Item 185
git_comments:

1. let's assume its a collection of 1
2. **comment:** TODO: For GROOVY-2599 do we need something like below but it breaks other things return (String) invokeMethod(arguments, "toString", EMPTY_ARGS);
   label: code-design

git_commits:

1. **summary:** GROOVY-7563: Make InvokerHelper output self-consistent (minor refactor - closes #112)
   message: GROOVY-7563: Make InvokerHelper output self-consistent (minor refactor - closes #112)

github_issues:

github_issues_comments:

github_pulls:

1. **title:** GROOVY-7563: Make InvokerHelper output self-consistent
   body: Changes the internal API of the formatting methods in InvokerHelper and also the output in some cases when using verbose or safe arguments. This PR contains currently controversial changes to the formatting methods in InvokerHelper that could break client code. Further analysis and discussion required. This currently contains the commits from #96 (until that one is merged). (This supercedes #111, which had the wrong JIRA number in commit messages)

github_pulls_comments:

1. compileJava is failing: http://ci.groovy-lang.org/viewLog.html?buildId=26801&tab=buildResultsDiv&buildTypeId=Groovy_Jdk6Build
2. I'll fix that, just an API change that was lost when I changed git history. it would be great if there was a notification system that would send me an email. Possibly Travis could be used as an additional automated test for better integration with github, if the groovy ci chain cannot easily be extended towards github.
3. Further integration would be nice. In the meantime, it might require a quick manual check: http://ci.groovy-lang.org/project.html?projectId=Groovy
4. I'll try to look at this again shortly. I've forgotten where we got up to. Any chance of a rebase and removal of the commits from PR#96? Thanks.
5. Rebased. Jochen had comments at: https://issues.apache.org/jira/browse/GROOVY-7563
6. It seems Jochen agrees with all changes. He had a concern about exceptions appearing - but really it is only the Classname@hashCode that appears which is what we do now. I am leaning that way as well.
7. All commits merged. Thanks!

github_pulls_reviews:

jira_issues:

1. **summary:** InvokerHelper formatting methods have inconsistent API
   description: in class org.codehaus.groovy.runtime.InvokerHelper, there are methods used to print out Objects. The methods sometimes have a maxsize, verbose or safe argument, but they do not have them consistently. I suggest changing the private API so that methods for all Collection/Map types have the same API and same functionality. See https://github.com/apache/incubator-groovy/pull/96

jira_issues_comments:

1. **body:** In GROOVY-7564 Paul suggests extracting formatting methods into a class outside InvokerHelper and improving the documentation. Also changes to the output that can likely break other projects need clear documentation in the release process, and ideally the existing API should remain backwards compatible. So I can work on clarifying the changes to the output the PR will produce.
   label: documentation

2. I created https://github.com/apache/incubator-groovy/pull/111 focussing on behavior-changing commits to InvokerHelper. https://github.com/apache/incubator-groovy/pull/96 now contains only refactorings in the strict sense (no change to output).
3. GitHub user tkruse opened a pull request: https://github.com/apache/incubator-groovy/pull/112 GROOVY-7563: Make InvokerHelper output self-consistent Changes the internal API of the formatting methods in InvokerHelper and also the output in some cases when using verbose or safe arguments. This PR contains currently controversial changes to the formatting methods in InvokerHelper that could break client code. Further analysis and discussion required. This currently contains the commits from #96 (until that one is merged). (This supercedes #111, which had the wrong JIRA number in commit messages) You can merge this pull request into a Git repository by running: $ git pull https://github.com/tkruse/incubator-groovy groovy-7563 Alternatively you can review and apply these changes as the patch at: https://github.com/apache/incubator-groovy/pull/112.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #112 ---- commit 7348234fc682e8bdc63d1d7e3bd0e0581d9307a1 Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-08-21T14:02:32Z Refactor: Code style - if block commit 8787aea597cef9c47dc0b3a619cc44d9da34a5ac Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-08-21T14:06:46Z InvokerHelper: Refactor: Better StringBuffer pre-allocation commit f898b92f32a14e03bff6f17ac0afbe163cf59a72 Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-08-21T14:07:38Z InvokerHelper: Refactor: Rename private method commit 677cccf903e154b2d16be55c2df0d52a2a48a403 Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-08-21T14:11:52Z InvokerHelper: Refactor: Implement Array formatting like Collection formatting commit 4571a7fa8d5be7ab16ccca7910a70287a58efee9 Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-09-01T22:15:55Z InvokerHelper: Refactor: invoke specific methods. commit a6826f1d1d00607ee9259b8ba81dbd9e3217e78a Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-09-01T22:17:40Z InvokerHelper: Refactor: Avoid unnecessary array transformation for Object[] commit fc41a62c9351f0714db950df7b5ad043f5b19ea0 Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-09-01T22:17:43Z InvokerHelper: Refactor: Remove redundant if cases, format method has them commit 9982da56abcb608be9ff6e61a6562d43a8da4599 Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-09-01T22:25:13Z InvokerHelper: Regression tests for formatting commit e3ba0a24ac0b42c92525df286ab32b56a3dfbaaf Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-09-01T22:17:43Z GROOVY-5763: InvokerHelper: Remove redundant if cases, format method has them commit 0b493fb6c8ab70d901053879d4d5122961f576a8 Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-08-21T14:23:10Z GROOVY-7563: InvokerHelper: Provide safe formatting consistently (not just for lists) commit 70b4e2cf0620a20dab53e8bbe107db4fd4f27e4f Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-08-21T14:23:37Z GROOVY-7563: InvokerHelper: Map formatting safe and self-detecting also for keys commit 4f17c6bf9d7af37e859e05afb766b85b9a92603a Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-08-21T16:28:00Z GROOVY-7563: InvokerHelper: Safe formatting of ranges, rethrow RuntimeException without wrap. commit d158abdb73a6bcec0bdcbd7c8dc925c1c0fa53be Author: Thibault Kruse <thibault.kruse@gmx.de> Date: 2015-09-01T22:37:51Z GROOVY-5763: InvokerHelper: Additional regression tests for formatting ----
4. **body:** Okay, there are 4 distinct changes that I suggest to make. We can discuss for each whether they need to be provided with a new API, or can replace existing behavior without unduly breaking existing code. h2. 1. toString(Range x) The current state is this: {code} InvokerHelper.format(1..4, false) == 1..4 InvokerHelper.format([1..4], false) == [1..4] InvokerHelper.toString(1..4) == [1, 2, 3, 4] // surprise InvokerHelper.toString([1..4]) == [1..4] {code} So ranges get expanded to lists in printing, unless they are contained in any other Structure (Collection, array, Map). That is because ranges get special treatment in format, but not in toString(), and toString uses format for recursions. Possible alternatives for groovy-2.5+: - Always print ranges in List-form in toString() - Always print ranges in range-form in toString. - Keep current inconsistent behavior. Any change here can impact users of Groovy who rely on the current inconsistent behavior, so I don't know how to handle this. h2. Provide safe printing for maps and arrays as well The current state is this: {code} class ExceptionOnToString {String toString(){ throw new UnsupportedOperationException()}} def eInst = new ExceptionOnToString() InvokerHelper.toListString([eInst], -1, true) == '[<ExceptionOnToString@a1b2c3d4>]' InvokerHelper.toListString([[eInst]], -1, true) == '[<ArrayList@a1b2c3d4>]' // unclear exactly what Object threw exception InvokerHelper.toListString([[[[[eInst, 1, 2, 3]]]]], -1, true) == '[<ArrayList@a1b2c3d4>]' // same as preceding line InvokerHelper.toListString([[x: eInst]] -1, true) == '[<LinkedHashMap@a1b2c3d4>]' InvokerHelper.toListString([[x: [y: [z: eInst, a: 1, b: 2]]]] -1, true) ==

'[<LinkedHashMap@a1b2c3d4>]' // same as preceding line // no similar api with 'safe' argument for arrays or maps {code} So toListString has an argument safe that makes the method catch exceptions (though not for nested structures), but that feature does not exist for the other data structures. As a result, the provided information is poor in some cases, and there just is not the same useful feature for printing other things than lists. Possible alternatives for groovy-2.5+: - Keep current inconsistent behavior. - remove (/ deprecate+later remove) method with safe flag - Provide Safe variant non-recursive methods for other collections - Provide recursively safe method for nested structures I believe no client code can justifiedly rely on the fact that passing safe will *NOT* be safe unless for arguments directly (not nested) contained in a collection. So I would say providing safe alternatives that are recursively safe would be okay for Groovy2.5+ One problem is whether to make that change to the format method, thus changing the output of power-asserts in cases when toString() throws Exceptions, or whether to provide an alternative safe format method that does not impact powerasserts in any way. Note though that in GROOVY-7569 I suggest changing the output of powerasserts in a much more drastic way anyway, so if that were to be accepted, it would seem strange to refuse changes in this case. h2. Map: key-self detection The current state is this: {code} Map m = [:] m.put('x', m) InvokerHelper.toMapString(m) == '[x: (this Map)]' Map m2 = [:] m2.put(m2, 'x') InvokerHelper.toMapString(m2) // StackOverflowError {code} Again I would argue that no client code can justifiedly rely on a StackOverFlowError being thrown, so changing this to not throw it cannot impact users in an unjustified way. h2. Provide safe printing for Ranges as well When the objects defining the range borders throw an exception in toString(), the question is whether to handle that like for Collections. This is such a rare and exotic case that I would think it does not matter either way. Since implementing safety here is simple (if a safe format method exists), I would suggest to allow it being safe. I changed the PR to have 4 commits to closely match the 4 suggested features above.
label: code-design

5. **body:** I must say I care less about invokerHelper.format, I care more about println(x) and power asserts. I think the difference is especially, that println does not use safe, but the power assert does. And for those I am for: * always print range in range format * provide recursively safe method for nested structures * as for the map key-self-detection.... I am for trying to recognize this, yes * but not for just putting the exception itself in the text
   label: code-design
6. I opened PRs (in September 2015, see comments above). One PR has refactorings, the other has separate commits for the separate changes I suggest here. So to go forward I would suggest merging the refactorings and cherry picking the changes that are accepted. The controversial changes not cherry-picked can then be discussed in greater detail in a separate PR.
7. Github user asfgit closed the pull request at: https://github.com/apache/groovy/pull/112
8. **body:** This is a breaking change if you were relying on the specific output from InvokerHelper format (which is used by some of the Groovy overriding println methods under the covers). The new results won't be unexpected just slightly different than before, and more importantly, more consistent.
   label: code-design