributor

...

That would be nice. Creating an object just to 'With' it seems wonky. But it also concerns me that this would only work with AbstractProjects.

CyrusNajmabadi reviewed on 4 May 2017

View changes

src/VisualStudio/Core/Def/Implementation/ProjectSystem/DocumentProvider.StandardTextDocument.cs

160 + {

161 + // set project file path if possible. this will let us have persistable checksum

162 + // for the document

163 + Info = Info.WithProjectFilePath(abstractProject.ProjectFilePath);

CyrusNajmabadi on 4 May 2017

Contributor

...

why can't we just have ProjectFilePath in DocumentInfo?

CyrusNajmabadi on 4 May 2017

Contributor

...

Err... this seems super wonky. should ProjectFilePath just move down to Project? Or Project Info/State?

heejaechang on 4 May 2017

Author

Contributor

...

problem is that state doesn't have parent pointer. documentid have projectid. but document state only contains document file path so can't distinguish itself for linked file. but can't be shared between 2 projects since documentid has projectid.

CyrusNajmabadi reviewed on 4 May 2017

View changes

src/Workspaces/Core/Portable/Workspace/Solution/DocumentInfo.cs

260 220 Id.Writer(writer);

261 221

262 222 writer.WriteString(FilePath);

CyrusNajmabadi on 4 May 2017

Contributor

...

you're changint he format of this. Are you incrementing a version number somewhere so these will not be read improperly by a previous version?

heejaechang on 4 May 2017

Author

Contributor

...

this data is never serialized. so it should be fine. saved data is checksum which is not changed. value of checksum will be changed which naturally cause things to be invalidated.

CyrusNajmabadi on 4 May 2017

Contributor

...

I don't thin kit's fine. because we now have someone trying to get the checksum of something without knowing that that checksum isn't safe for persisting. Someone could write a feature, and teh silently have all their persisted data never actually matter.

If this is possible, i'd like the caller to be able to specify that they want a persistable checksum or not.

heejaechang on 4 May 2017

Author

Contributor

...

so your concern is someone might save data which is not actually persistable but doing it without knowing it? that is already happening in current system. there is nothing that prevent people saving data which is not persistable.

...

sure we can create 2 subtype of checksum which is persistable and non persistable checksum so at least give users a way to check. but, the one who creates the checksum automatically favoring persistable checksum seems better system.

CyrusNajmabadi on 4 May 2017

Contributor

...

so your concern is someone might save data which is not actually persistable but doing it without knowing it?

Yes. Of course :) That's how all the new IDE features are working. So if their checksums are nto actually persistable that's a problem :)

that is already happening in current system. there is nothing that prevent people saving data which is not persistable.

That's a problem :)

ut, the one who creates the checksum automatically favoring persistable checksum seems better system.

Agreed. But right now that's not what this PR is doing. it seems to be silently failing over to non-persistable checksums, and the caller has no idea that that's the case.

CyrusNajmabadi on 4 May 2017

Contributor

...

n/m. i see why we need ProjectFilePath.

CyrusNajmabadi reviewed on 4 May 2017

View changes

src/Workspaces/Core/Portable/Workspace/Solution/DocumentInfo.cs

265 265 {

266 266 // this checksum is not persistable because

267 267 // this info doesn't have non volatile info

268 268 Id.Writer(writer);

CyrusNajmabadi on 4 May 2017

Contributor

...

How would this happen?

heejaechang on 4 May 2017

Author

Contributor

...

solution not built from file can have only id but not file path, think solution we create in unit test or forked solution and etc.

CyrusNajmabadi on 4 May 2017

Contributor

...

UGH. i wish we had not allowed that.

I still don't like that we just silently switch to writing out the ID. can we instead write out something that says "i'm anonymous, and i'm at this index in my parent?"

heejaechang on 4 May 2017

Author

Contributor

...

checksum can be persistable and not persistable. but regardless of that, within same session, checksum must be consistent, deterministic and valid. having ID there make it consistent, deterministic. it just can't be persisted and re-used in different session.

CyrusNajmabadi on 4 May 2017

Contributor

...

it just can't be persisted and re-used in different session.

The problem is that the client has no way to know if a checksum is persistable or not. Thus, they cannot use checksums as the persistence key.

I would prefer if there was either a boolean that says that the checksum needs to be persistable, or have two endpoints: "GetChecksum" and "GetPersistableChecksum". The client can then ask for a persistable checksum and decide what to do if they can't get one.

I get that we may not be able to make a "consistent, deterministic persistable checksum" in all cases. What i'm asking for is that in those cases we don't silently fail over to making a non-persistable checksum.

...

And, at the same time, i would really like us to actually figure out if we can make these persistable in some manner.

CyrusNajmabadi reviewed on 4 May 2017

View changes

src/Workspaces/Core/Portable/Workspace/Solution/DocumentInfo.cs

261 261 {

262 262 writer.WriteString(nameof(DocumentAttributes));

263 263

264 264 if (FilePath == null || ProjectFilePath == null)

CyrusNajmabadi on 4 May 2017

Contributor

...

How would this happen?

heejaechang on 4 May 2017

Author

Contributor

...

answered above.

CyrusNajmabadi reviewed on 4 May 2017

View changes

src/Workspaces/Core/Portable/Workspace/Solution/ProjectInfo.cs

479 479 {

480 480 writer.WriteString(nameof(ProjectAttributes));

481 481

482 482 if (FilePath == null)

CyrusNajmabadi on 4 May 2017

Contributor

...

How does this happen?

heejaechang on 4 May 2017

Author

Contributor

...

answered above.

CyrusNajmabadi commented on 4 May 2017

Contributor

...

I approve of the overall concept here. Namely that IDs are not safe for persisting. However, some things seem strange here and it seems too easy to get things wrong and not realize it.

CyrusNajmabadi commented on 4 May 2017

Contributor

...

For example, i would much rather have the system throw if it can't give the info you need. For example, if you try to persist or make a checksum, and the system can't give it to you, it shouldn't silently switch to using IDs. It should throw so you know something is wrong.

heejaechang commented on 4 May 2017

Contributor

Author

...

@CyrusNajmabadi but we already created system that not having FilePath. Project FilePath is completely fine for forked and in memory solution.

CyrusNajmabadi commented on 4 May 2017

Contributor

...

But we already created system that not having FilePath

I agree. My point is simply that to some feature, it's impossible to tell if checksums are working properly or not. They'll get checksums that look ok, but which don't actually work well.

For anonymous documents, i would prefer if we came up with a checksum encoding for them that did not use Id.

For example, we could write things out in a specific form:

1. A bit saying if we have a name or not.

2. If the bit is set, it is followed by teh string name.

3. If the bit is not set, it is followed by an int specifying what index the document is in the project.

heejaechang commented on 4 May 2017

Contributor

Author

...

@CyrusNajmabadi I am not sure whether I am following you. checksum is still valid and consistent and deterministic regardless it has ID or not within same session.

it is just not persist-able. so I dont think the checksum is broken. It is fine.

again, if your desire is wanting to know which checksum is persistable or not, we can add property in Checksum type that says whether checksum is persistable or not.

CyrusNajmabadi commented on 4 May 2017

Contributor

...

again, if your desire is wanting to know which checksum is persistable or not, we can add property in Checksum type that says whether checksum is persistable or not.

Good. Do that. :)

Here's what i want, in order of desire:

1. Make it so that all checksums can be persistable. I'm not convinced that we can't provide that.

2. If we can't provide that, provide some mechanism so that a client knows if a checksum is persistable or not.

CyrusNajmabadi commented on 4 May 2017

Contributor

...

Some scenarios aren't clear to me. Can something like the FilePath change for a project? If so, will all teh Documents in that project get their ProjectFilePath updated somehow? Or can that never happen?

CyrusNajmabadi commented on 4 May 2017

Contributor

...

Are there non test scenarios where we create Projects/Files with no filepath and we use them in the OOP process?

If it's only for tests, then i would rather us update the tests to pass in dummy names, and then make it so that getting the checksum is something that fails for these sorts of projects/files.

heejaechang commented on 4 May 2017 • edited

Contributor

Author

...

"Some scenarios aren't clear to me. Can something like the FilePath change for a project? If so, will all teh Documents in that project get their ProjectFilePath updated somehow? Or can that never happen?"

I assumed, it won't since moving file to different project caused new document to be added, but let me make it more explicitly checked.

heejaechang commented on 4 May 2017

Contributor

Author

...

"Are there non test scenarios where we create Projects/Files with no filepath and we use them in the OOP process?"

If it's only for tests, then i would rather us update the tests to pass in dummy names, and then make it so that getting the checksum is something that fails for these sorts of projects/files"

we have both tests to make sure checksum is valid in any case. checksum being persistable is extra thing. main purpose of checksum is being checksum of data regardless whether that data is persistable or not.

CyrusNajmabadi commented on 4 May 2017

Contributor

...

Your PR is titled "Make checksum persistable" :)

heejaechang changed the title Make checksum persistable Make Solution/Project/Document checksum persist-able when its data is persist-able. on 4 May 2017

heejaechang commented on 4 May 2017

Contributor

Author

...

changed title.

CyrusNajmabadi commented on 4 May 2017

Contributor

...

Heejae,

If there are no actual scenarios that are imporant for non-persistable checksums, i would like to just disallow them from the system entirely. Non-persistable checksums are very easy to screw up. Avoiding them entirely would be desirable from a design perspective.

Again, if this only comes up in tests, and in all real OOP scenarios, the checksums should be persistable, then i would rather that just be an invariant that we can enforce and which code can depend on. Otherwise, we risk silently introducing unintentional changes in behavior down the line.

So, again are there actual non-test scenarios that both need OOP and have null paths? If not, then let's just disallow it for now. We can always weaken that constraint later. But we should nto start with a weak system if it doesn't solve an actual need.

heejaechang commented on 4 May 2017

Contributor

Author

...

I think we shouldn't mix checksum for persistancy. non persistable data can surely have checksum which is valid within same session. it is not the checksum that is persistable or not. it is the data which is persistable or not.

heejaechang closed this on 25 Apr 2018

Sign up for free to join this conversation on GitHub. Already have an account? Sign in to comment