# Demo: Pyrrho DBMS and concurrency

Malcolm Crowe, 8 June 2021

This demo is updated from a tutorial at DBKDA 2021.

We look in detail at Pyrrho's Commit() method for a transaction, and the detection of conflicts.

At the start of Transaction Commit, there is a validation check, to ensure that the transaction still fits on the current shared state of the database, that is, that we have no conflict with transaction that committed since our transaction started.

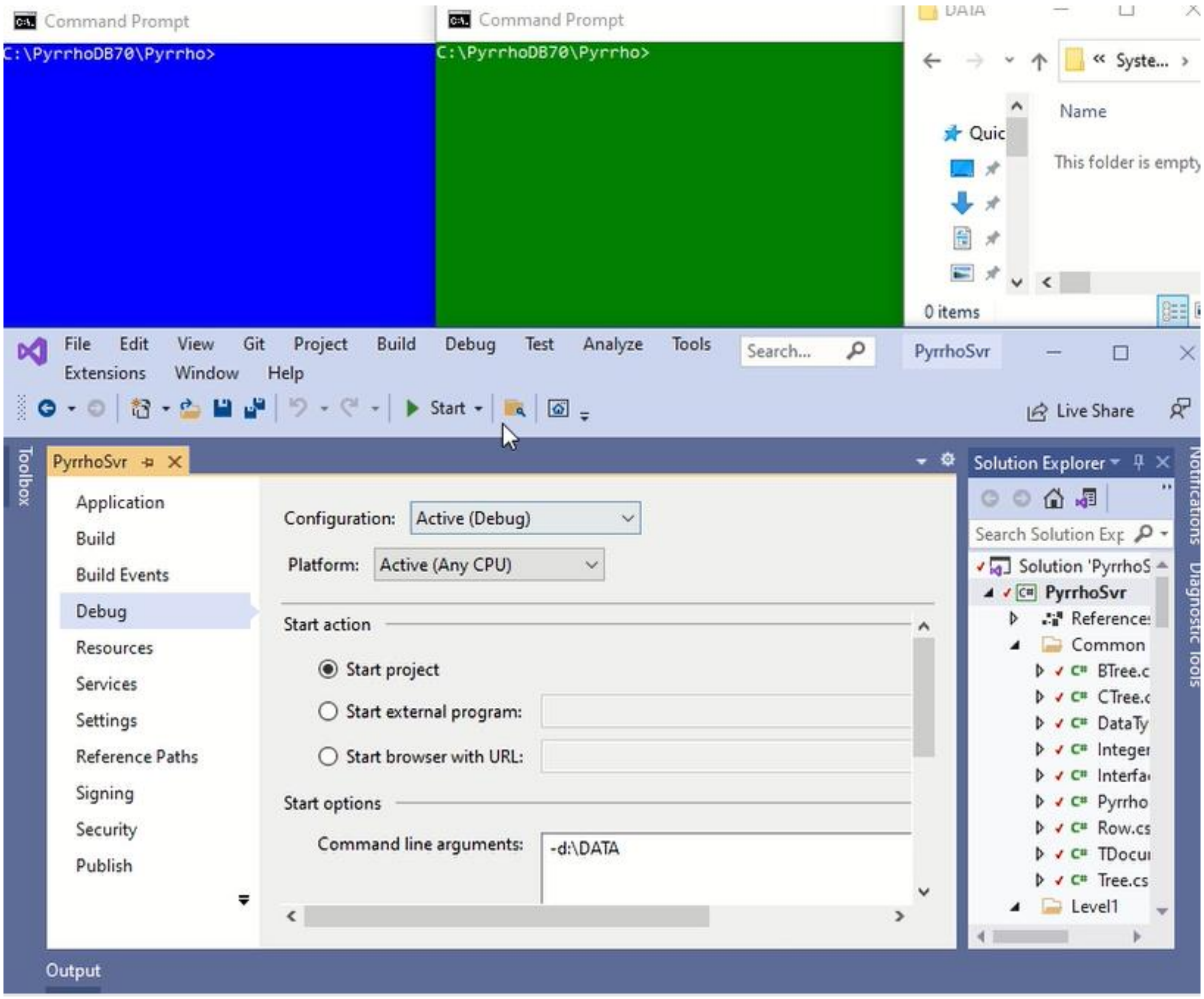
If that is the case, we can relocate all our proposed changes to come after the committed transactions.

The tests for write-write conflicts involve comparing our list of physicals with those of the other transactions.

For checking read-write conflicts, we collect "read constraints" when we are making Cursors.

## Part 1: Setting up the demo

Use Visual Studio to open the solution in Pyrrho's src\Shared folder. In the Solution properties, under Debug>Command Line Arguments, enter -d: followed by a suitable database folder such as \DATA. The folder should not contain a file t10.



```
C:\PyrrhoDB70\Pyrrho>
```

```
C:\PyrrhoDB70\Pyrrho>
```

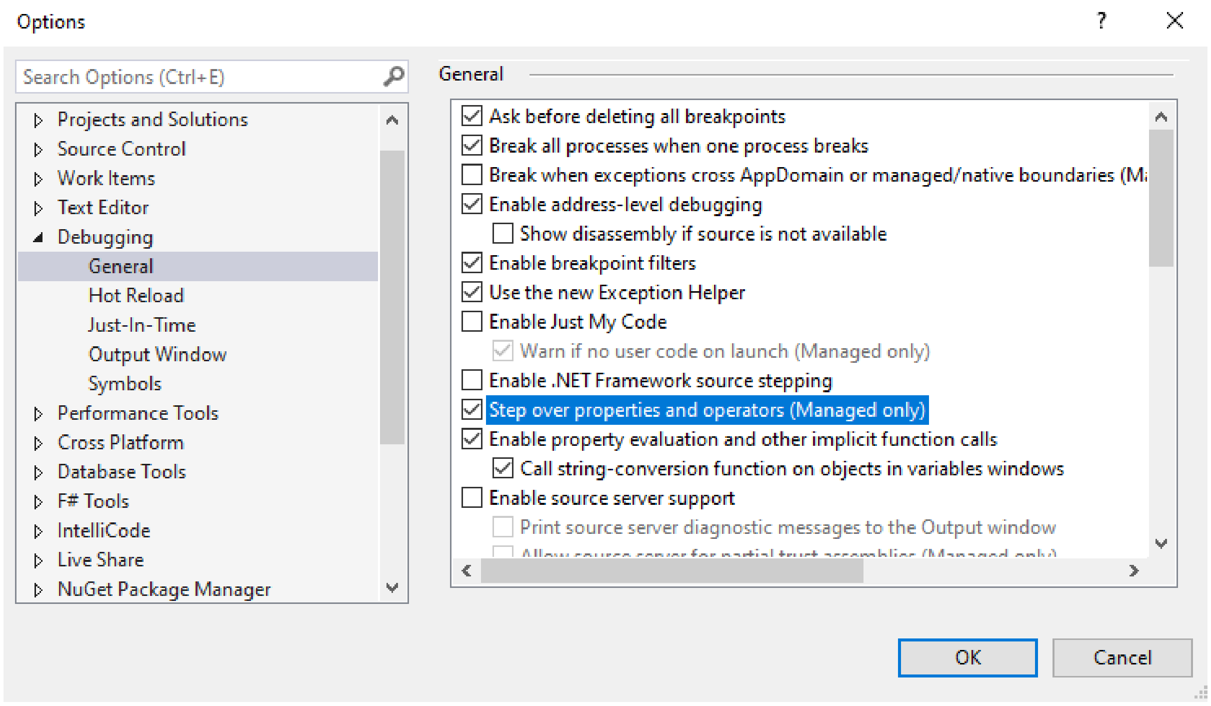
Configuration: Active (Debug)

Platform: Active (Any CPU)

Start action: Start project

Command line arguments: -d:\DATA

Also ensure that in the Debug>Options.. window,



Options

Search Options (Ctrl+E)

- Projects and Solutions
- Source Control
- Work Items
- Text Editor
- Debugging
  - General
  - Hot Reload
  - Just-In-Time
  - Output Window
  - Symbols
- Performance Tools
- Cross Platform
- Database Tools
- F# Tools
- IntelliCode
- Live Share
- NuGet Package Manager

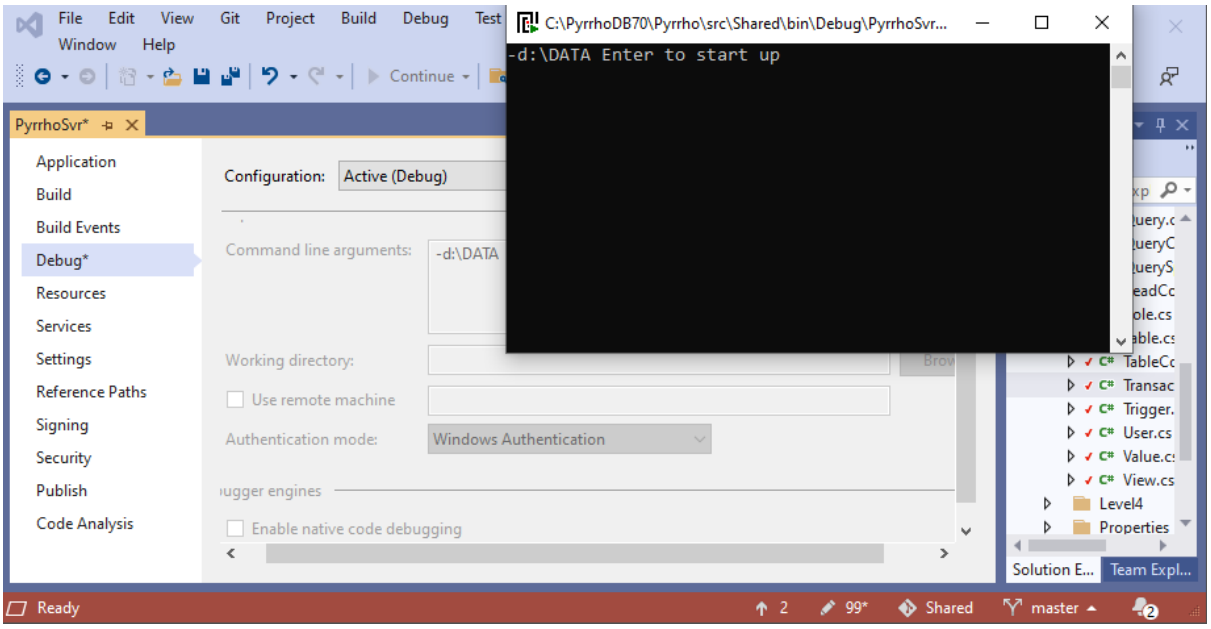
General

- [✓] Ask before deleting all breakpoints
- [✓] Break all processes when one process breaks
- [ ] Break when exceptions cross AppDomain or managed/native boundaries (Mi
- [✓] Enable address-level debugging
  - [ ] Show disassembly if source is not available
- [✓] Enable breakpoint filters
- [✓] Use the new Exception Helper
- [ ] Enable Just My Code
  - [✓] Warn if no user code on launch (Managed only)
- [ ] Enable .NET Framework source stepping
- [✓] Step over properties and operators (Managed only)
- [✓] Enable property evaluation and other implicit function calls
  - [✓] Call string-conversion function on objects in variables windows
- [ ] Enable source server support
  - [ ] Print source server diagnostic messages to the Output window

OK | Cancel

the checkbox Step over properties and operators (managed only) is checked.

Click Start.



C:\PyrrhoDB70\Pyrrho\src\Shared\bin\Debug\PyrrhoSvr...

```
-d:\DATA Enter to start up
```

PyrrhoSvr*

Configuration: Active (Debug)

Command line arguments: -d:\DATA

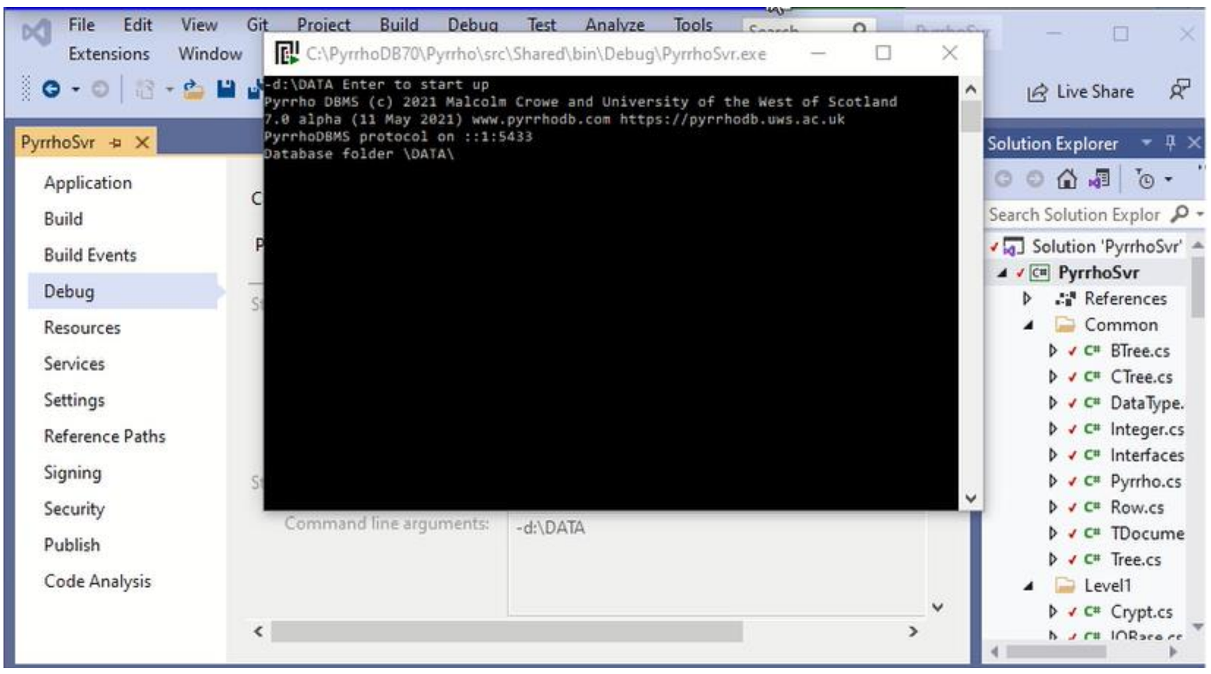
Working directory:

Use remote machine

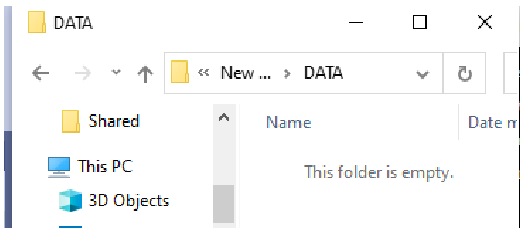
Authentication mode: Windows Authentication

Enable native code debugging

In the pop-up command window, hit the enter key to start up the server:



```
-d:\DATA Enter to start up
Pyrrho DBMS (c) 2021 Malcolm Crowe and University of the West of Scotland
7.0 alpha (11 May 2021) www.pyrrhodb.com https://pyrrhodb.uws.ac.uk
PyrrhoDBMS protocol on ::1:5433
Database folder \DATA\
```

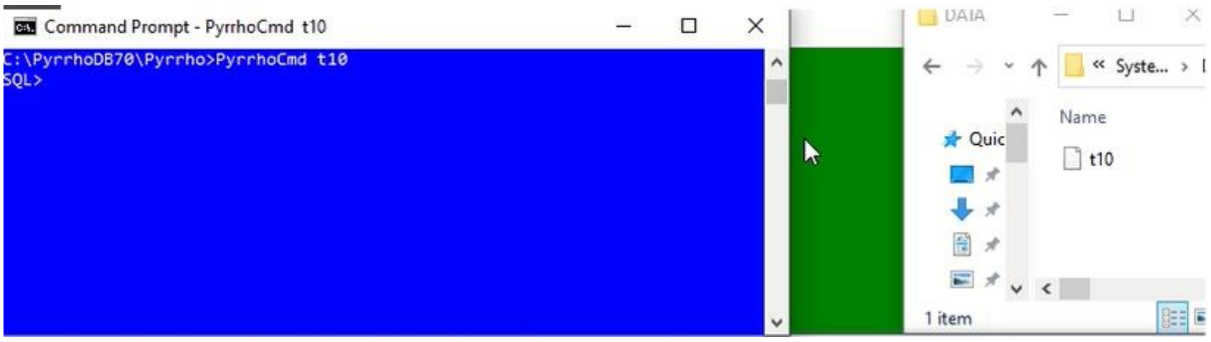


Minimise the pop-up command window.

At this point ensure there is no database t10 in the nominated folder.

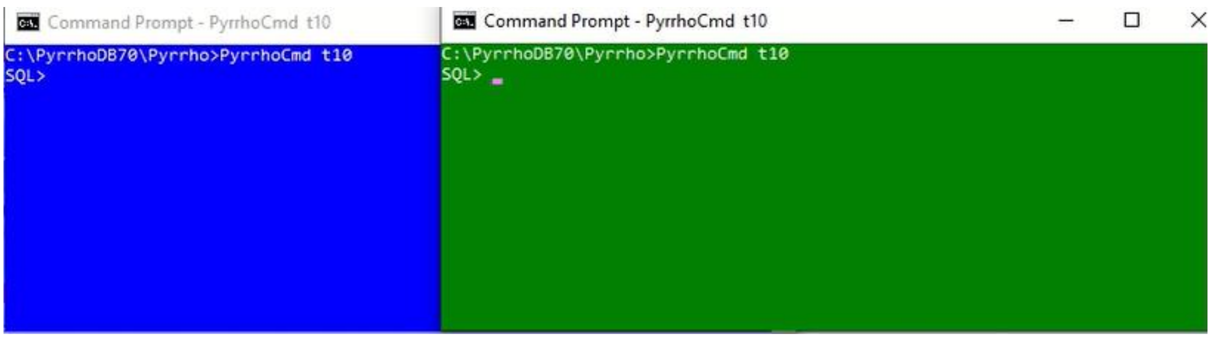
The command we want to run in both windows is the same: PyrrhoCmd t10. Once we do one of them, the DBMS immediately creates the database as we have seen in demo 1.

**`PyrrhoCmd t10`**



```
C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL>
```

From now on, the folder window can be hidden (it does not change).



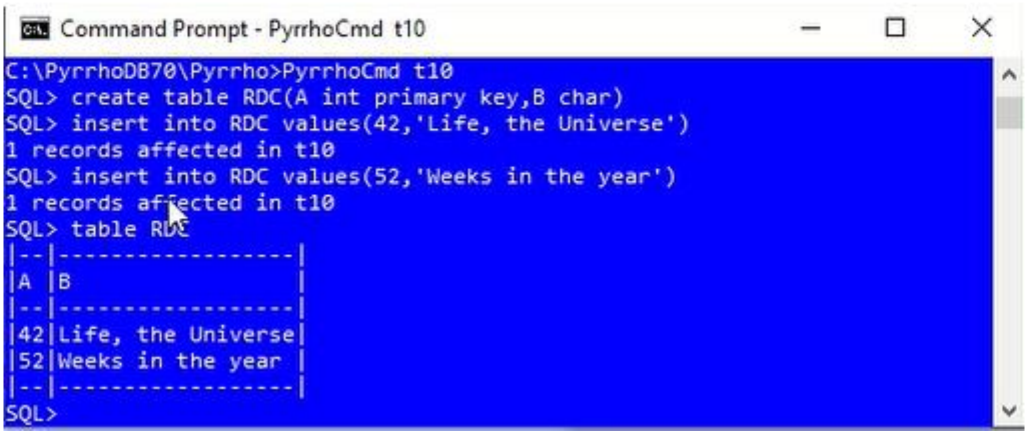
```
C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL>
```

```
C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL>
```

Command Prompt - PyrrhoCmd t10
Command Prompt - PyrrhoCmd t10

In the blue window, let us create a table (it is called RDC because we will demonstrate read-write conflicts), by giving the following commands:

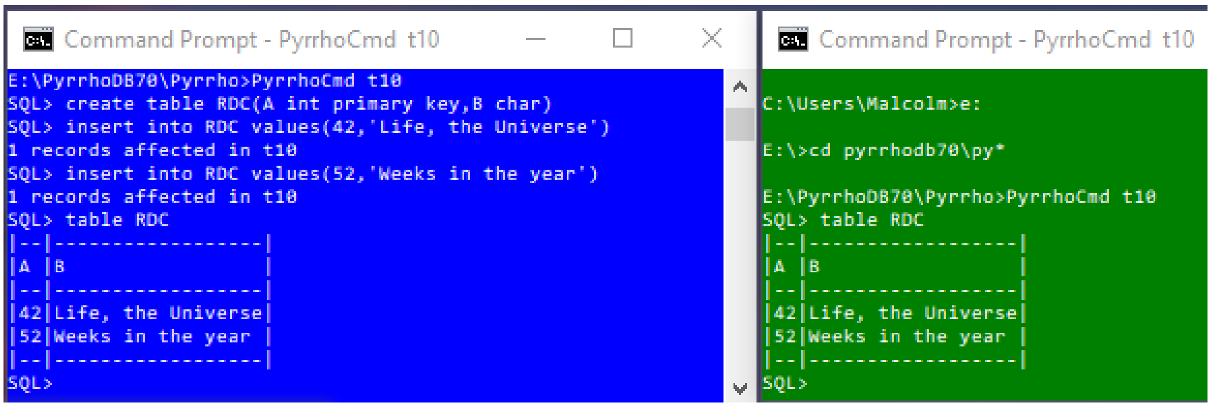
```
create table RDC(A int primary key,B char)

insert into RDC values(42,'Life, the Universe')

insert into RDC values(52,'Weeks in the year')

table RDC
```



```
Command Prompt - PyrrhoCmd t10
C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> create table RDC(A int primary key,B char)
SQL> insert into RDC values(42,'Life, the Universe')
1 records affected in t10
SQL> insert into RDC values(52,'Weeks in the year')
1 records affected in t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL>
```

That has all been committed, because we are running in auto-commit mode, so the green window will pick it up. It is also running in auto-commit mode, so a new command starts a new transaction, and it will check the current state of the database.

```
table RDC
```



```
Command Prompt - PyrrhoCmd t10
E:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> create table RDC(A int primary key,B char)
SQL> insert into RDC values(42,'Life, the Universe')
1 records affected in t10
SQL> insert into RDC values(52,'Weeks in the year')
1 records affected in t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL>
```

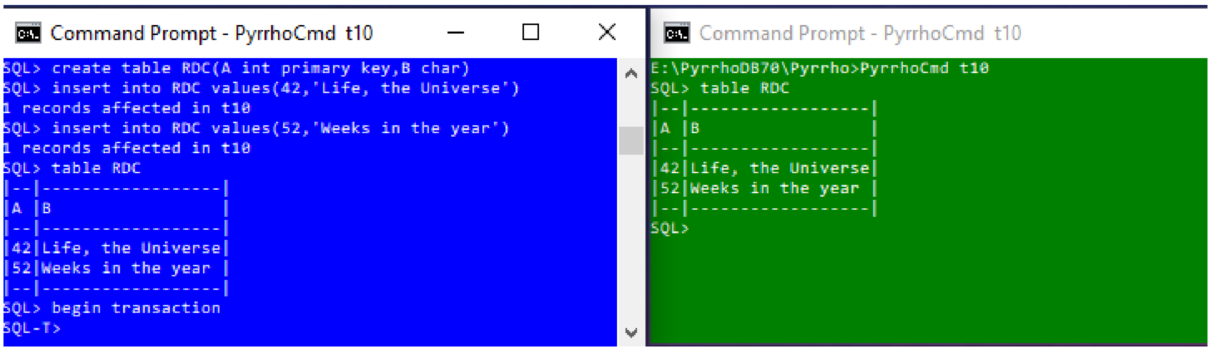
```
Command Prompt - PyrrhoCmd t10
C:\Users\Malcolm>e:

E:\>cd pyrrhodb70\py*

E:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL>
```

This completes the set up for this demo. We will repeat this part of the demo later.

## Part 2: A Write-write conflict

Now we start an explicit transaction in the blue window.

```
begin transaction
```


Command Prompt - PyrrhoCmd t10
Command Prompt - PyrrhoCmd t10

```
SQL> create table RDC(A int primary key,B char)
SQL> insert into RDC values(42,'Life, the Universe')
1 records affected in t10
SQL> insert into RDC values(52,'Weeks in the year')
1 records affected in t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> begin transaction
SQL-T>
```

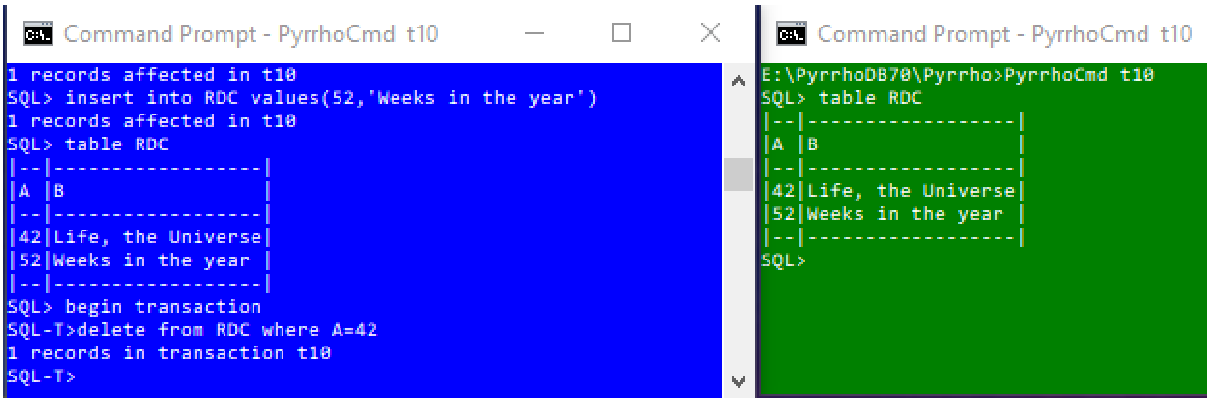
```
E:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL>
```

If we just relied on auto-commit mode it is very difficult to synchronise an overlap of transactions. For a demo, it works very well to use explicit transactions.

Notice that in an explicit transaction the prompt changes to SQL-T>. As long as the transaction is running, we will get these prompts. When the transaction is over, either because of commit, or rollback, or because we have done something wrong, it will go back to SQL>.

We will request conflicting changes to table RDC in these two clients. In the blue window, let's "delete from RDC where A=42", to delete the first row.

```
delete from RDC where A=42
```


Command Prompt - PyrrhoCmd t10
Command Prompt - PyrrhoCmd t10

```
1 records affected in t10
SQL> insert into RDC values(52,'Weeks in the year')
1 records affected in t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> begin transaction
SQL-T>delete from RDC where A=42
1 records in transaction t10
SQL-T>
```

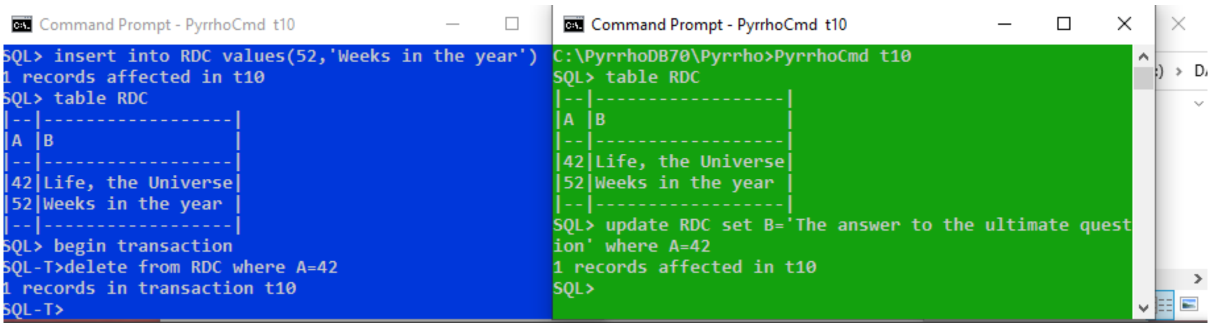
```
E:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL>
```

That's in a transaction, so nothing has been written to disk yet.

In the green window, we are still in auto-commit mode and we are going to make an update to the same row, which will be committed straightaway to disk.

```
update RDC set B='The answer to the ultimate question' where A=42
```

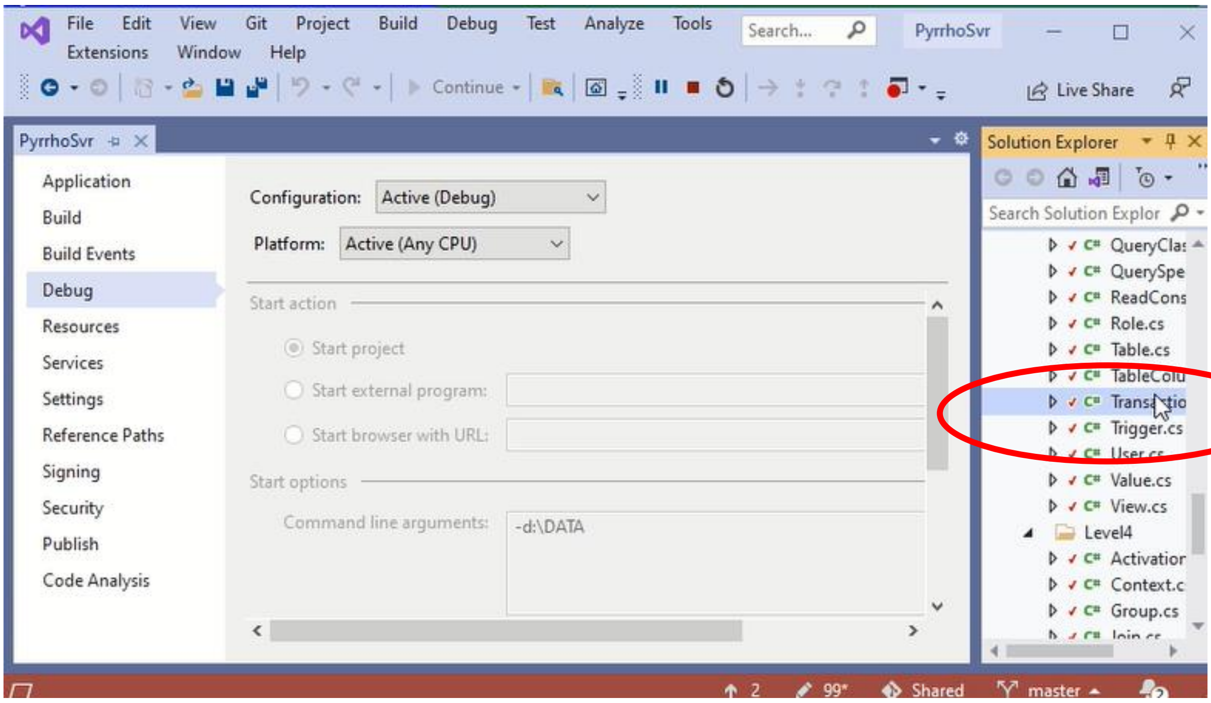
This should make the blue window unable to commit its transaction because of the conflict.


Command Prompt - PyrrhoCmd t10
Command Prompt - PyrrhoCmd t10

```
SQL> insert into RDC values(52,'Weeks in the year')
1 records affected in t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> begin transaction
SQL-T>delete from RDC where A=42
1 records in transaction t10
SQL-T>
```

```
C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> update RDC set B='The answer to the ultimate quest
ion' where A=42
1 records affected in t10
SQL>
```

In order to see what happens, let us set a breakpoint in the debugger. In Solution Explorer, in the Level3 folder, locate file Transaction.cs and double-click.



Configuration: Active (Debug)

Platform: Active (Any CPU)

Start action

Start project

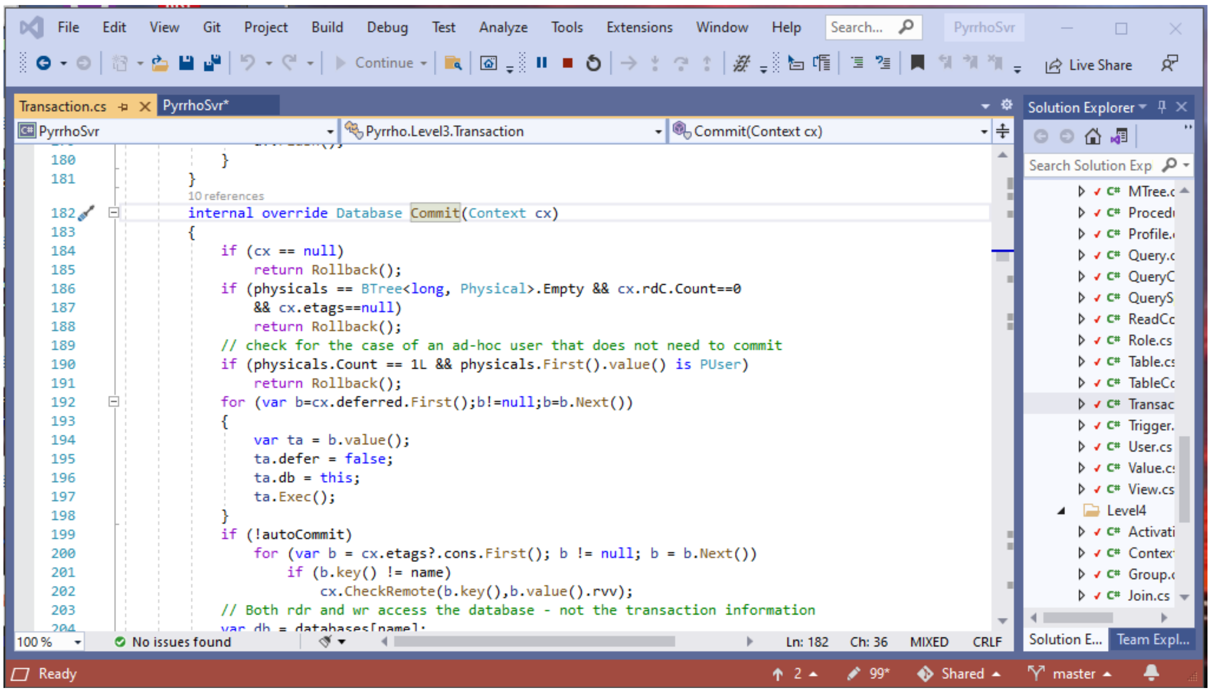
Start external program:

Start browser with URL:

Start options

Command line arguments: -d:\DATA

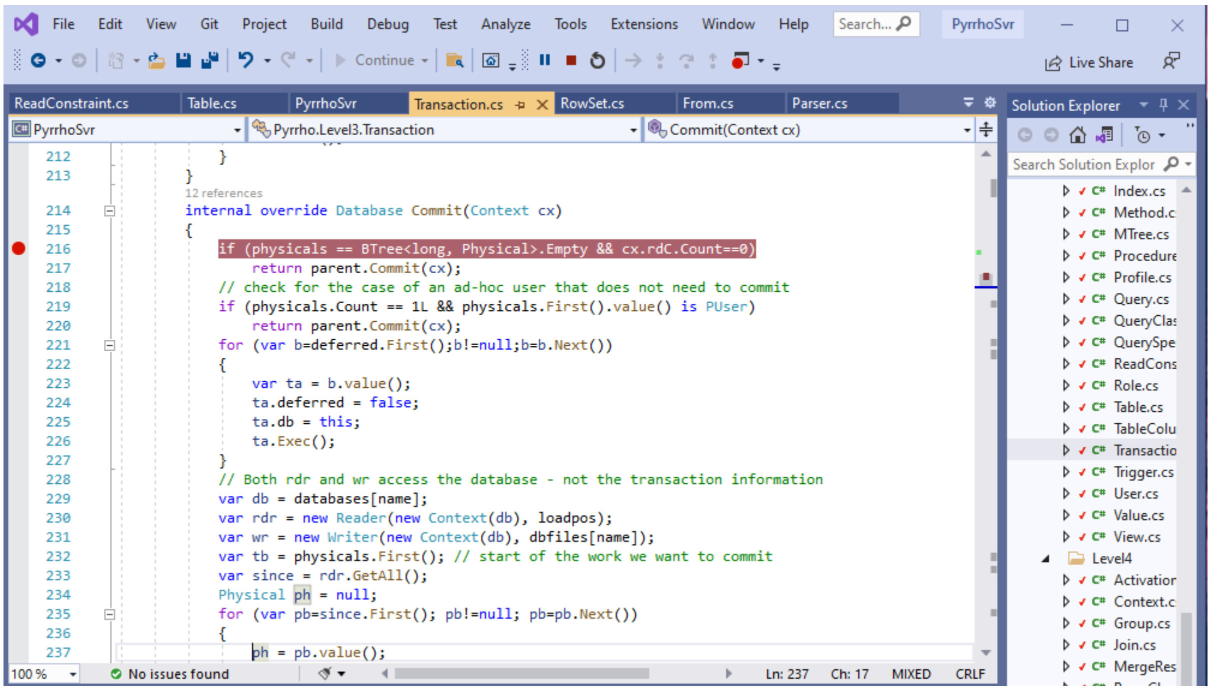
Scroll down to line 182 at the start of the Transaction.Commit() method.



```
180                 }
181             }
                10 references
182             internal override Database Commit(Context cx)
183             {
184                 if (cx == null)
185                     return Rollback();
186                 if (physicals == BTree<long, Physical>.Empty && cx.rdC.Count==0
187                     && cx.etags==null)
188                     return Rollback();
189                 // check for the case of an ad-hoc user that does not need to commit
190                 if (physicals.Count == 1L && physicals.First().value() is PUser)
191                     return Rollback();
192                 for (var b=cx.deferred.First();b!=null;b=b.Next())
193                 {
194                     var ta = b.value();
195                     ta.defer = false;
196                     ta.db = this;
197                     ta.Exec();
198                 }
199                 if (!autoCommit)
200                     for (var b = cx.etags?.cons.First(); b != null; b = b.Next())
201                         if (b.key() != name)
202                             cx.CheckRemote(b.key(),b.value().rvv);
203                 // Both rdr and wr access the database - not the transaction information
```

100 %   No issues found   Ln: 182   Ch: 36   MIXED   CRLF

Click the mouse in the left margin to set a break point at line 184 of the Transaction.cs file.

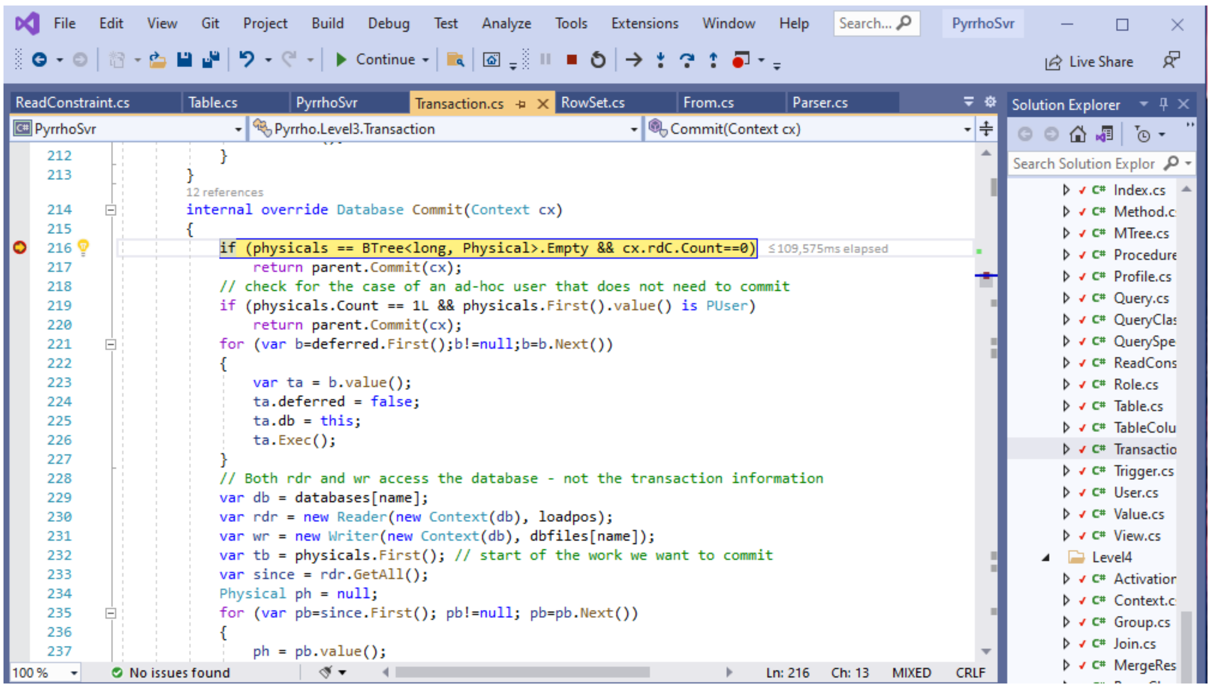

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PyrrhoSvr
Continue
Live Share
ReadConstraint.cs Table.cs PyrrhoSvr Transaction.cs RowSet.cs From.cs Parser.cs
Pyrrho.Level3.Transaction
Commit(Context cx)
Solution Explorer
Index.cs Method.c MTree.cs Procedure Profile.cs Query.cs QueryClas QuerySpe ReadCons Role.cs Table.cs TableColu Transactio Trigger.cs User.cs Value.cs View.cs
Level4
Activation Context.c Group.cs Join.cs MergeRes
100 % No issues found Ln: 237 Ch: 17 MIXED CRLF

```
        }
    }
    12 references
    internal override Database Commit(Context cx)
    {
        if (physicals == BTree<long, Physical>.Empty && cx.rdC.Count==0)
            return parent.Commit(cx);
        // check for the case of an ad-hoc user that does not need to commit
        if (physicals.Count == 1L && physicals.First().value() is PUser)
            return parent.Commit(cx);
        for (var b=deferred.First();b!=null;b=b.Next())
        {
            var ta = b.value();
            ta.deferred = false;
            ta.db = this;
            ta.Exec();
        }
        // Both rdr and wr access the database - not the transaction information
        var db = databases[name];
        var rdr = new Reader(new Context(db), loadpos);
        var wr = new Writer(new Context(db), dbfiles[name]);
        var tb = physicals.First(); // start of the work we want to commit
        var since = rdr.GetAll();
        Physical ph = null;
        for (var pb=since.First(); pb!=null; pb=pb.Next())
        {
            ph = pb.value();
```

Then when we issue the commit command in the blue window,

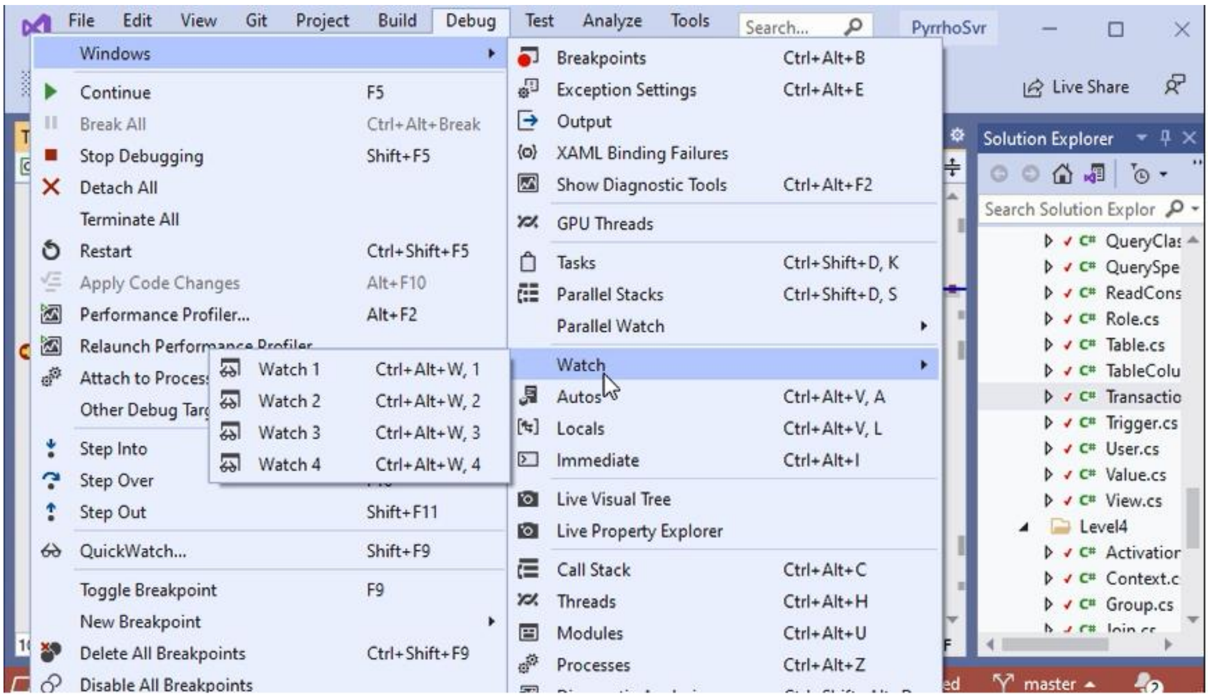
**`commit`**

Visual Studio stops at the breakpoint.

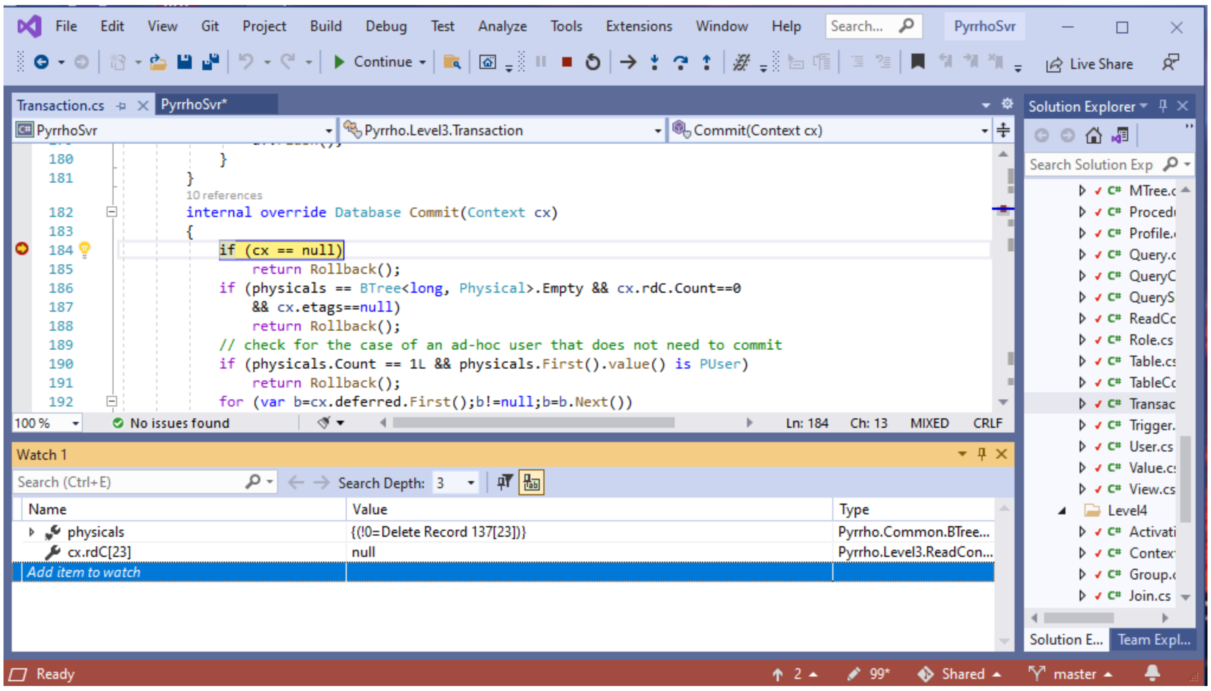

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PyrrhoSvr
Continue
Live Share
ReadConstraint.cs Table.cs PyrrhoSvr Transaction.cs RowSet.cs From.cs Parser.cs
Pyrrho.Level3.Transaction
Commit(Context cx)
Solution Explorer
Index.cs Method.c MTree.cs Procedure Profile.cs Query.cs QueryClas QuerySpe ReadCons Role.cs Table.cs TableColu Transactio Trigger.cs User.cs Value.cs View.cs
Level4
Activation Context.c Group.cs Join.cs MergeRes
100 % No issues found Ln: 216 Ch: 13 MIXED CRLF

```
        }
    }
    12 references
    internal override Database Commit(Context cx)
    {
        if (physicals == BTree<long, Physical>.Empty && cx.rdC.Count==0)  ≤109,575ms elapsed
            return parent.Commit(cx);
        // check for the case of an ad-hoc user that does not need to commit
        if (physicals.Count == 1L && physicals.First().value() is PUser)
            return parent.Commit(cx);
        for (var b=deferred.First();b!=null;b=b.Next())
        {
            var ta = b.value();
            ta.deferred = false;
            ta.db = this;
            ta.Exec();
        }
        // Both rdr and wr access the database - not the transaction information
        var db = databases[name];
        var rdr = new Reader(new Context(db), loadpos);
        var wr = new Writer(new Context(db), dbfiles[name]);
        var tb = physicals.First(); // start of the work we want to commit
        var since = rdr.GetAll();
        Physical ph = null;
        for (var pb=since.First(); pb!=null; pb=pb.Next())
        {
            ph = pb.value();
```

Make the Debug>Windows>Watch>Watch 1 window visible,



and Dock it below the text window. Use it to examine `physicals` and `cx.rdC[23]` :



```
        }
    }
    10 references
    internal override Database Commit(Context cx)
    {
        if (cx == null)
            return Rollback();
        if (physicals == BTree<long, Physical>.Empty && cx.rdC.Count==0
            && cx.etags==null)
            return Rollback();
        // check for the case of an ad-hoc user that does not need to commit
        if (physicals.Count == 1L && physicals.First().value() is PUser)
            return Rollback();
        for (var b=cx.deferred.First();b!=null;b=b.Next())
```

Watch 1

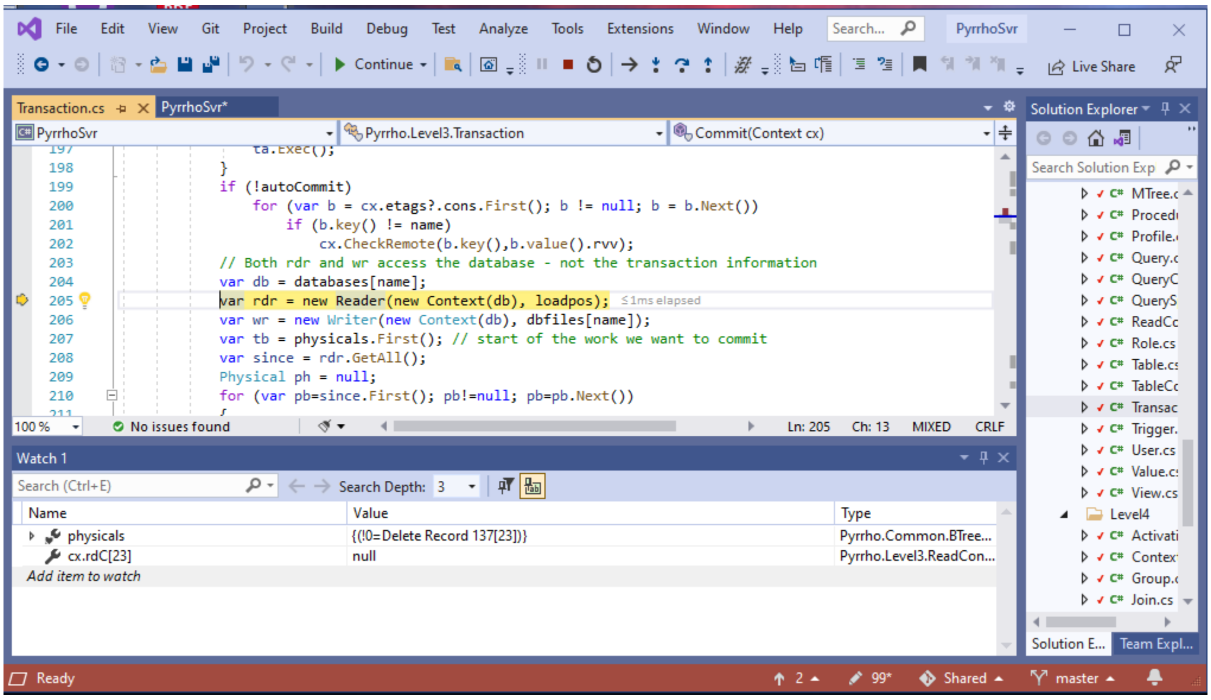
Search (Ctrl+E) Search Depth: 3

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| Add item to watch | | |

Let's look at the physicals. **physicals** is the list of Physical records that the Transaction wishes to commit, and it's just the single Physical record to delete row 137 in the database which is "Life, the Universe".

Also look at cx.rdC[23], which is to do with the ReadConstraints for this particular Transaction. It is null, as we have read no rows. (In the next experiment it will be the other way round.)

Step Over a few times to get to line 205.



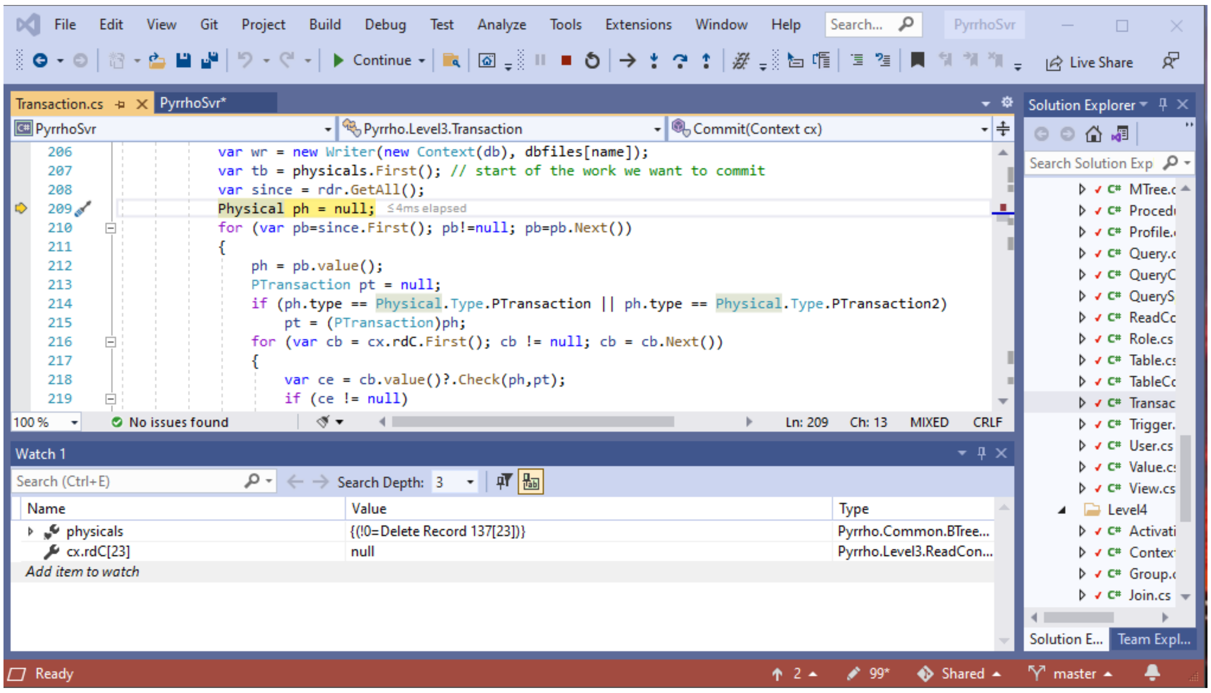
```
197                 ta.Exec();
198             }
199             if (!autoCommit)
200                 for (var b = cx.etags?.cons.First(); b != null; b = b.Next())
201                     if (b.key() != name)
202                         cx.CheckRemote(b.key(),b.value().rvv);
203             // Both rdr and wr access the database - not the transaction information
204             var db = databases[name];
205             var rdr = new Reader(new Context(db), loadpos);  ≤1ms elapsed
206             var wr = new Writer(new Context(db), dbfiles[name]);
207             var tb = physicals.First(); // start of the work we want to commit
208             var since = rdr.GetAll();
209             Physical ph = null;
210             for (var pb=since.First(); pb!=null; pb=pb.Next())
```

```
Name          Value                            Type
physicals     {(!0=Delete Record 137[23])}     Pyrrho.Common.BTree...
cx.rdC[23]    null                             Pyrrho.Level3.ReadCon...
```

The validation step will use the up-to-date copy of the database, as it was left by the green window. We also open a Reader and a Writer: a Reader to look at this database, specifically at the records that have been committed since the start of our Transaction; and a Writer, where we will prepare the records that we are going to add to the database if our validation succeeds.

These are not shareable and are subject to locking protocols. They also work on the same FileStream. (Transaction Commit() repeats the validation step after locking the FilleStream.)

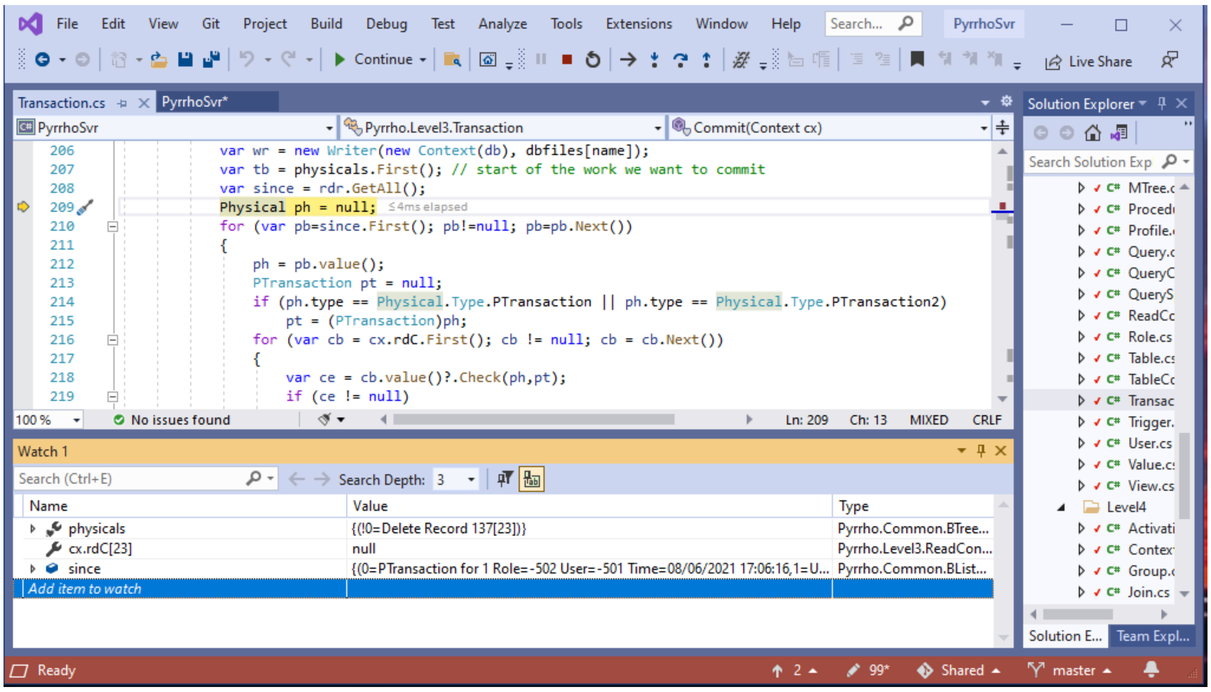
Step Over to line 209, just after a line saying “since=rdr.GetAll();” This gets the records that have been committed to the database since the start of our transaction.



```
206             var wr = new Writer(new Context(db), dbfiles[name]);
207             var tb = physicals.First(); // start of the work we want to commit
208             var since = rdr.GetAll();
209             Physical ph = null;  ≤4ms elapsed
210             for (var pb=since.First(); pb!=null; pb=pb.Next())
211             {
212                 ph = pb.value();
213                 PTransaction pt = null;
214                 if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
215                     pt = (PTransaction)ph;
216                 for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
217                 {
218                     var ce = cb.value()?.Check(ph,pt);
219                     if (ce != null)
```

```
Name          Value                            Type
physicals     {(!0=Delete Record 137[23])}     Pyrrho.Common.BTree...
cx.rdC[23]    null                             Pyrrho.Level3.ReadCon...
```

Use the Watch window to examine **since**.



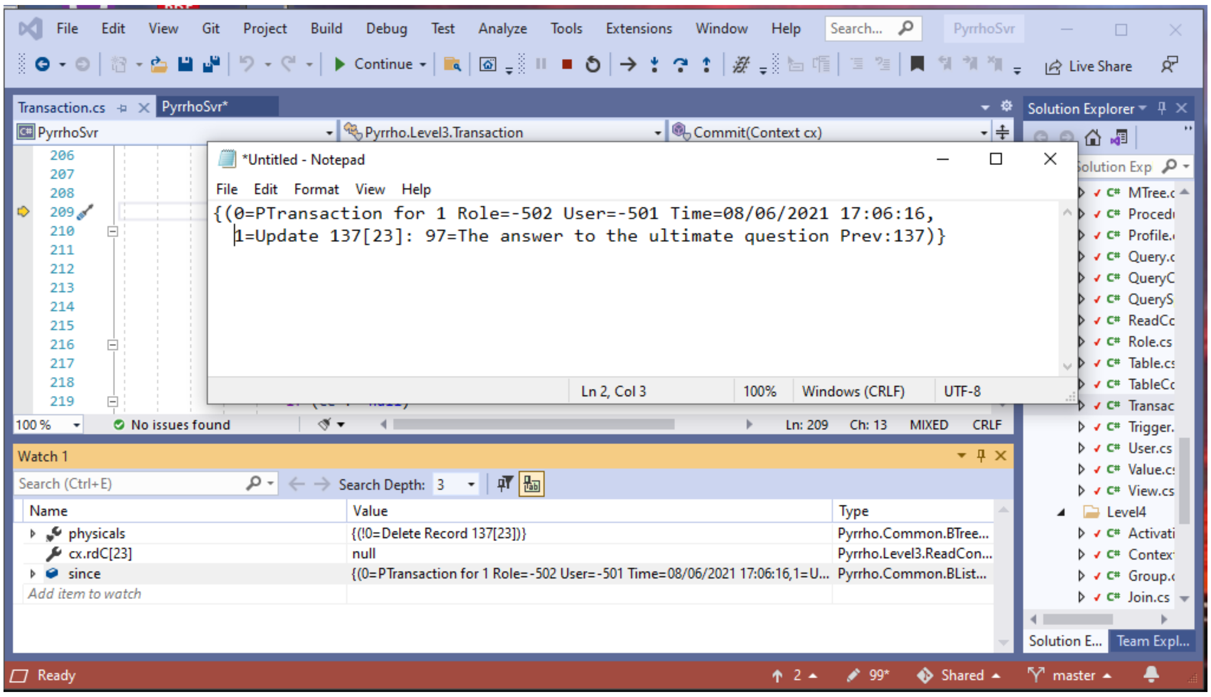
```
206            var wr = new Writer(new Context(db), dbfiles[name]);
207            var tb = physicals.First(); // start of the work we want to commit
208            var since = rdr.GetAll();
209            Physical ph = null;  ≤4ms elapsed
210            for (var pb=since.First(); pb!=null; pb=pb.Next())
211            {
212                ph = pb.value();
213                PTransaction pt = null;
214                if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
215                    pt = (PTransaction)ph;
216                for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
217                {
218                    var ce = cb.value()?.Check(ph,pt);
219                    if (ce != null)
```

Watch 1

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:06:16,1=U... | Pyrrho.Common.BList... |

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PyrrhoSvr
Continue
Live Share
Transaction.cs
Pyrrho.Level3.Transaction
Commit(Context cx)
No issues found
Ln: 209 Ch: 13 MIXED CRLF
Solution Explorer
Ready
master

Right-click and copy the value into a Notepad. Add a little extra white space to make it more readable.



*Untitled - Notepad

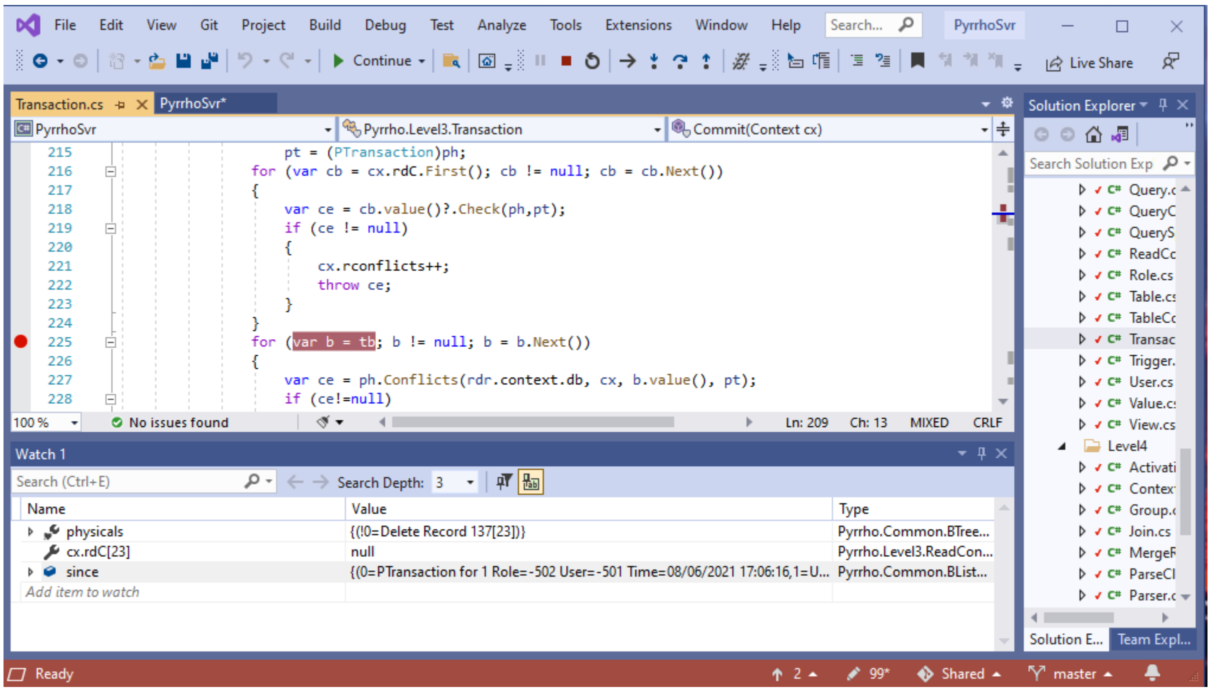
```
{(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:06:16,
 1=Update 137[23]: 97=The answer to the ultimate question Prev:137)}
```

Watch 1

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:06:16,1=U... | Pyrrho.Common.BList... |

We can see we have got the transaction marker and the update that the green window has made.

Set a breakpoint at line 225.

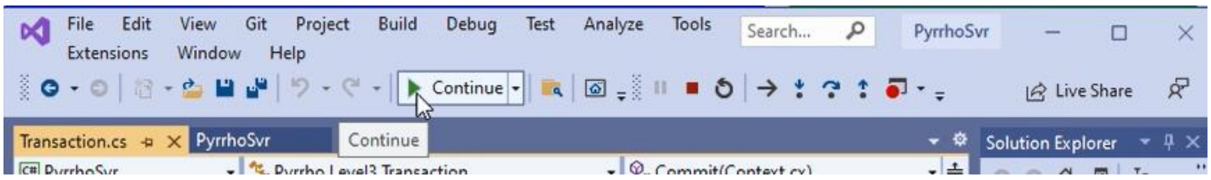


```
            pt = (PTransaction)ph;
            for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
            {
                var ce = cb.value()?.Check(ph,pt);
                if (ce != null)
                {
                    cx.rconflicts++;
                    throw ce;
                }
            }
            for (var b = tb; b != null; b = b.Next())
            {
                var ce = ph.Conflicts(rdr.context.db, cx, b.value(), pt);
                if (ce!=null)
```

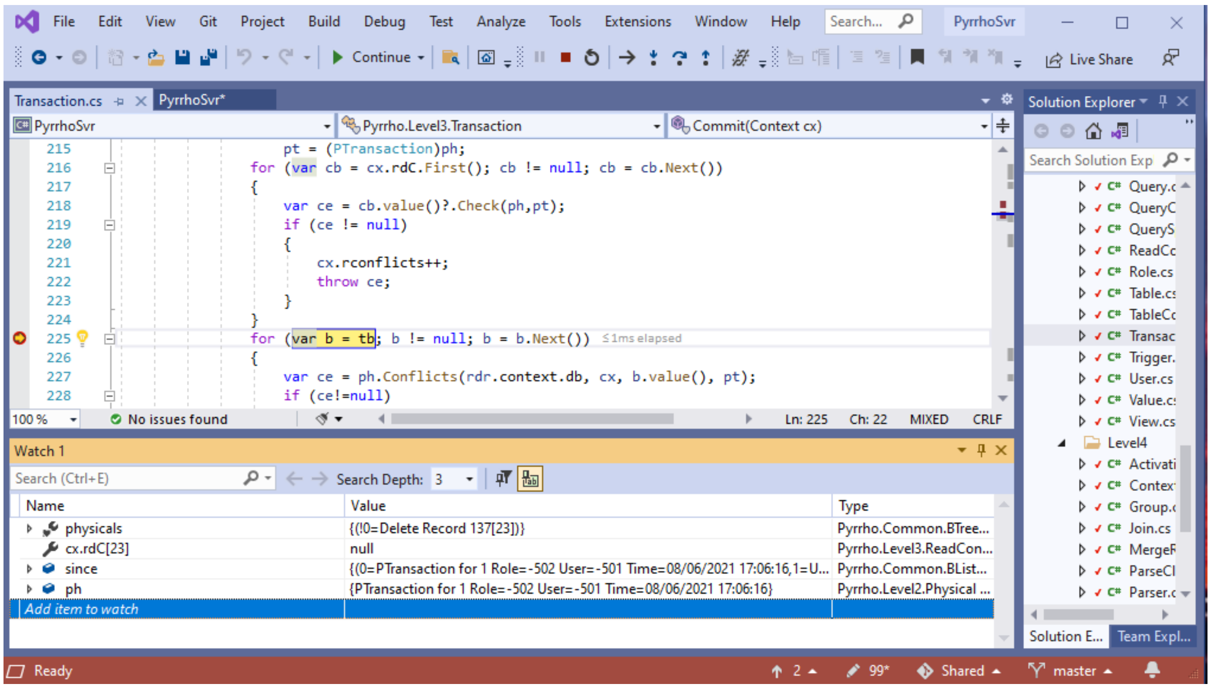
| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:06:16,1=U... | Pyrrho.Common.BList... |

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PyrrhoSvr
Continue
Live Share
Transaction.cs
Solution Explorer
Pyrrho.Level3.Transaction
Commit(Context cx)
No issues found
Ln: 209 Ch: 13 MIXED CRLF
Watch 1
Search Depth: 3
Ready
master

Click Continue



The first time we hit this particular line, ph is the transaction record, as we can see in the Watch window,

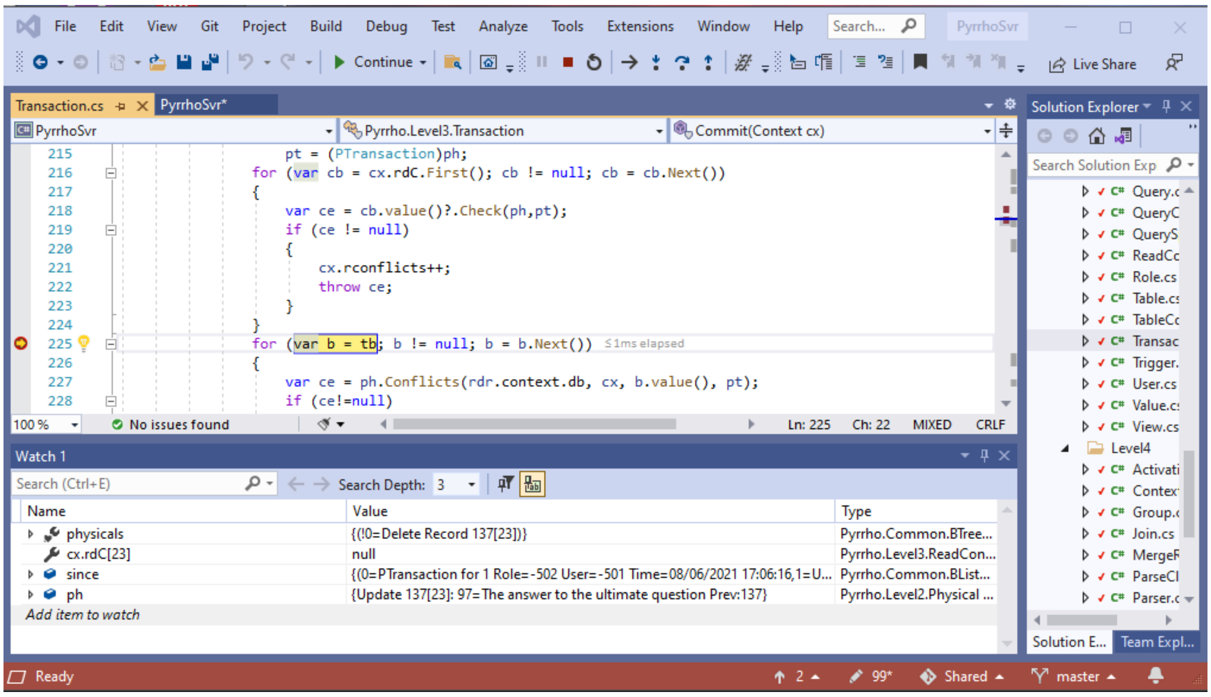


```
            pt = (PTransaction)ph;
            for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
            {
                var ce = cb.value()?.Check(ph,pt);
                if (ce != null)
                {
                    cx.rconflicts++;
                    throw ce;
                }
            }
            for (var b = tb; b != null; b = b.Next())  ≤1ms elapsed
            {
                var ce = ph.Conflicts(rdr.context.db, cx, b.value(), pt);
                if (ce!=null)
```

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:06:16,1=U... | Pyrrho.Common.BList... |
| ph | {PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:06:16} | Pyrrho.Level2.Physical ... |

which is not very interesting.

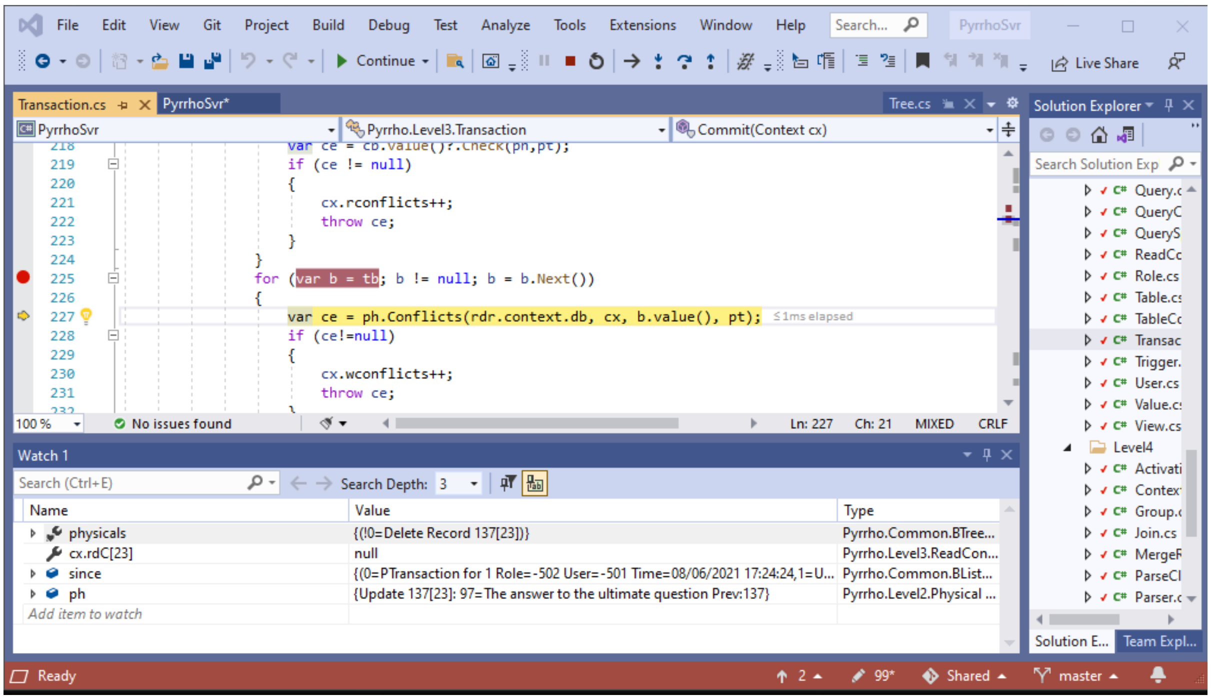
Click Continue



```
            pt = (PTransaction)ph;
        for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
        {
            var ce = cb.value()?.Check(ph,pt);
            if (ce != null)
            {
                cx.rconflicts++;
                throw ce;
            }
        }
        for (var b = tb; b != null; b = b.Next())  ≤1ms elapsed
        {
            var ce = ph.Conflicts(rdr.context.db, cx, b.value(), pt);
            if (ce!=null)
```

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:06:16,1=U... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The answer to the ultimate question Prev:137} | Pyrrho.Level2.Physical ... |

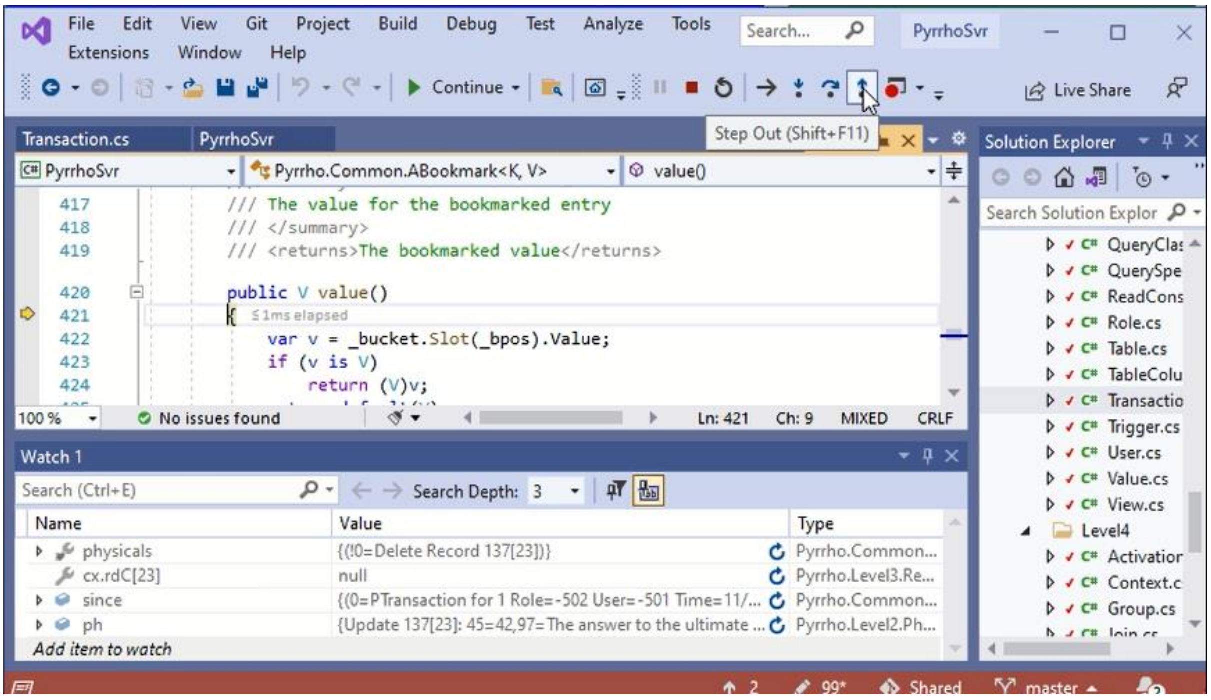
The second time, ph is the Update, and we Step over three times:



```
            var ce = cb.value()?.Check(ph,pt);
            if (ce != null)
            {
                cx.rconflicts++;
                throw ce;
            }
        }
        for (var b = tb; b != null; b = b.Next())
        {
            var ce = ph.Conflicts(rdr.context.db, cx, b.value(), pt);  ≤1ms elapsed
            if (ce!=null)
            {
                cx.wconflicts++;
                throw ce;
```

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:24:24,1=U... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The answer to the ultimate question Prev:137} | Pyrrho.Level2.Physical ... |

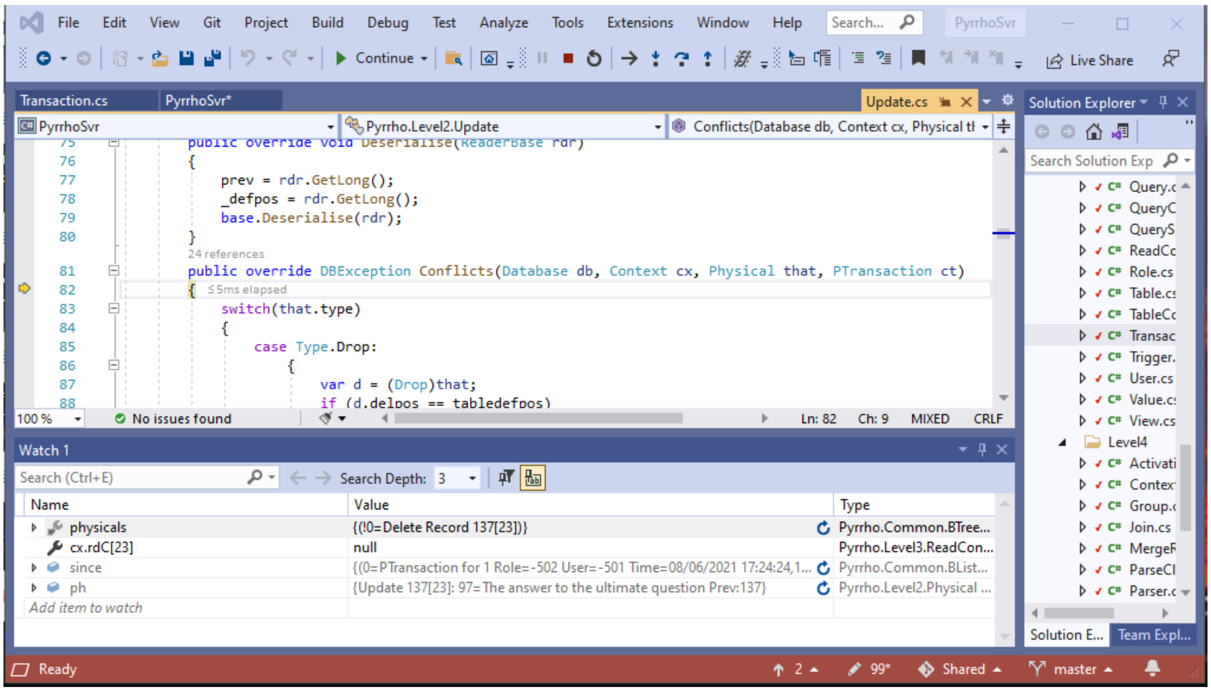
Click StepInto: the debugger stops during the evaluation of cb.value(),



```
        /// The value for the bookmarked entry
        /// </summary>
        /// <returns>The bookmarked value</returns>
        public V value()
        { ≤1ms elapsed
            var v = _bucket.Slot(_bpos).Value;
            if (v is V)
                return (V)v;
```

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common... |
| cx.rdC[23] | null | Pyrrho.Level3.Re... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=11/... | Pyrrho.Common... |
| ph | {Update 137[23]: 45=42,97=The answer to the ultimate ... | Pyrrho.Level2.Ph... |
| Add item to watch | | |

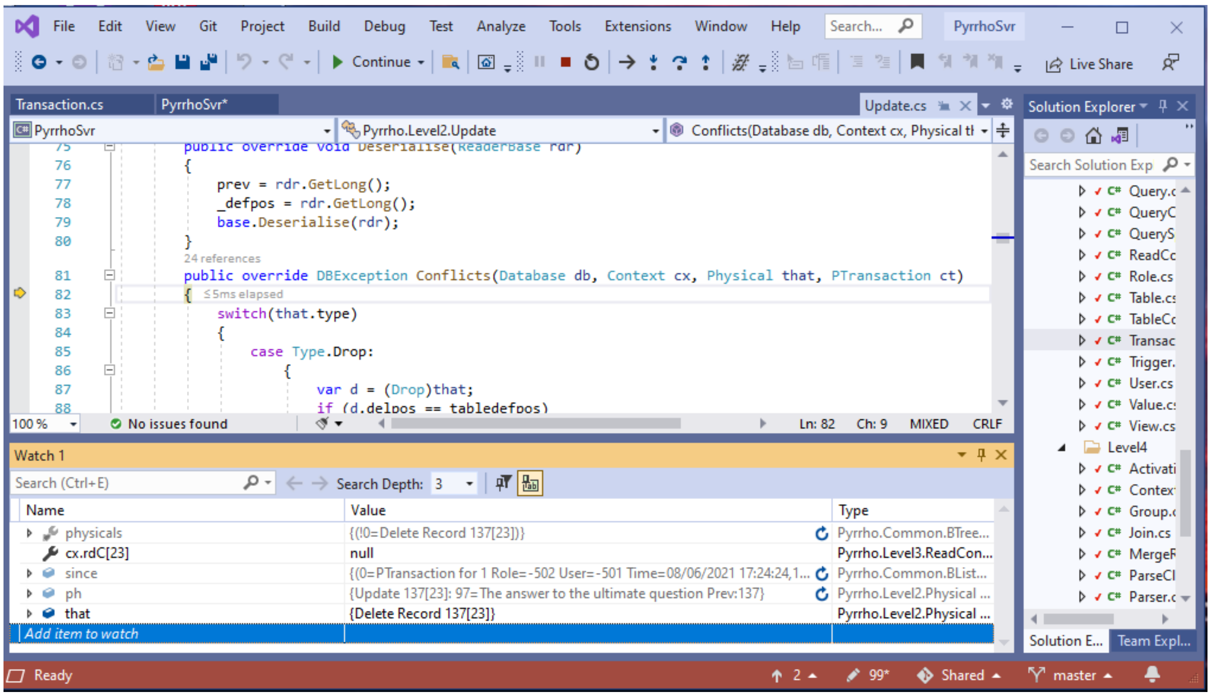
so Step Out, and them immediately Step Into:



```
        public override void Deserialise(ReaderBase rdr)
        {
            prev = rdr.GetLong();
            _defpos = rdr.GetLong();
            base.Deserialise(rdr);
        }
        24 references
        public override DBException Conflicts(Database db, Context cx, Physical that, PTransaction ct)
        { ≤5ms elapsed
            switch(that.type)
            {
                case Type.Drop:
                    {
                        var d = (Drop)that;
                        if (d.delpos == tabledefpos)
```

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:24:24,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The answer to the ultimate question Prev:137} | Pyrrho.Level2.Physical ... |
| Add item to watch | | |

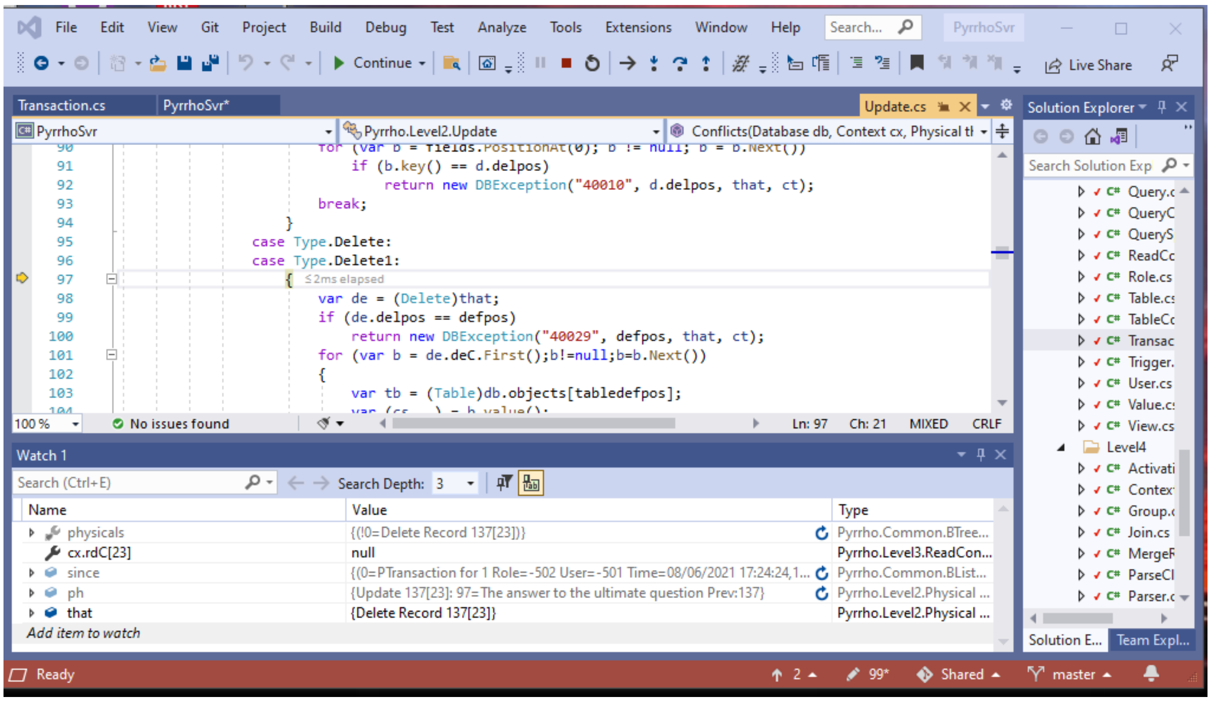
In Update.Conflicts() examine that in the Watch window: it is our Delete.



```
        public override void Deserialise(ReaderBase rdr)
        {
            prev = rdr.GetLong();
            _defpos = rdr.GetLong();
            base.Deserialise(rdr);
        }
        24 references
        public override DBException Conflicts(Database db, Context cx, Physical that, PTransaction ct)
        {  ≤5ms elapsed
            switch(that.type)
            {
                case Type.Drop:
                    {
                        var d = (Drop)that;
                        if (d.delpos == tabledefpos)
```

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:24:24,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The answer to the ultimate question Prev:137} | Pyrrho.Level2.Physical ... |
| that | {Delete Record 137[23]} | Pyrrho.Level2.Physical ... |

Step Over twice to see the case Delete:

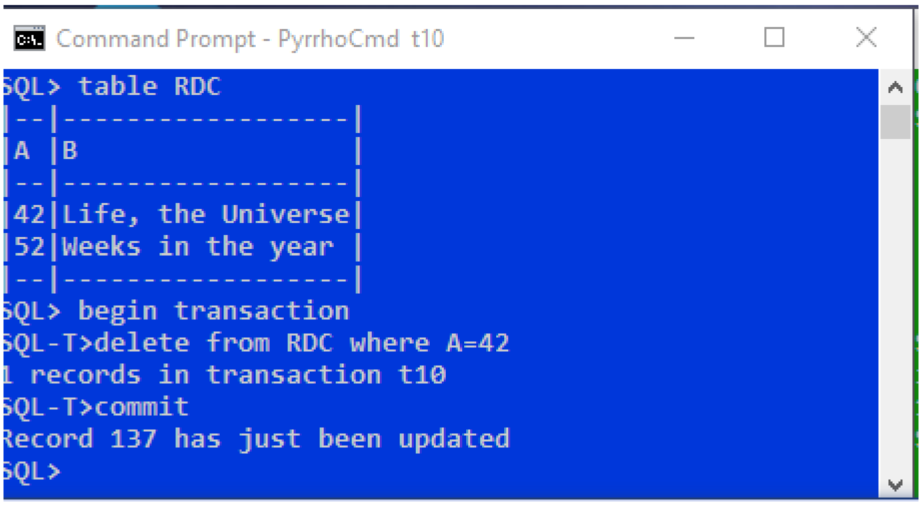


```
                        for (var b = fields.PositionAt(0); b != null; b = b.Next())
                            if (b.key() == d.delpos)
                                return new DBException("40010", d.delpos, that, ct);
                        break;
                    }
                case Type.Delete:
                case Type.Delete1:
                    {  ≤2ms elapsed
                        var de = (Delete)that;
                        if (de.delpos == defpos)
                            return new DBException("40029", defpos, that, ct);
                        for (var b = de.deC.First();b!=null;b=b.Next())
                        {
                            var tb = (Table)db.objects[tabledefpos];
                            var (cs, _) = b.value();
```

| Name | Value | Type |
|---|---|---|
| physicals | {(!0=Delete Record 137[23])} | Pyrrho.Common.BTree... |
| cx.rdC[23] | null | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:24:24,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The answer to the ultimate question Prev:137} | Pyrrho.Level2.Physical ... |
| that | {Delete Record 137[23]} | Pyrrho.Level2.Physical ... |

We can see that if the Delete’s delpos matches the Update’s defpos, there will be a conflict. (We are trying to delete a record that another transaction has updated.)

In this case they are both 137 and will conflict, so let’s just click Continue.



```
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> begin transaction
SQL-T>delete from RDC where A=42
1 records in transaction t10
SQL-T>commit
Record 137 has just been updated
SQL>
```

We see that we get a complaint back in the blue window that record 137 has just been updated. The transaction has been rolled back, as we see from the prompt.

That completes the first experiment that we want to do in this demonstration.

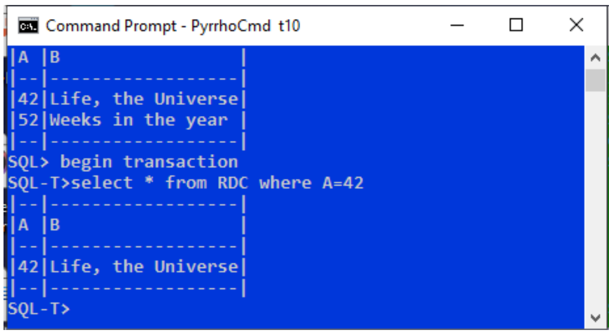
## Part 3: A Read-Write Transaction

Stop the server, delete the file t10 in \DATA, remove the breakpoints, and repeat the steps in part 1.

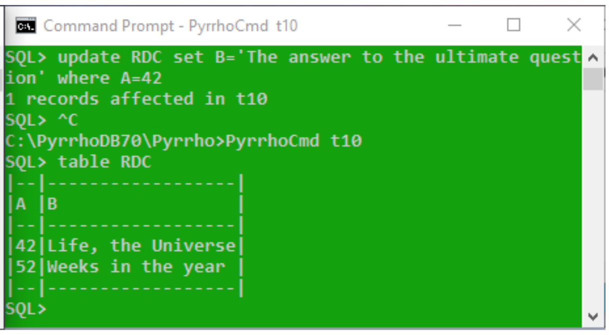
In the blue window

**`begin transaction`**

**`select * from RDC where A=42`**



```
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> begin transaction
SQL-T>select * from RDC where A=42
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|--|------------------|
SQL-T>
```

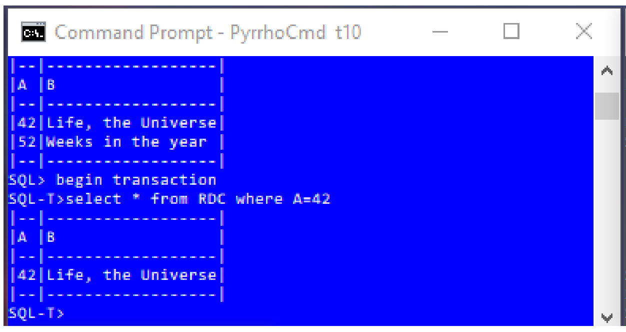


```
SQL> update RDC set B='The answer to the ultimate quest
ion' where A=42
1 records affected in t10
SQL> ^C
C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL>
```

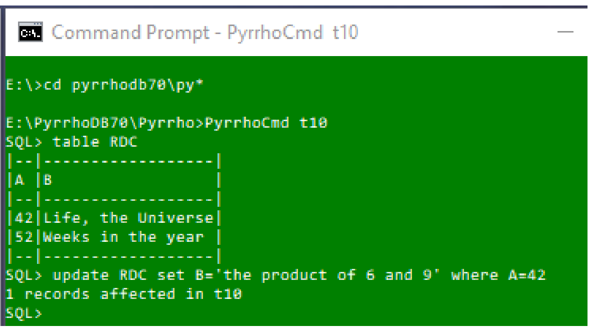
Just as before, the green window makes an update to the same row,

**`update RDC set B='The product of 6 and 9' where A=42`**

which is auto-committed. For the reasons explained earlier, we expect that the blue window will now be unable to commit.

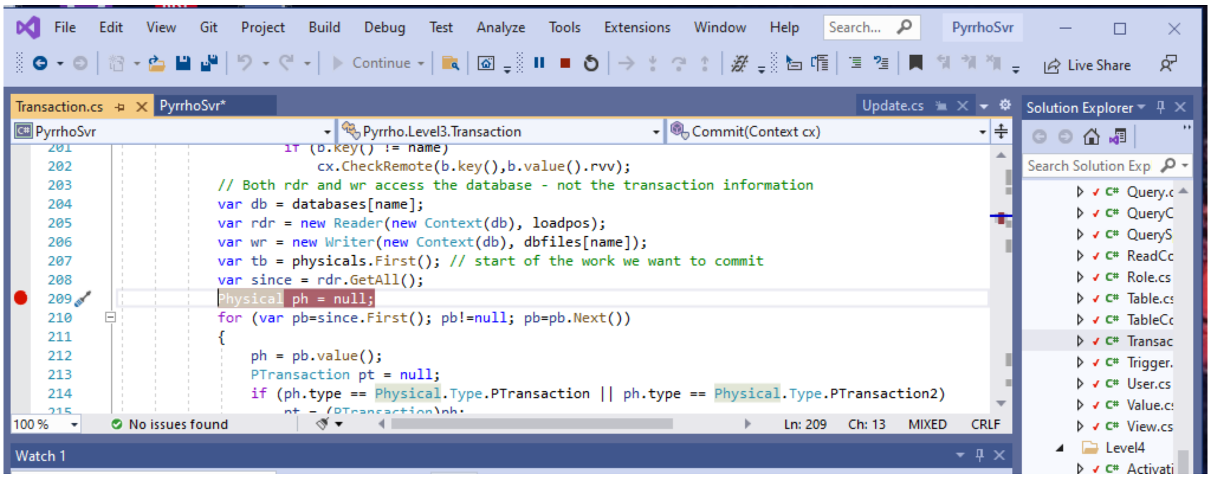


```
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> begin transaction
SQL-T>select * from RDC where A=42
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|--|------------------|
SQL-T>
```



```
E:\>cd pyrrhodb70\py*

E:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10
SQL> table RDC
|--|------------------|
|A |B                 |
|--|------------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> update RDC set B='the product of 6 and 9' where A=42
1 records affected in t10
SQL>
```

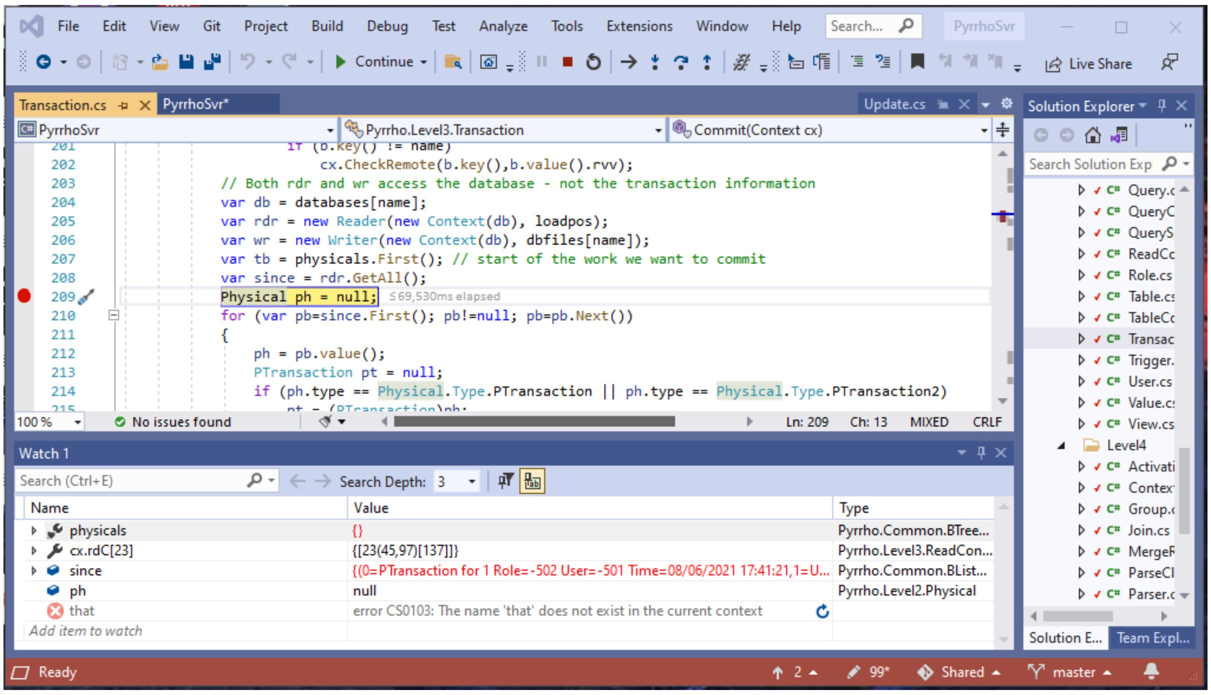
Again we stop in Transaction.Commit to see what happens. Place a breakpoint at line 209 of Transaction.cs:



```
201                 if (b.key() != name)
202                     cx.CheckRemote(b.key(),b.value().rvv);
203             // Both rdr and wr access the database - not the transaction information
204             var db = databases[name];
205             var rdr = new Reader(new Context(db), loadpos);
206             var wr = new Writer(new Context(db), dbfiles[name]);
207             var tb = physicals.First(); // start of the work we want to commit
208             var since = rdr.GetAll();
209             Physical ph = null;
210             for (var pb=since.First(); pb!=null; pb=pb.Next())
211             {
212                 ph = pb.value();
213                 PTransaction pt = null;
214                 if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
215                     pt = (PTransaction)ph;
```

and in the blue window

**`commit`**

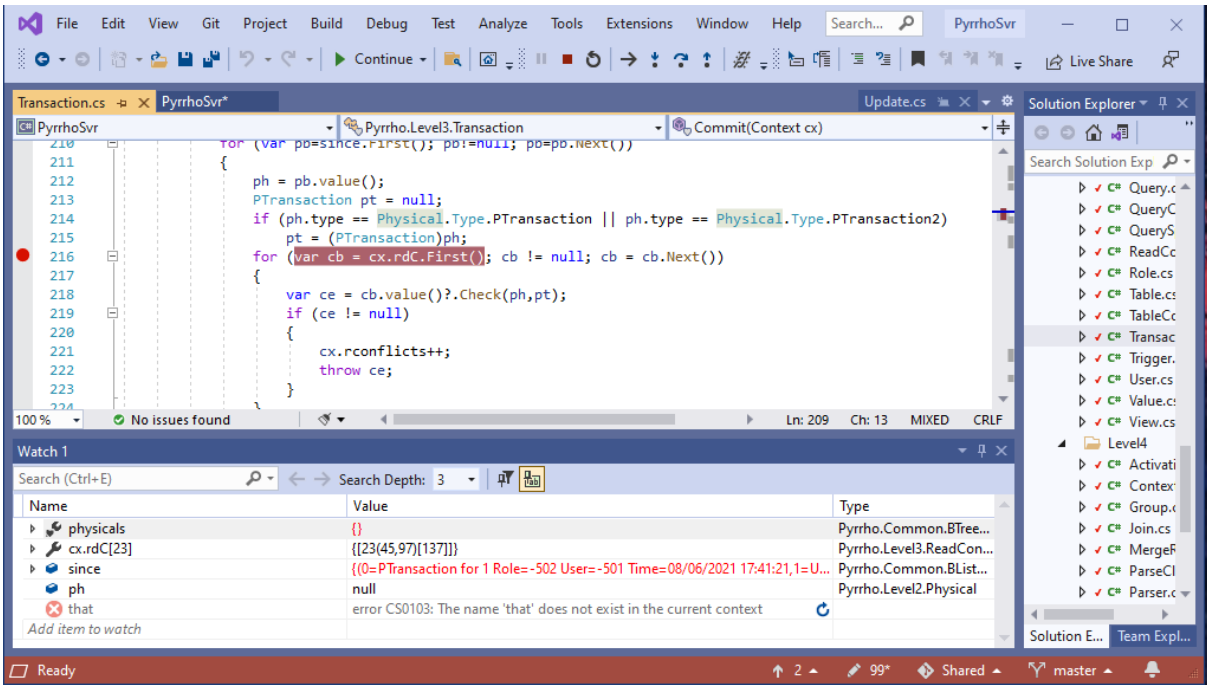


```
201                 if (b.key() != name)
202                     cx.CheckRemote(b.key(),b.value().rvv);
203             // Both rdr and wr access the database - not the transaction information
204             var db = databases[name];
205             var rdr = new Reader(new Context(db), loadpos);
206             var wr = new Writer(new Context(db), dbfiles[name]);
207             var tb = physicals.First(); // start of the work we want to commit
208             var since = rdr.GetAll();
209             Physical ph = null;  ≤69,530ms elapsed
210             for (var pb=since.First(); pb!=null; pb=pb.Next())
211             {
212                 ph = pb.value();
213                 PTransaction pt = null;
214                 if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
215                     pt = (PTransaction)ph;
```

Watch 1

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1=U... | Pyrrho.Common.BList... |
| ph | null | Pyrrho.Level2.Physical |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

This time, we see that physicals is empty, and instead we have a readConstraint. Set a breakpoint at line 216, to see the effect of this rdC:


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PyrrhoSvr
Continue
Live Share
Transaction.cs
PyrrhoSvr*
Update.cs
Solution Explorer
Pyrrho.Level3.Transaction
Commit(Context cx)
Search Solution Exp
Query.c QueryC QueryS ReadCc Role.cs Table.cs TableCc Transac Trigger. User.cs Value.cs View.cs
Level4
Activati Contex Group.c Join.cs MergeF ParseCl Parser.c
Solution E... Team Expl...
Ready
master

```
        for (var pb=since.First(); pb!=null; pb=pb.Next())
        {
            ph = pb.value();
            PTransaction pt = null;
            if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
                pt = (PTransaction)ph;
            for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
            {
                var ce = cb.value()?.Check(ph,pt);
                if (ce != null)
                {
                    cx.rconflicts++;
                    throw ce;
                }
```

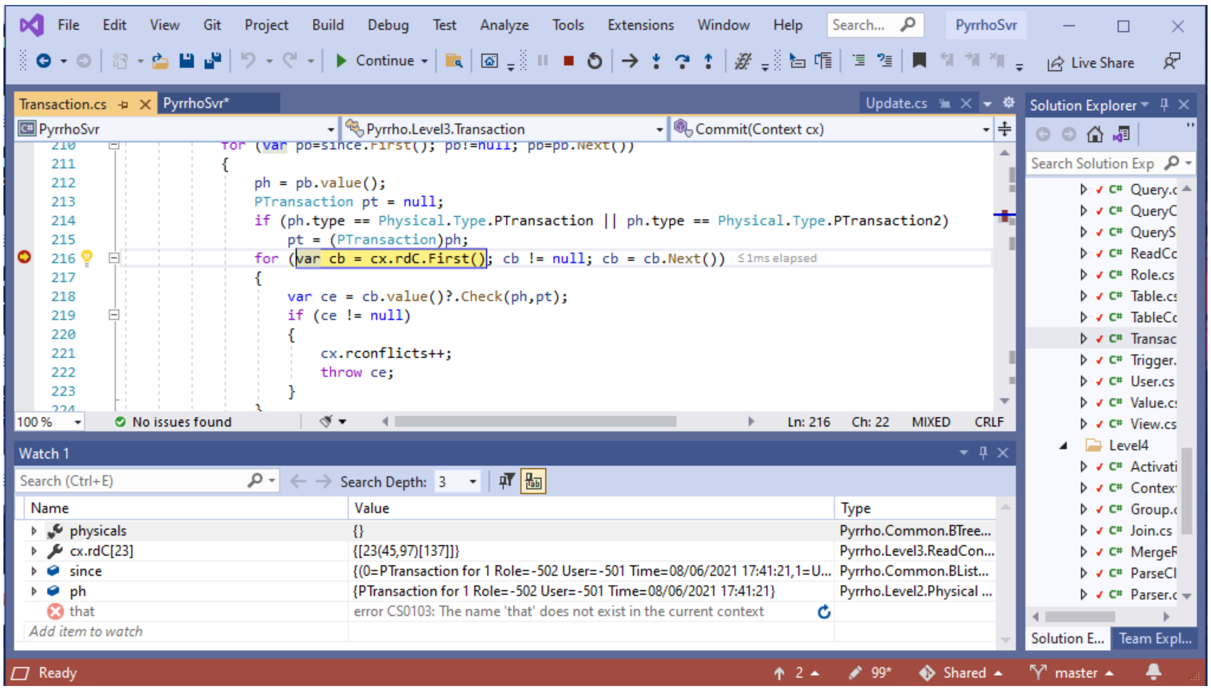
100 %  No issues found  Ln: 209  Ch: 13  MIXED  CRLF

Watch 1

Search (Ctrl+E)  Search Depth: 3

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1=U... | Pyrrho.Common.BList... |
| ph | null | Pyrrho.Level2.Physical |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

and click Continue, to reach this point.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PyrrhoSvr
Continue
Live Share
Transaction.cs
PyrrhoSvr*
Update.cs
Solution Explorer
Pyrrho.Level3.Transaction
Commit(Context cx)
Search Solution Exp
Query.c QueryC QueryS ReadCc Role.cs Table.cs TableCc Transac Trigger. User.cs Value.cs View.cs
Level4
Activati Contex Group.c Join.cs MergeF ParseCl Parser.c
Solution E... Team Expl...
Ready
master

```
        for (var pb=since.First(); pb!=null; pb=pb.Next())
        {
            ph = pb.value();
            PTransaction pt = null;
            if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
                pt = (PTransaction)ph;
            for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())  ≤1ms elapsed
            {
                var ce = cb.value()?.Check(ph,pt);
                if (ce != null)
                {
                    cx.rconflicts++;
                    throw ce;
                }
```

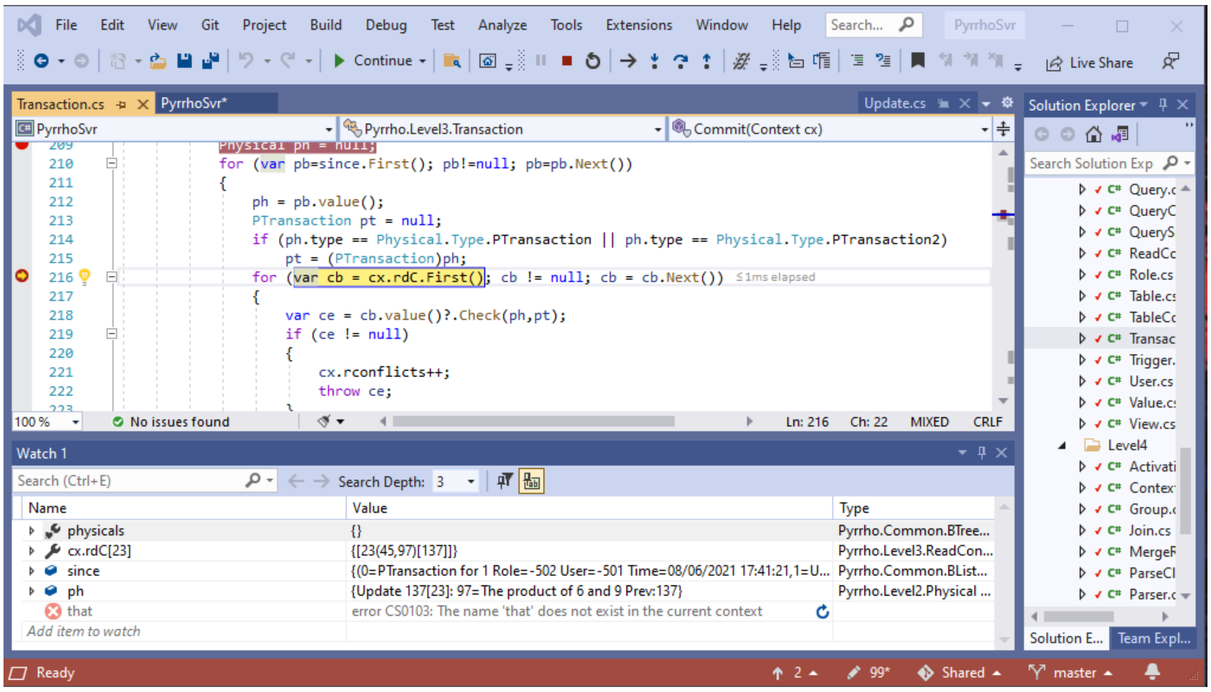
100 %  No issues found  Ln: 216  Ch: 22  MIXED  CRLF

Watch 1

Search (Ctrl+E)  Search Depth: 3

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1=U... | Pyrrho.Common.BList... |
| ph | {PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

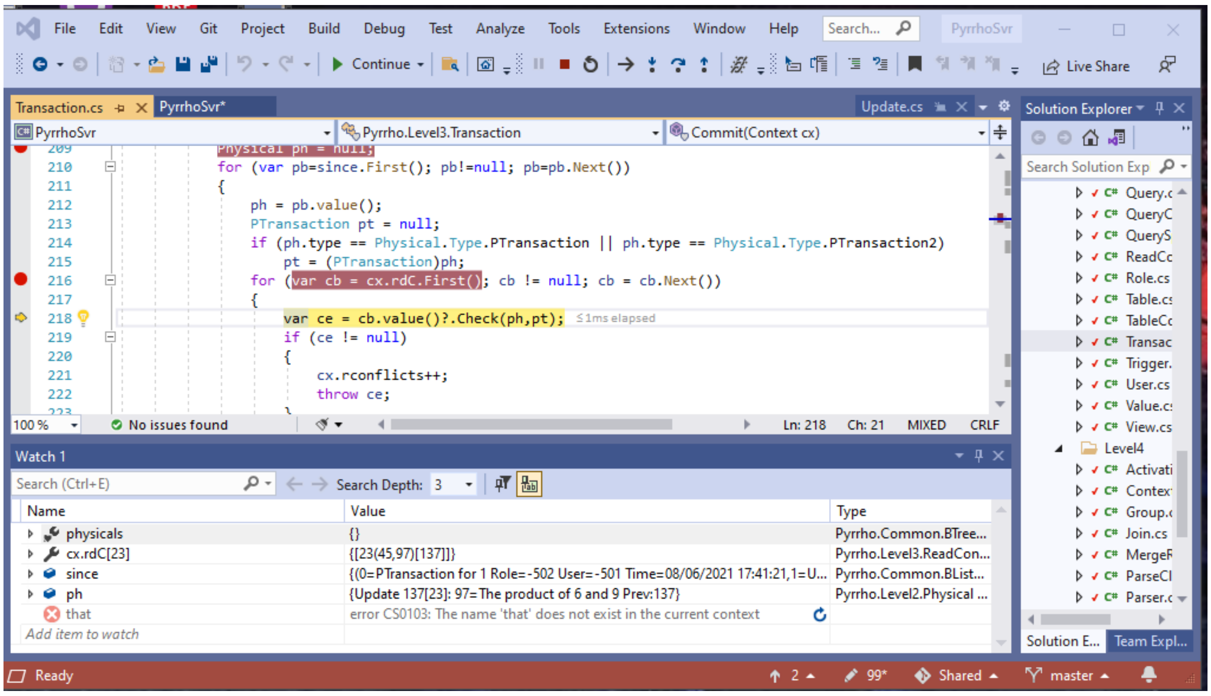
Again, we see that ph is the PTransaction, which is not interesting. Click Continue again:



```
209                 Physical ph = null;
210                 for (var pb=since.First(); pb!=null; pb=pb.Next())
211                 {
212                     ph = pb.value();
213                     PTransaction pt = null;
214                     if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
215                         pt = (PTransaction)ph;
216                     for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())  ≤1ms elapsed
217                     {
218                         var ce = cb.value()?.Check(ph,pt);
219                         if (ce != null)
220                         {
221                             cx.rconflicts++;
222                             throw ce;
```

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1=U... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

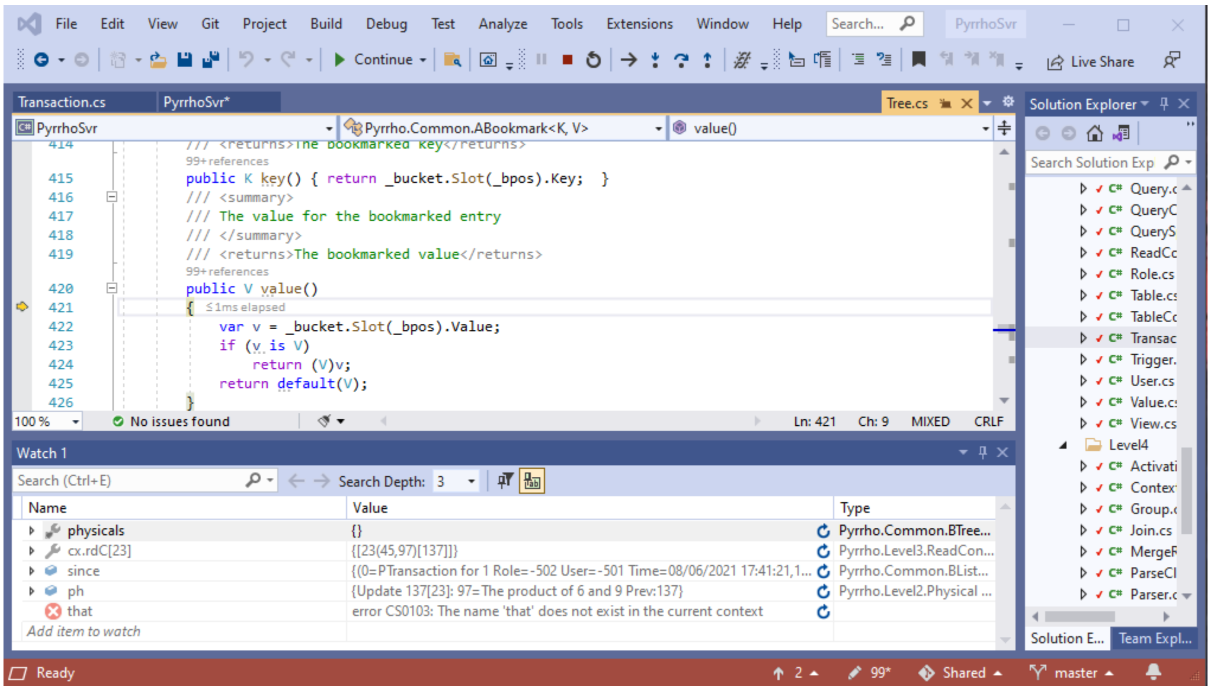
This time, ph is the Update. We want to Step into the Check method here. So Step Over to get to line 218.



```
209                 Physical ph = null;
210                 for (var pb=since.First(); pb!=null; pb=pb.Next())
211                 {
212                     ph = pb.value();
213                     PTransaction pt = null;
214                     if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
215                         pt = (PTransaction)ph;
216                     for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
217                     {
218                         var ce = cb.value()?.Check(ph,pt);  ≤1ms elapsed
219                         if (ce != null)
220                         {
221                             cx.rconflicts++;
222                             throw ce;
```

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1=U... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

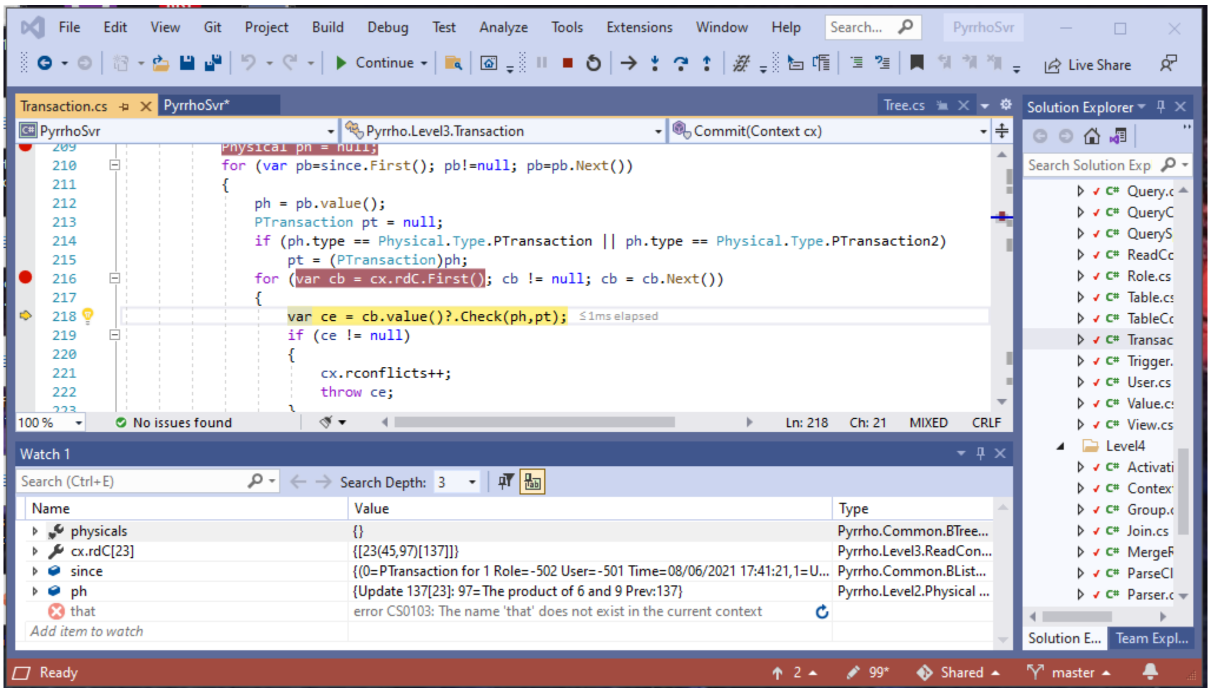
At line 218, Step Into, b.value():



```
/// <returns>The bookmarked key</returns>
99+ references
public K key() { return _bucket.Slot(_bpos).Key;  }
/// <summary>
/// The value for the bookmarked entry
/// </summary>
/// <returns>The bookmarked value</returns>
99+ references
public V value()
{  ≤1ms elapsed
    var v = _bucket.Slot(_bpos).Value;
    if (v is V)
        return (V)v;
    return default(V);
}
```

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

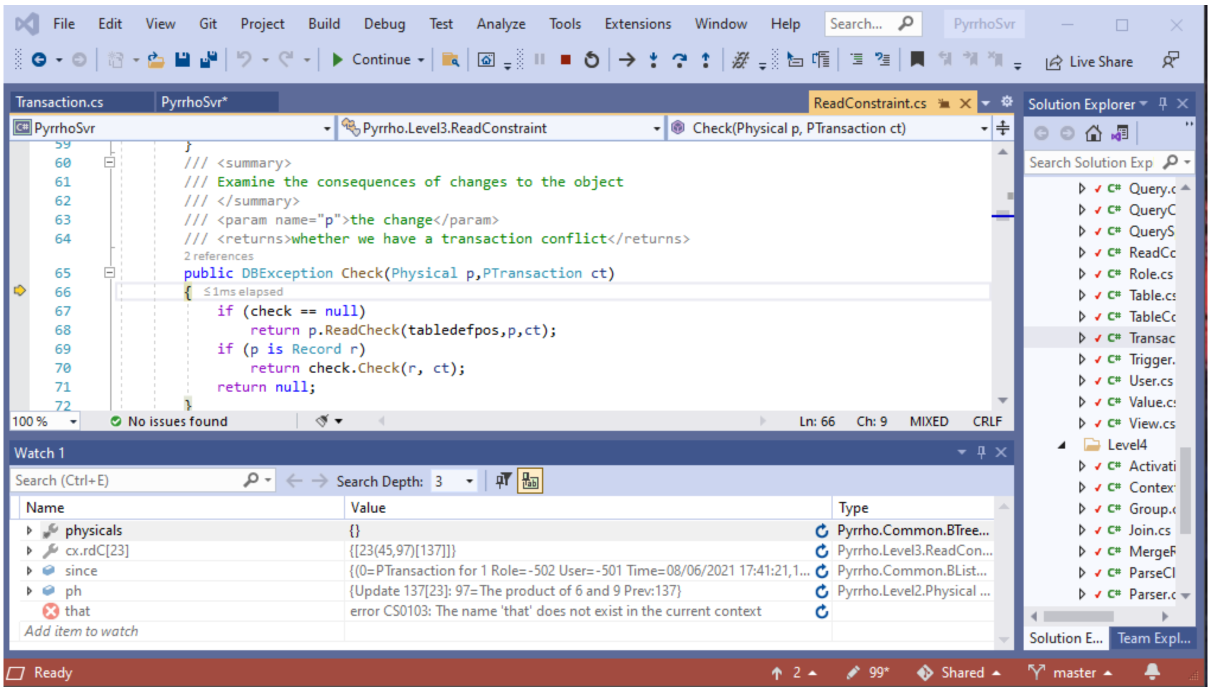
Step Out, (apparently back to where we were):



```
        Physical ph = null;
        for (var pb=since.First(); pb!=null; pb=pb.Next())
        {
            ph = pb.value();
            PTransaction pt = null;
            if (ph.type == Physical.Type.PTransaction || ph.type == Physical.Type.PTransaction2)
                pt = (PTransaction)ph;
            for (var cb = cx.rdC.First(); cb != null; cb = cb.Next())
            {
                var ce = cb.value()?.Check(ph,pt);  ≤1ms elapsed
                if (ce != null)
                {
                    cx.rconflicts++;
                    throw ce;
```

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1=U... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

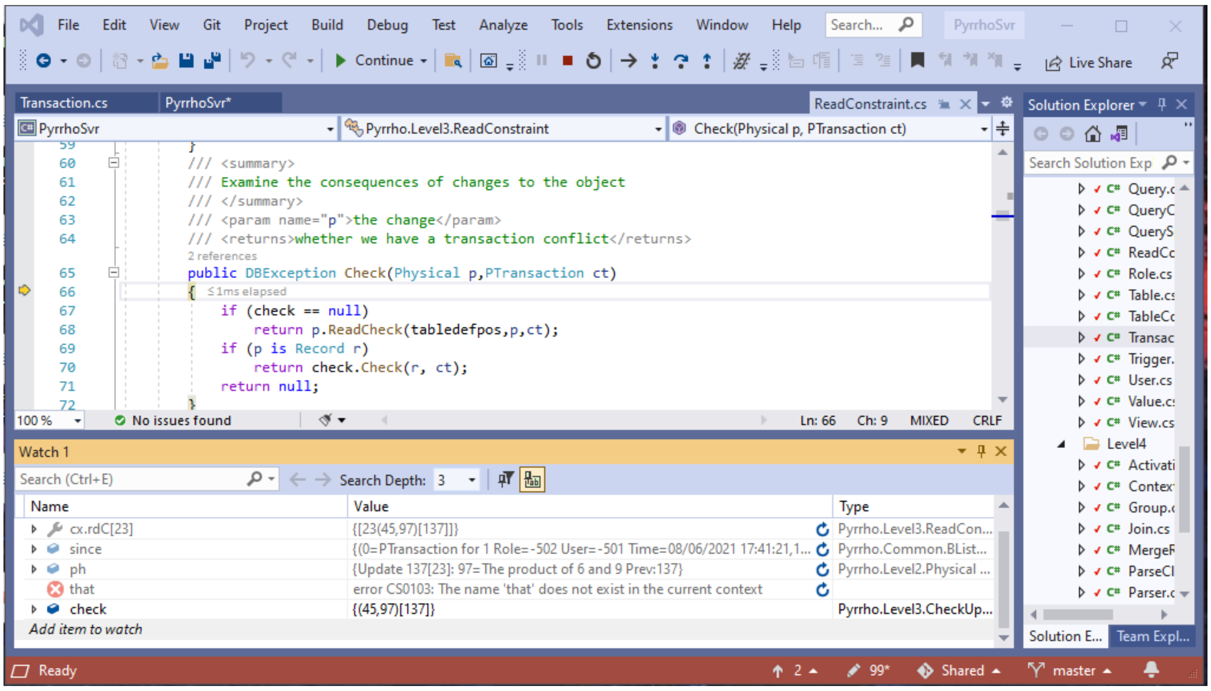
Step Into Check():



```
59          }
60          /// <summary>
61          /// Examine the consequences of changes to the object
62          /// </summary>
63          /// <param name="p">the change</param>
64          /// <returns>whether we have a transaction conflict</returns>
            2 references
65          public DBException Check(Physical p,PTransaction ct)
66          {  ≤1ms elapsed
67              if (check == null)
68                  return p.ReadCheck(tabledefpos,p,ct);
69              if (p is Record r)
70                  return check.Check(r, ct);
71              return null;
72          }
```

| Name | Value | Type |
|---|---|---|
| physicals | {} | Pyrrho.Common.BTree... |
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| Add item to watch | | |

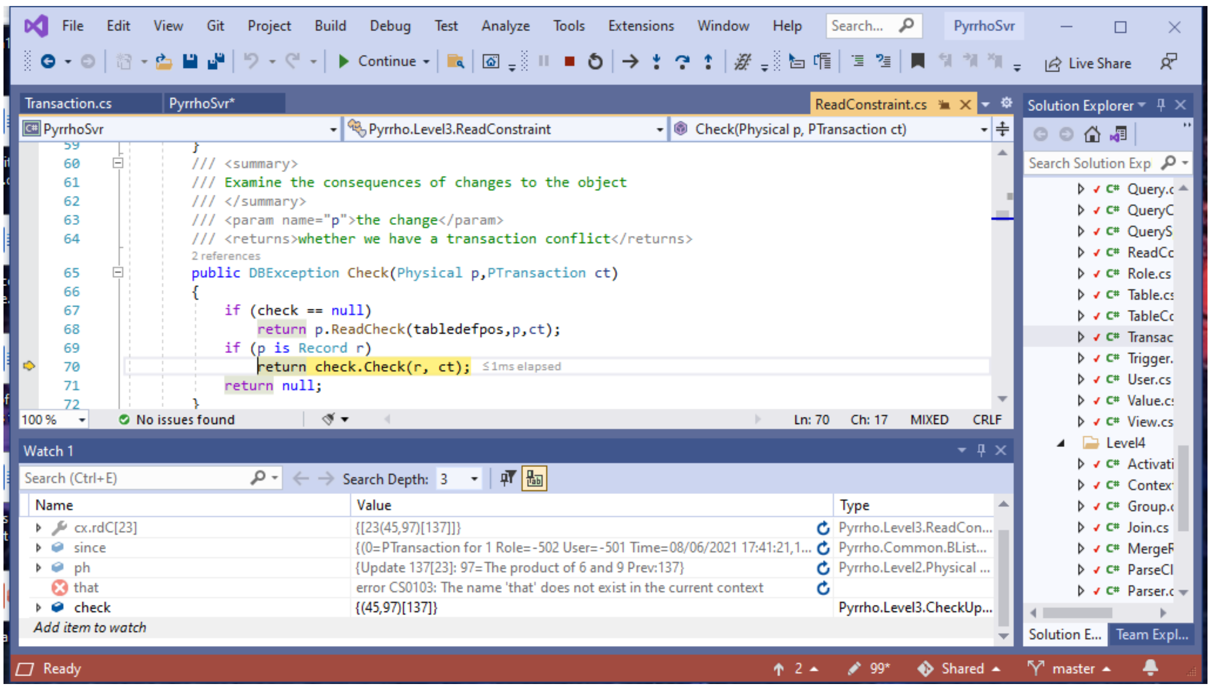
and examine check in the Watch window:



```
59          }
60          /// <summary>
61          /// Examine the consequences of changes to the object
62          /// </summary>
63          /// <param name="p">the change</param>
64          /// <returns>whether we have a transaction conflict</returns>
            2 references
65          public DBException Check(Physical p,PTransaction ct)
66          {  ≤1ms elapsed
67              if (check == null)
68                  return p.ReadCheck(tabledefpos,p,ct);
69              if (p is Record r)
70                  return check.Check(r, ct);
71              return null;
72          }
```

| Name | Value | Type |
|---|---|---|
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| check | {(45,97)[137]} | Pyrrho.Level3.CheckUp... |
| Add item to watch | | |

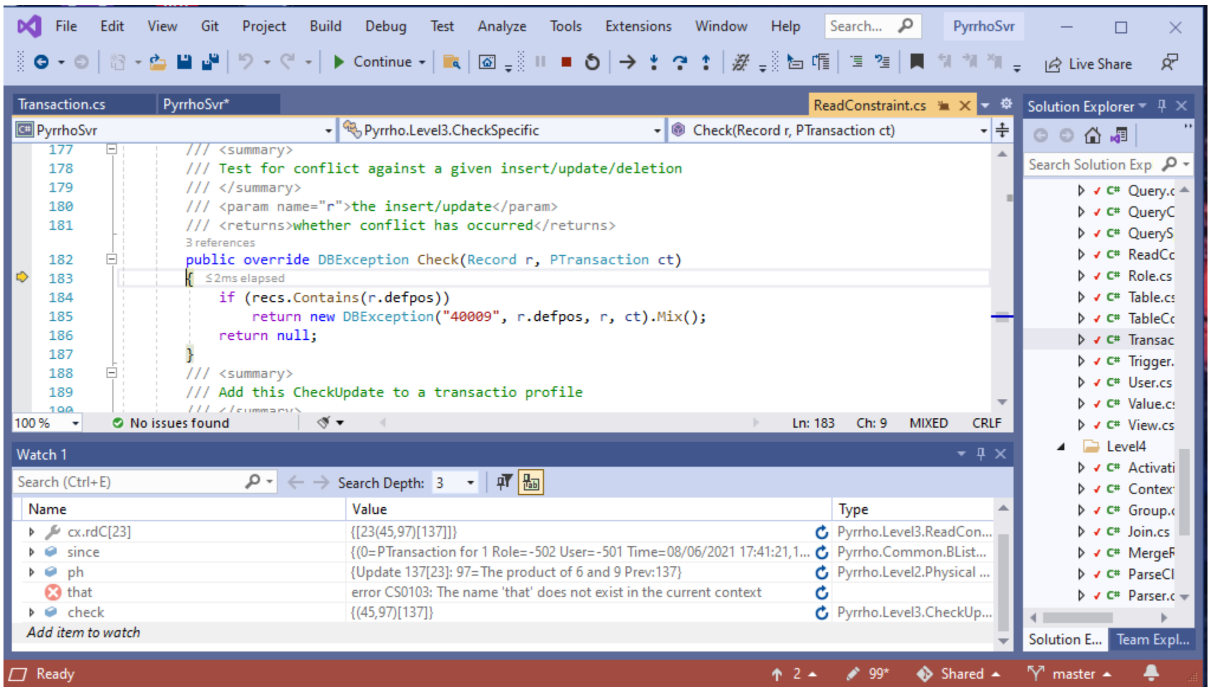
And p is the Update, which is a subclass of Record. Step Over to line 70:



```
        }
        /// <summary>
        /// Examine the consequences of changes to the object
        /// </summary>
        /// <param name="p">the change</param>
        /// <returns>whether we have a transaction conflict</returns>
        2 references
        public DBException Check(Physical p,PTransaction ct)
        {
            if (check == null)
                return p.ReadCheck(tabledefpos,p,ct);
            if (p is Record r)
                return check.Check(r, ct);  ≤1ms elapsed
            return null;
        }
```

| Name | Value | Type |
|---|---|---|
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| check | {(45,97)[137]} | Pyrrho.Level3.CheckUp... |
| Add item to watch | | |

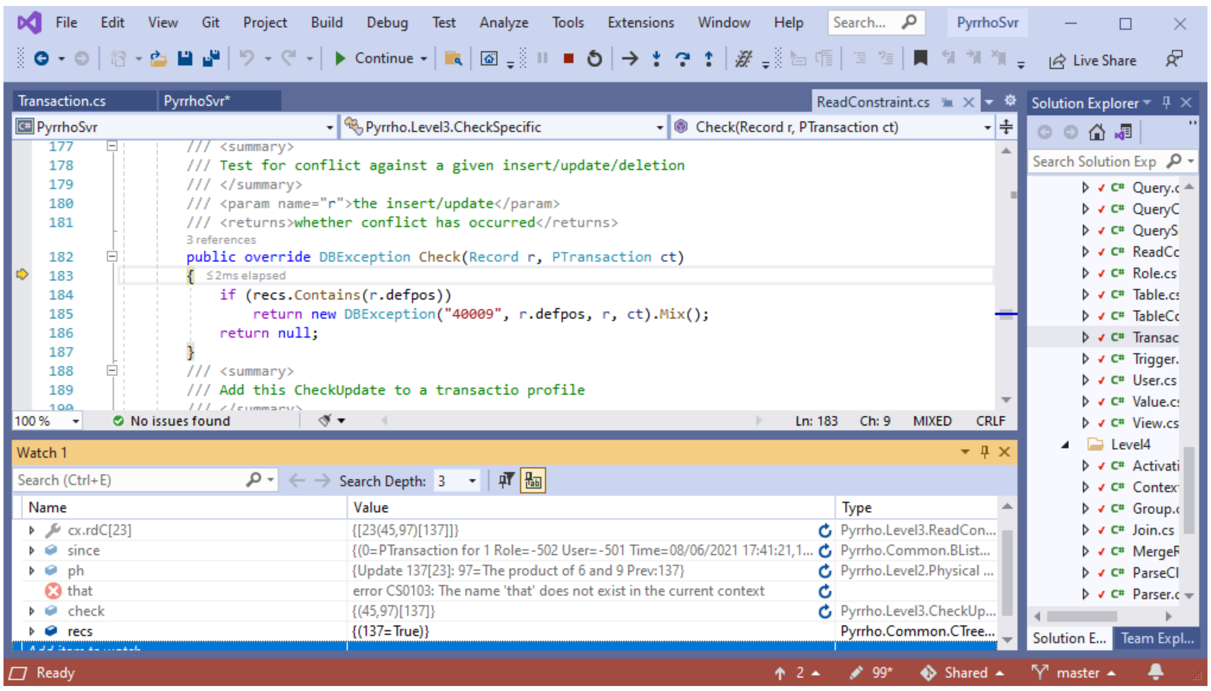
and Step Into check.Check():



```
        /// <summary>
        /// Test for conflict against a given insert/update/deletion
        /// </summary>
        /// <param name="r">the insert/update</param>
        /// <returns>whether conflict has occurred</returns>
        3 references
        public override DBException Check(Record r, PTransaction ct)
        {  ≤2ms elapsed
            if (recs.Contains(r.defpos))
                return new DBException("40009", r.defpos, r, ct).Mix();
            return null;
        }
        /// <summary>
        /// Add this CheckUpdate to a transactio profile
```

| Name | Value | Type |
|---|---|---|
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| check | {(45,97)[137]} | Pyrrho.Level3.CheckUp... |
| Add item to watch | | |

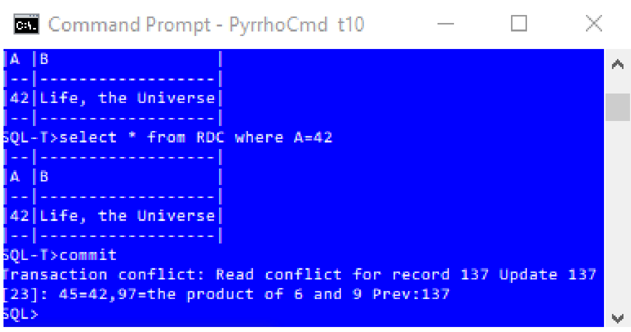
Examine recs in the Watch window:

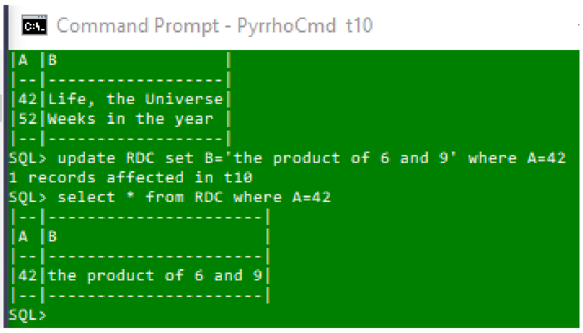

```
    /// <summary>
    /// Test for conflict against a given insert/update/deletion
    /// </summary>
    /// <param name="r">the insert/update</param>
    /// <returns>whether conflict has occurred</returns>
    3 references
    public override DBException Check(Record r, PTransaction ct)
    {  ≤2ms elapsed
        if (recs.Contains(r.defpos))
            return new DBException("40009", r.defpos, r, ct).Mix();
        return null;
    }
    /// <summary>
    /// Add this CheckUpdate to a transactio profile
```

Watch 1

| Name | Value | Type |
|---|---|---|
| cx.rdC[23] | {[23(45,97)[137]]} | Pyrrho.Level3.ReadCon... |
| since | {(0=PTransaction for 1 Role=-502 User=-501 Time=08/06/2021 17:41:21,1... | Pyrrho.Common.BList... |
| ph | {Update 137[23]: 97=The product of 6 and 9 Prev:137} | Pyrrho.Level2.Physical ... |
| that | error CS0103: The name 'that' does not exist in the current context | |
| check | {(45,97)[137]} | Pyrrho.Level3.CheckUp... |
| recs | {(137=True)} | Pyrrho.Common.CTree... |

recs contains defpos (that is, the row being updated is one we read), so there is a conflict. Click Continue.


Command Prompt - PyrrhoCmd t10
```
|A |B                |
|--|-----------------|
|42|Life, the Universe|
|--|-----------------|
SQL-T>select * from RDC where A=42
|--|-----------------|
|A |B                |
|--|-----------------|
|42|Life, the Universe|
|--|-----------------|
SQL-T>commit
Transaction conflict: Read conflict for record 137 Update 137
[23]: 45=42,97=the product of 6 and 9 Prev:137
SQL>
```

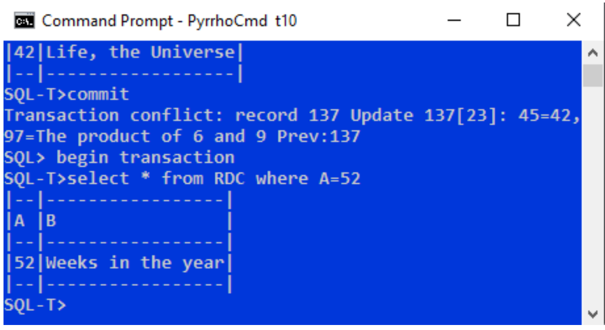

Command Prompt - PyrrhoCmd t10
```
|A |B                |
|--|-----------------|
|42|Life, the Universe|
|52|Weeks in the year |
|--|-----------------|
SQL> update RDC set B='the product of 6 and 9' where A=42
1 records affected in t10
SQL> select * from RDC where A=42
|--|---------------------|
|A |B                    |
|--|---------------------|
|42|the product of 6 and 9|
|--|---------------------|
SQL>
```

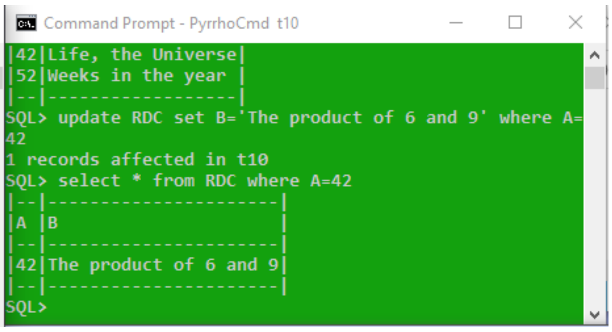
This concludes the second experiment.

As an extra (optional) experiment, let us show that read-write conflicts operate at the row level by using different rows. We can continue in the current state, or repeat the setting up stage. In the blue window:

**`begin transaction`**

**`select * from RDC where A=52`**

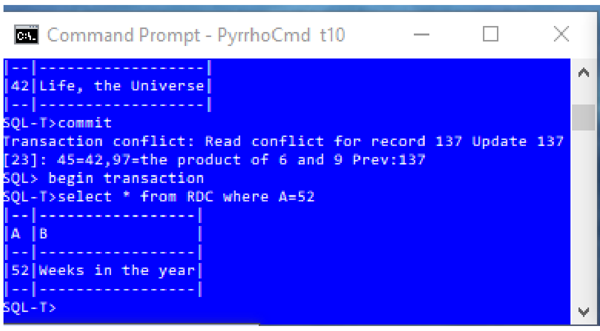

Command Prompt - PyrrhoCmd t10
```
|42|Life, the Universe|
|--|-----------------|
SQL-T>commit
Transaction conflict: record 137 Update 137[23]: 45=42,
97=The product of 6 and 9 Prev:137
SQL> begin transaction
SQL-T>select * from RDC where A=52
|--|-----------------|
|A |B                |
|--|-----------------|
|52|Weeks in the year|
|--|-----------------|
SQL-T>
```


Command Prompt - PyrrhoCmd t10
```
|42|Life, the Universe|
|52|Weeks in the year |
|--|-----------------|
SQL> update RDC set B='The product of 6 and 9' where A=
42
1 records affected in t10
SQL> select * from RDC where A=42
|--|---------------------|
|A |B                    |
|--|---------------------|
|42|The product of 6 and 9|
|--|---------------------|
SQL>
```

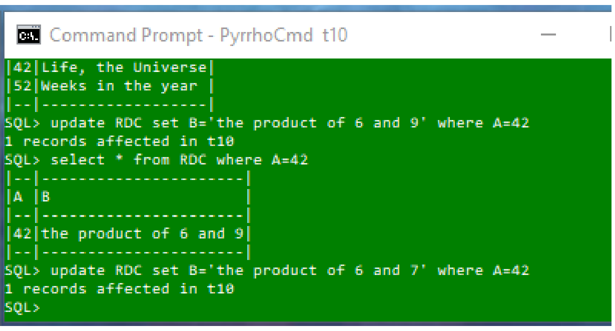
and in the green window:

```
update RDC set B='the product of 6 and 7' where A=42
```

If you still have the breakpoint in Transaction.Commit, just Continue



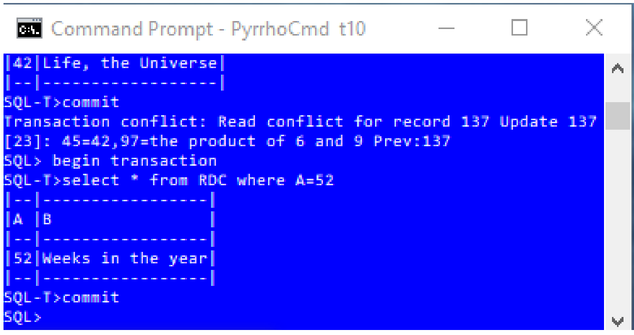
```
|--|------------------|
|42|Life, the Universe|
|--|------------------|
SQL-T>commit
Transaction conflict: Read conflict for record 137 Update 137
[23]: 45=42,97=the product of 6 and 9 Prev:137
SQL> begin transaction
SQL-T>select * from RDC where A=52
|--|-----------------|
|A |B                |
|--|-----------------|
|52|Weeks in the year|
|--|-----------------|
SQL-T>
```



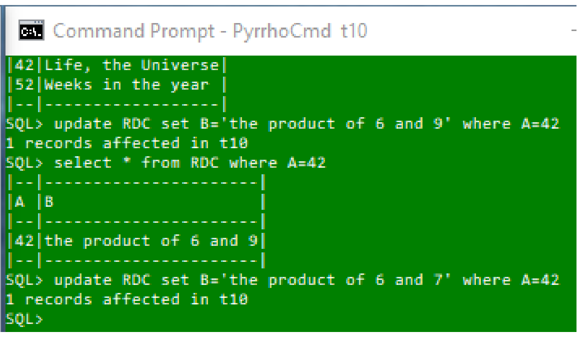
```
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> update RDC set B='the product of 6 and 9' where A=42
1 records affected in t10
SQL> select * from RDC where A=42
|--|----------------------|
|A |B                     |
|--|----------------------|
|42|the product of 6 and 9|
|--|----------------------|
SQL> update RDC set B='the product of 6 and 7' where A=42
1 records affected in t10
SQL>
```

Now in the blue window everything is fine on commit:

```
commit
```



```
|42|Life, the Universe|
|--|------------------|
SQL-T>commit
Transaction conflict: Read conflict for record 137 Update 137
[23]: 45=42,97=the product of 6 and 9 Prev:137
SQL> begin transaction
SQL-T>select * from RDC where A=52
|--|-----------------|
|A |B                |
|--|-----------------|
|52|Weeks in the year|
|--|-----------------|
SQL-T>commit
SQL>
```



```
|42|Life, the Universe|
|52|Weeks in the year |
|--|------------------|
SQL> update RDC set B='the product of 6 and 9' where A=42
1 records affected in t10
SQL> select * from RDC where A=42
|--|----------------------|
|A |B                     |
|--|----------------------|
|42|the product of 6 and 9|
|--|----------------------|
SQL> update RDC set B='the product of 6 and 7' where A=42
1 records affected in t10
SQL>
```