Nirva Application Platform - Bug #30

Exception propagation in Java procedures

10/13/2011 11:36 AM - Lionel Martin

Status:	Closed	**Start date:**	10/13/2011
Priority:	Normal	**Due date:**	
Assignee:	Pierre Marc	**% Done:**	100%
Category:	Application Platform	**Estimated time:**	0.00 hour
Target version:	4.7.000		
Operating System:	Any	**Tested:**	
Version:			

Description

Hello,

As described in the subject, the problem is linked to Nirva exception propagation in Java procedures.
The Java file joined to the issue contains 4 methods.
main : method calling the 2 other methods as Java procedures to display the Nirva error caught
wrongError : method to illustrate the problem.
goodError : method to illustrate when it is ok.
wrongErrorNotWorkaroundable : method to illustrate that it may be impossible to reorganize code.

Description when problem occurs (wrongError method):
The code sets the Nirva error via SetErrorEx method.
It then launches a nirva Command, with NV_NO_ERROR and NV_KEEP_ERROR flags to YES.
A debug message is logged. It shows the correct error.
But in the main procedure, the error caught is not the logged error.

In the wrongError method, several workarounds exist : moving up Nirva command or resetting SetErrorEx with ("", "", -1, "") parameters right before return statement.
But in the wrongErrorNotWorkaroundable method the executed code is the same but the workarounds can not be applied.
The only solution found is to set a SetErrorEx with ("", "", -1, "") parameters in the finally statement. But it is executed also when no errors occur and I do not know if I can rely on this command not to send an error to Nirva.

I also noted (see main method) that returning 0 to the procedure does not mean the procedure won't throw a Nirva exception if SetErrorEx has been set with ("", "", -1, "") parameters and a previous error exists. Is it the wanted behaviour ?

Note : tested on Linux plateform, not the others, but it should not be related.

History

#1 - 10/13/2011 12:19 PM - Pierre Marc

- Assignee set to Pierre Marc

Hello,

Will have a look and come back to you.

#2 - 10/18/2011 03:45 PM - Pierre Marc

- Status changed from New to In Progress

- % Done changed from 0 to 100

- Version deleted (4.5.000)

This is a bug. In fact there are 2 bugs.

First bug is the propagation of the error code to the calling command. This has been corrected so the parent command now receives the correcct error code.

Second bug is the fact that a correct return value (0) from the procedure may generate an error if there is a last error. This should not occur and the code has been modified in the correct way so when a procedure returns a correct value (0) there is no error code on the calling command. This bug correction may require a regression tests if you have some procedures using SetErrorEx() or SetError() but returning 0. In this case, if you generate an error you should return a value different of 0 from your procedure.

#3 - 10/19/2011 11:56 AM - Pierre Marc

- Status changed from In Progress to Feedback

A beta version that corrects the issue has been sent to you. Please let me know when you have made your tests.

Kind regards

Pierre MARC

#4 - 12/28/2011 10:47 AM - Pierre Marc

- Project changed from Nirva Software to Nirva Application Platform

Moved to Nirva Application Platform project

#5 - 12/28/2011 10:48 AM - Pierre Marc

- Category set to Application Platform

- Status changed from Feedback to Closed

- Target version set to 4.7.000

Files

BugPropagateError.class	2.1 KB	10/13/2011	Lionel Martin
BugPropagateError.java	2.22 KB	10/13/2011	Lionel Martin