NEO Community TechNote

How to run NPCdApp 1.0

NEO Testnet: 0x7074acf3f06dd3f456e11053ebf61c5b04b07ebc

Submission to the CoZ dApp Competition #2
Closing date: February 24, 2018

Michael Herman

https://github.com/mwherman2000/neo-persistibleclasses

mwherman@parallelspace.net

NEO
smart economy

What is NPCdApp 1.0?

- NPC
  - NEO Persistable Classes
  - Long name: NEO Persistable Class (NPC) Framework 1.0
  - Byline: An Efficient Object-Oriented Framework for C#.NEO Smart Contract Development

- NPCdApp 1.0 is a C#.NEO smart contract dApp built for the CoZ dApp Competition #2 (closing date: February 24, 2018) to demonstrate NPC's
  - Unique object-oriented programming style for C#.NEO smart contract development
  - Layered entity persistence model

Unique Object-Oriented Programming Style

```
Point p4 = Point.New();
Point.Set(p4, 10, 20);
Point.Log("p4", p4);

for (int index = 30; index < 40; index++)
{
    NeoTrace.Trace("index", index);
    Point.Set(p4, index, -index);
    Point.PutElement(p4, vau, index);
}

for (int index = 30; index < 40; index++)
{
    Point.Set(p4, index, -index);
    Point x = Point.GetElement(vau, index);
    Point.Log("loop.x", x);
    if (Point.GetX(p4) != index || Point.GetY(p4) != -index)
    {
        msg = ">>>>(x,y) are different";
        NeoTrace.Trace(msg);
        break;
    }
}
```

Layered Entity Persistence Model

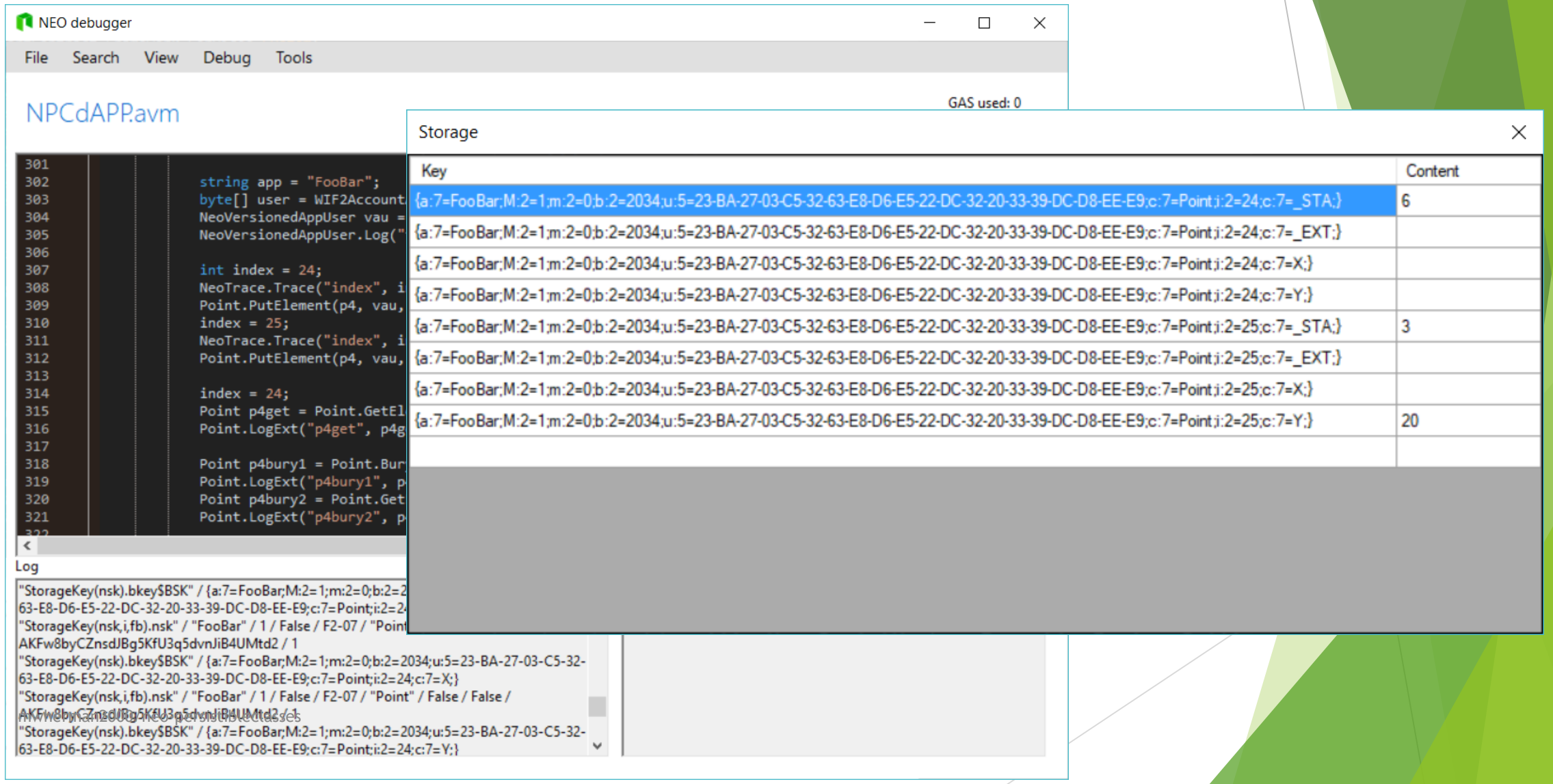
```
Point p4 = Point.New();
Point.Set(p4, 10, 20);
Point.Log("p4", p4);

for (int index = 30; index < 40; index++)
{
    NeoTrace.Trace("index", index);
    Point.Set(p4, index, -index);
    Point.PutElement(p4, vau, index);
}

for (int index = 30; index < 40; index++)
{
    Point.Set(p4, index, -index);
    Point x = Point.GetElement(vau, index);
    Point.Log("loop.x", x);
    if (Point.GetX(p4) != index || Point.GetY(p4) != -index)
    {
        msg = ">>>>(x,y) are different";
        NeoTrace.Trace(msg);
        break;
    }
}
```

See Appendix A of this PPT for more C#.NEO object-oriented programming examples

Running NPCdApp 1.0 with **neo-debugger**

- Source to build NPCdApp.avm for loading into neod.exe:
  - https://github.com/mwherman2000/neo-persistibleclasses/blob/master/NPCdApp/NPCdApp.cs
- Use latest version of neo-debugger with new Storage viewer (F6) smart formatting support
  - Should be in https://github.com/CityOfZion/neo-debugger-tools/pull/48
  - Backup plan: https://github.com/mwherman2000/neo-debugger-tools
- This version includes specific smart formatting support for NEO Storage Keys that in implement the new NEO-KONG NEO Storage Key format
  - NEO-KONG = NEO Key-Object Notation for Geeks
- **Calling Conventions**
  - Prototype: Main(string operation, param objects[])
  - Operations: “test1” | “test2” | “test3” | “test4” | “test5” | “test6”
  - ArrayOfObjects: [5] – only used by “test6” to control the maximum number of iterations
- Results
  - Main will return “success”
  - The real results are available by pressing F6 to display the neo-debugger Storage viewer
  - The real results also include the messages in the Event Log panel (lower-left corner of neo-debugger)

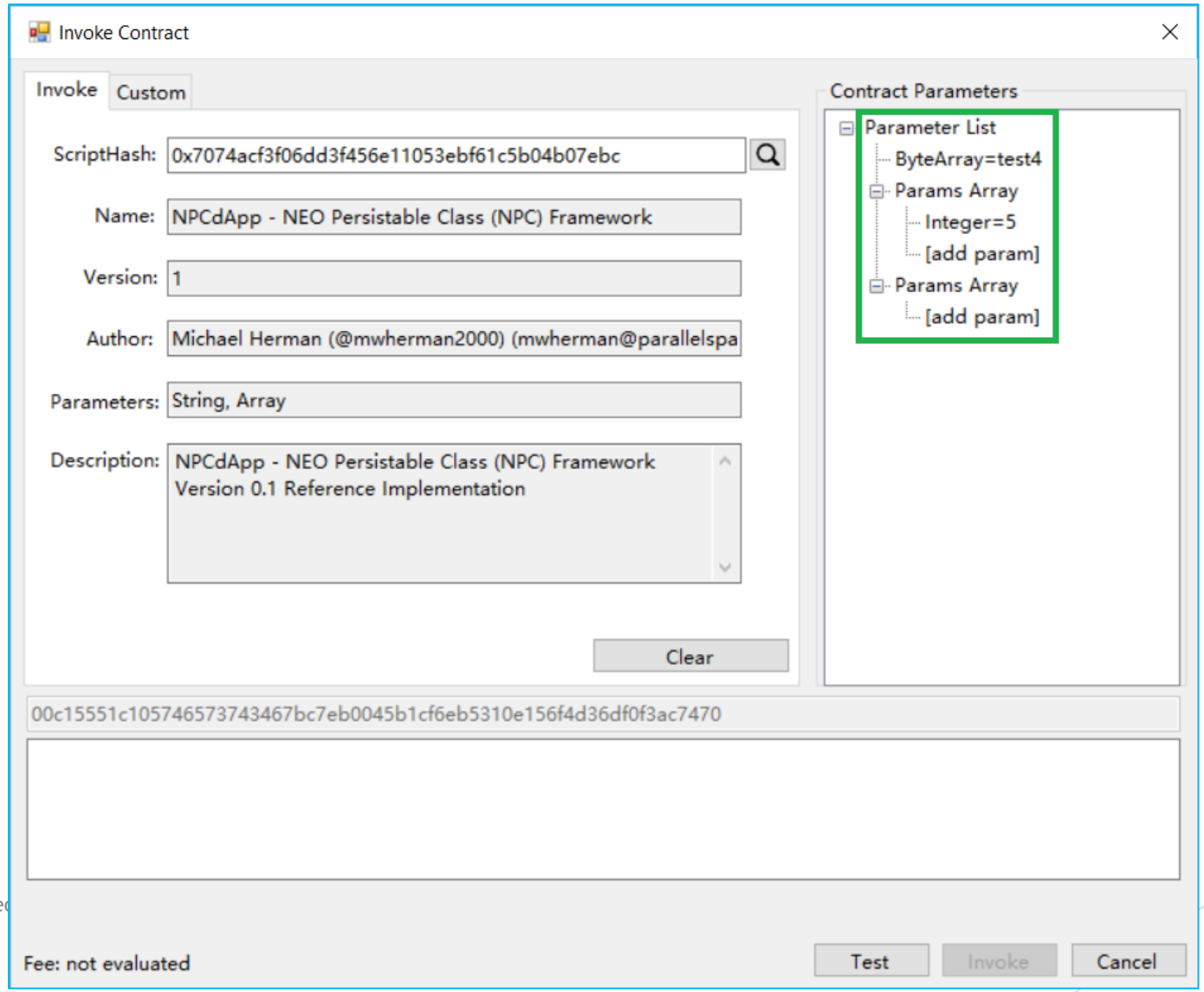
Running NPCdApp 1.0 with neo-debugger



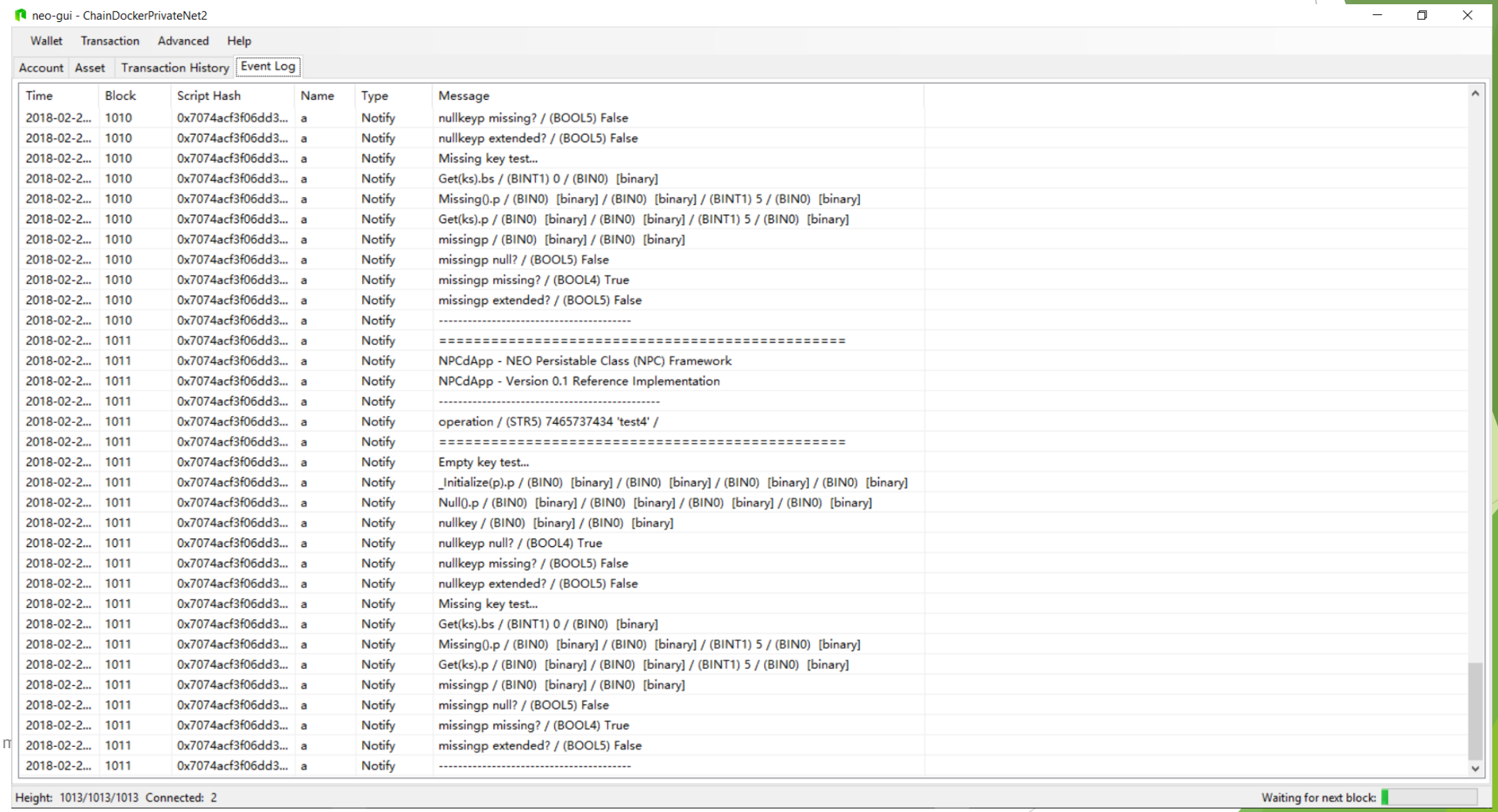
Running NPCdApp 1.0 with **neo-gui-developer**

- Network: **NEO Testnet**
- NPCdApp SC Address: **0x7074acf3f06dd3f456e11053ebf61c5b04b07ebc**
- Use latest version of neo-gui-developer with new Event Log smart formatting support
  - Should be here: https://github.com/CityOfZion/neo-gui-developer/pull/46
  - Backup plan: https://github.com/mwherman2000/neo-gui-developer
  - This version includes specific smart formatting support for a range of NEO data types and key types
  - No direct support for NEO-KONG (yet) = NEO Key-Object Notation for Geeks
- **Calling Conventions**
  - Prototype: Main(string operation, param objects[])
  - Parameter Types: 0710
  - Operations: test1| test2| test3 | test4 | test5 | test6
  - ArrayOfObjects: [5] – only used by test6 to control the maximum number of iterations
- Results
  - Main will return “success”
  - The real results are the messages in the Event Log tab

Running NPCdApp 1.0 with neo-gui-developer



Running NPCdApp 1.0 with neo-gui-developer



Appendix A – Test Case Code

mwherman2000/neo-persistibleclasses

Test1() - Code

```
public static string test1(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("NullHash", NeoEntityModel.NullScriptHash);

    NeoTrace.Trace("NeoEntityModel.EntityState...");
    NeoEntityModel.EntityState state1 = NeoEntityModel.EntityState.MISSING;
    NeoTrace.Trace("state", state1);
    int istate = (int)state1;
    NeoTrace.Trace("state1", state1);

    BigInteger bis = state1.AsBigInteger();
    NeoTrace.Trace("bis", bis);

    byte[] bsta = { 0x4 };
    NeoTrace.Trace("bsta", bsta);
    NeoEntityModel.EntityState state2 = NeoEntityModel.BytesToEntityState(bsta);
    NeoTrace.Trace("state2", state2);

    return msg;
}
```


Test2() - Code

```
public static string test2(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Make P0...");
    Point p0 = Point.New();
    Point.Log("p0", p0);
    Point.SetX(p0, 7);
    Point.SetY(p0, 8);
    Point.Log("p0", p0);
    Point.Set(p0, 9, 10);
    Point.Log("p0", p0);

    NeoTrace.Trace("Make P1...");
    Point p1 = Point.New();
    Point.Set(p1, 2, 4);
    Point.Log("p1", p1);

    NeoTrace.Trace("Make P2...");
    Point p2 = Point.New();
    Point.Set(p2, 15, 16);
    Point.Log("p2", p2);

    NeoTrace.Trace("Make line1...");
    Point[] line1 = new[]
    {
        p1, p2
    };
    NeoTrace.Trace("line1", line1, p1, p2); // TODO: neo-gui doesn't understand this: line1

    NeoTrace.Trace("Add 2 points...");
    Point p3 = Add(line1[0], line1[1]);
    Point.Log("p3", p3);

    return msg;
}
```

Test3() - Code

```
public static string test3(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Make P1...");
    Point p1 = Point.New();
    Point.Set(p1, 2, 4);
    Point.Log("p1", p1);

    NeoTrace.Trace("Make P2...");
    Point p2 = Point.New();
    Point.Set(p2, 12, 14);
    Point.Log("p2", p2);

    NeoTrace.Trace("Make P3...");
    Point p3 = Point.New();
    Point.Set(p2, 22, 24);
    Point.Log("p3", p3);

    NeoTrace.Trace("Put P1...");
    Point.Put(p1, "p1");
    NeoTrace.Trace("Put P2...");
    Point.Put(p2, "p2");
    NeoTrace.Trace("Put P3...");
    Point.Put(p3, "p3");

    NeoTrace.Trace("Get P1...");
    Point p1get = Point.Get("p1");
    Point.Log("p1get", p1get);
    NeoTrace.Trace("Get P2...");
    Point p2get = Point.Get("p2");
    Point.Log("p2get", p2get);
    NeoTrace.Trace("Get P3...");
    Point p3get = Point.Get("p3");
    Point.Log("p3get", p3get);

    return msg;
}
```

Test4() - Code

```
public static string test4(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Empty key test...");
    Point nullkeyp = Point.Get("");
    Point.Log("nullkey", nullkeyp);
    NeoTrace.Trace("nullkeyp null?", Point.IsNull(nullkeyp));
    NeoTrace.Trace("nullkeyp missing?", Point.IsMissing(nullkeyp));
    NeoTrace.Trace("nullkeyp extended?", Point.IsExtended(nullkeyp));

    NeoTrace.Trace("Missing key test...");
    Point missingp = Point.Get("missingp");
    Point.Log("missingp", missingp);
    NeoTrace.Trace("missingp null?", Point.IsNull(missingp));
    NeoTrace.Trace("missingp missing?", Point.IsMissing(missingp));
    NeoTrace.Trace("missingp extended?", Point.IsExtended(missingp));

    return msg;
}
```


Test5() - Code

```
public static string test5(object[] args)
{
    string msg = "success";

    NeoTrace.Trace("Test NeoStorageKeys...");
    Point p4 = Point.New();
    Point.Set(p4, 10, 20);
    Point.Log("p4", p4);

    string app = "FooBar";
    byte[] user = WIF2AccountAddressScriptHash;
    NeoVersionedAppUser vau = NeoVersionedAppUser.New(app, 1, 0, 2034, user);
    NeoVersionedAppUser.Log("test5.vau", vau);

    int index = 24;
    NeoTrace.Trace("index", index);
    Point.PutElement(p4, vau, index);
    index = 25;
    NeoTrace.Trace("index", index);
    Point.PutElement(p4, vau, index);

    index = 24;
    Point p4get = Point.GetElement(vau, index);
    Point.LogExt("p4get", p4get);

    Point p4bury1 = Point.BuryElement(vau, index);
    Point.LogExt("p4bury1", p4bury1);
    Point p4bury2 = Point.GetElement(vau, index);
    Point.LogExt("p4bury2", p4bury2);

    return msg;
}
```

Test6() - Code

```
public static string test6(object[] args)
{
    string msg = "success";
    int maxIterations = 10;
    if (args.Length > 0)
    {
        maxIterations = (int)((byte[])args[0]).AsBigInteger();
        NeoTrace.Trace("maxIterations", maxIterations);
    }
    if (maxIterations <= 0) maxIterations = 10;
    if (maxIterations > 20) maxIterations = 10;
    NeoTrace.Trace("maxIterations", maxIterations);

    byte[] callingUserScriptHash = ExecutionEngine.CallingScriptHash;
    NeoTrace.Trace("callingUserScriptHash", callingUserScriptHash);
    byte[] entryUserScriptHash = ExecutionEngine.EntryScriptHash;
    NeoTrace.Trace("entryUserScriptHash", entryUserScriptHash);
    byte[] executingUserScriptHash = ExecutionEngine.ExecutingScriptHash;
    NeoTrace.Trace("executingUserScriptHash", executingUserScriptHash);
    byte[] invokingUserScriptHash = GetInvokingUserScriptHash();
    NeoTrace.Trace("invokingUserScriptHash", invokingUserScriptHash.Length, invokingUserScriptHash);
    if (invokingUserScriptHash.Length == 0) invokingUserScriptHash = WIF2AccountAddressScriptHash;
    NeoTrace.Trace("invokingUserScriptHash", invokingUserScriptHash);

    Point p4 = Point.New();
    Point.Set(p4, 10, 20);
    Point.Log("p4", p4);
```

Test6() - Code (con't)

```
    string app = "FooBar";
    NeoVersionedAppUser vau = NeoVersionedAppUser.New(app, 1, 0, 2034, invokingUserScriptHash);
    NeoVersionedAppUser.Log("test6.vau", vau);

    int iteration = 0;
    for (int index = 30; index < 40; index++)
    {
        NeoTrace.Trace("index", index);
        Point.Set(p4, index, -index);
        Point.PutElement(p4, vau, index);
        iteration++;
        if (iteration > maxIterations) break;
    }

    iteration = 0;
    for (int index = 30; index < 40; index++)
    {
        Point.Set(p4, index, -index);
        Point x = Point.GetElement(vau, index);
        Point.Log("loop.x", x);
        if (Point.GetX(p4) != index || Point.GetY(p4) != -index)
        {
            msg = ">>>>(x,y) are different";
            NeoTrace.Trace(msg);
            break;
        }
        iteration++;
        if (iteration > maxIterations) break;
    }

    return msg;
}
```

Questions?

Discord: Michael Herman (Toronto)

Email: mwherman@parallelspace.net

mwherman2000/neo-persistibleclasses