



WOWODC '012

MONTREAL JUNE 30, JULY 1ST AND 2ND 2012



Persistent Session Storage

Henceforth referred to as PSS :)

- Ramsey Gurley

Demo



WOWODC '012

MONTREAL JUNE 30, JULY 1ST AND 2ND 2012



What does this mean?

Robust, scalable, front end apps using component actions.

Advantages of PSS

- Memory consumption
- Long sessions
- Distribution

Memory Consumption

- Sessions storage handled by EOF
- Session memory freed after each request
- Drastically improves stateful performance under load

Long Sessions

- `WOSessionTimeOut=0`
- Bookmarkable component actions

Distribution

- WOSession's `setDistributionEnabled()`
- Restart instances anytime
- Improved scaling

How do I enable PSS?

Demo

Steps to enable PSS

- Add ERPersistentSessionStorage framework
- Set up the connection dictionary for the model
- Generate the schema from the model (Enable migrations)



WOWODC '012

MONTREAL JUNE 30, JULY 1ST AND 2ND 2012



How now brown cow?

Why has no one done this before now?

Serialization Bug

EOGenericRecord



- Serialize EO
- Serializes EC
- Deserialize EO
- Deserialize EC
- Populate EO

Workaround?

- Serialize EC first!
- Fails :(

EOGenericRecord



- Serialize EO
- Serializes related
- Deserialize EO
- Deserialize related
- Populate EO

What happens?

- EOCustomObject classDescription
- EOGenericRecord serializes classDescription
- Deserialize EOCustomObject sets values
- Dictionary not created until EOGenericRecord sets classDescription
- NullPointerException

Serialization Bug Fix

- Re-implement core classes
 - EOCustomObject
 - EOGenericRecord

Other WO Serialization Errors

- WODisplayGroup
- D2WPage

Serialization

- Classpath order - ERD2W & ERPSS
- General rule - Wonder first

PSS Requirements

PSS Requirements

- Serializable components and classes
- Serializable session
- No memory leaks

Serializable Classes

serialVersionUID

```
/**  
 * Do I need to update serialVersionUID? See section 5.6 <cite>Type Changes  
 * Affecting Serialization</cite> on page 51 of the <a  
 * href="http://java.sun.com/j2se/1.4/pdf/serial-spec.pdf">Java Object  
 * Serialization Spec</a>  
 */  
private static final long serialVersionUID = 1L;
```


Update serialVersionUID?

- Change class hierarchy
- Delete fields
 - Make non-static into static
 - Make non-transient into transient
 - Change type of a primitive field

Serializable Session

Serializable Session

- Serializable instance variables
- Serializable object store contents

No Memory Leaks

Helpful Tools

- FindBugs
- jvisualvm
- jMeter

Find Serialization Errors

FindBugs

<http://findbugs.cs.umd.edu/eclipse/>

FindBugs

The screenshot shows the Eclipse IDE interface with the FindBugs tool. The Bug Explorer on the left lists 103 bugs under the 'ERExtensions' project. The selected bug is 'Class er.extensions.eof.ERXEC defines non-transient non-serializable instance field openLockTraces'. The Bug Info panel on the right provides details for this bug.

Bug Explorer:

- ERExtensions (103) [wonder integration]
 - Bad implementation of cloneable idiom (8)
 - Bad use of return value from method (3)
 - Checking String equality using == or != (15)
 - Confusing method name (1)
 - Dead local store (3)
 - Equal objects must have equal hashcodes (1)
 - Incorrect definition of Serializable class (5)
 - Class er.extensions.eof.ERXEC defines non-transient non-serializable instance field openLockTraces
 - Class er.extensions.logging.ERXLog4JConfiguration defines private readResolve method in com.webobjects.foundation.
 - private readResolve method in com.webobjects.foundation.
 - private readResolve method in com.webobjects.foundation.
 - Misuse of static fields (3)
 - Null pointer dereference (24)
 - Possible atomicity violation (1)
 - Possible double check of field (3)
 - Synchronization on java.util.concurrent objects (3)
 - Synchronization on updated field (Mutable Lock) (1)
 - Uninitialized read of field in constructor (6)
 - Unintended contention or possible deadlock due to locking on
 - Unsynchronized Lazy Initialization (22)
 - Unwritten field (1)
 - Wait not in loop (2)
- JavaWOExtensions (4) [wonder integration]

Bug Info Panel:

ERXEC.java: 127

Navigation

Class er.extensions.eof.ERXEC defines non-transient non-serializable instance field er.extensions.eof.ERXEC.openLockTraces
Actual type Thread

Bug: Class er.extensions.eof.ERXEC defines non-transient non-serializable instance field openLockTraces

This Serializable class defines a non-primitive instance field which is neither transient, Serializable, or java.lang.Object, and does not appear to implement the Externalizable interface or the readObject() and writeObject() methods. Objects of this class will not be deserialized correctly if a non-Serializable object is stored in this field.

Confidence: High, **Rank:** Troubling (14)
Pattern: SE_BAD_FIELD
Type: Se, **Category:** BAD_PRACTICE (Bad practice)

Class er.extensions.eof.ERXEC defines non...erializable instance field openLockTraces

Serialization Debug Property

```
sun.io.serialization.extendedDebugInfo=true
```

Serialization Debug Property

```
Caused by: java.io.NotSerializableException: er.extensions.localization.ERXLocalizer
- field (class "er.modern.directtoweb.components.ERMDBatchSizeControl", name: "_localizer", ...)
- object (class "er.modern.directtoweb.components.ERMDBatchSizeControl",...)
- element of array (index: 5)
- array (class "[Ljava.lang.Object;", size: 7)
- field (class "com.webobjects.foundation.NSDictionary", name: "objects", type: "class [Ljava.lang.Object;")
- custom writeObject data (class "com.webobjects.foundation.NSDictionary")
- object (class "com.webobjects.foundation.NSMutableDictionary", ...)
- custom writeObject data (class "com.webobjects.appserver.WOComponent")
- object (class "er.modern.look.pages.ERMODListPage", ...)
- field (class "com.webobjects.appserver._private.WOTransactionRecord", name: ...)
- object (class "com.webobjects.appserver._private.WOTransactionRecord", ...)
- element of array (index: 1)
- array (class "[Ljava.lang.Object;", size: 2)
- field (class "com.webobjects.foundation.NSDictionary", name: "objects", type: "class [Ljava.lang.Object;")
- custom writeObject data (class "com.webobjects.foundation.NSDictionary")
- object (class "com.webobjects.foundation.NSMutableDictionary", ...)
- field (class "com.webobjects.appserver.WOSession", name: "_contextRecords", ...)
- root object (class "wowodc.modernpss.Session", ...)
```

Other Serialization Issues

D2WContext

- D2WContext is not serializable
- ERD2WContext is serializable
- Use ERD2WContext factory

ERD2WContext

- Does not serialize entire context
 - frame
 - task
 - entity
 - propertyKey
 - pageConfiguration
 - object

ERXEC

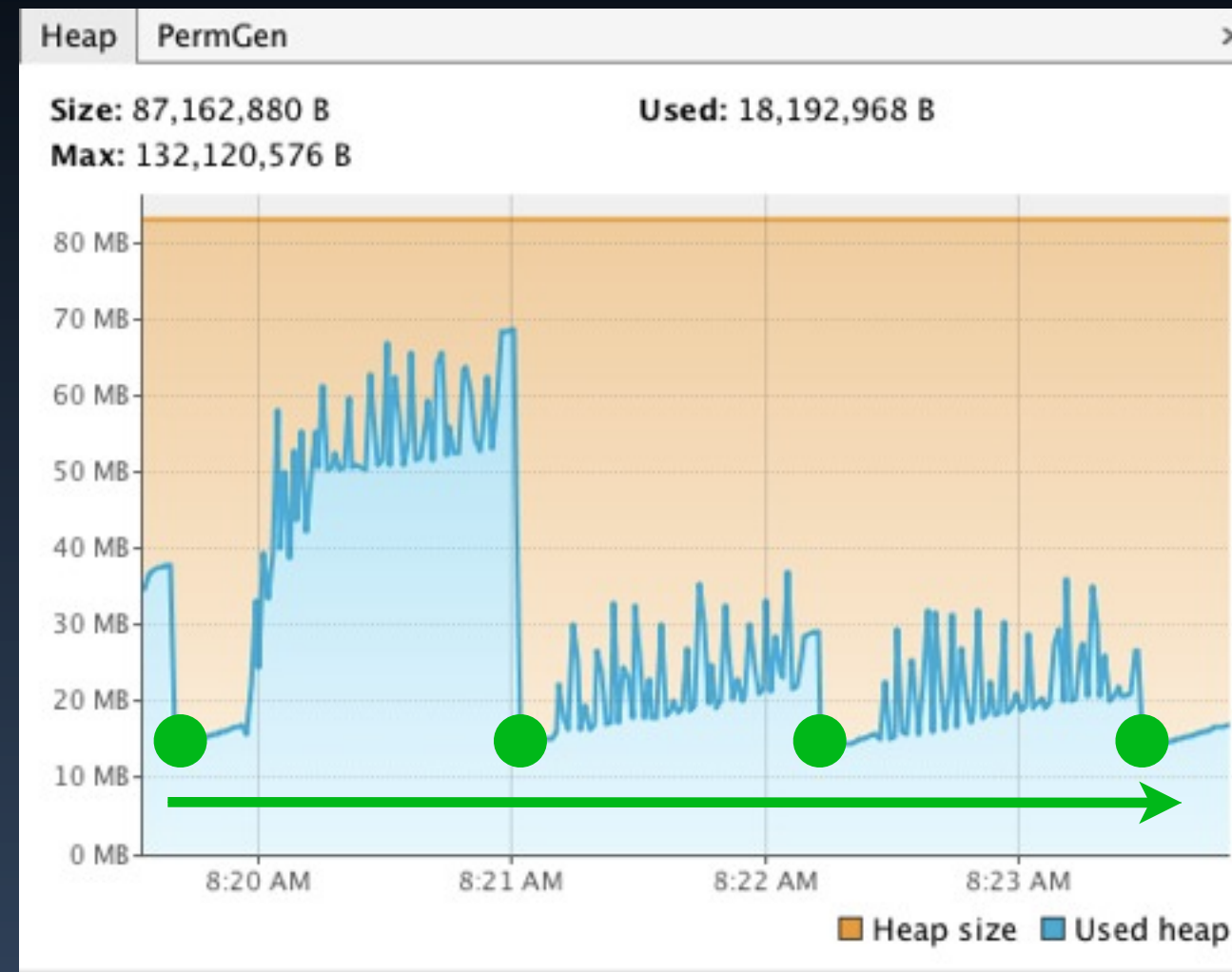
- calls `dispose()` during `writeObject()`

Find Memory Leaks

jvisualvm

- Installed with Java 1.6+
- Just type 'jvisualvm' into Terminal.app to launch
- View CPU and memory usage

No Memory Leak



Memory Leak



Analyzing Memory

- Heap Dump
- Object Query Language (OQL)
- <http://visualvm.java.net/oqlhelp.html>

Load Testing With JMeter

JMeter

- <http://jmeter.apache.org/>

Finding a Memory Leak Demo

Q & A