
Enhydra Shark

Quick Start Example

Copyright © 2006 Together Teamlösungen EDV-Dienstleistungen GmbH

Table of Contents

..... 1

This is an example of how Shark works by using it through Swing admin application directly through POJO interface. It is supposed that you had executed "runSA" script without previously modifying SharkClient.conf file, which started shark deployed inside swing admin application.

- Go to the Shark Admin connection dialog, for User name type "admin", for the password type "enhydra" and click "OK" button. Now, you are logged on Shark.
- Go to the "Package management" section, press "Load", select "test-JavaScript.xpdl" and press "Load", or double-click "test-JavaScript.xpdl" list item. Wait while shark loads the package into memory, and press "Exit".

Now the XPDL package is in the shark's memory, and you can instantiate processes from process definitions contained within this XPDL.

- Go to the "Process instantiation management" section and select the "Package-test_js->Process definition-The Game" from the package tree at the left of the screen.
- If using professional/demo version, press "View" button to view the process (otherwise, open test-JavaScript.xpdl file in JaWE to see process definition graph):

This process represents the game played by two players. When process starts, each player has to enter integer number (1-100). When both players have entered their numbers, the "Generate Random No" activity generates random number. This activity is performed automatically by shark calling appropriate Tool Agent application. After the number is generated, shark evaluates expression for transition going from activity "Generate Random No" to activity "Update Player1 Score" - this expression checks if the number entered by the "Player 1" is closer to the generated random number than the number entered by the "Player 2". If it is, process flow goes to activity "Update Player1 Score", and if it isn't, and if Player2 number is closer, it goes to activity "Update Player2 Score". These are also activities performed by the shark using tool agent application, and they update appropriate score (add one to an existing score). Now, the game counter is incremented by activity "Update Counter", also performed by shark using tool agent application. Then, shark creates two more manual activities used to display the result to the players. When they see the result, and complete their activities, shark evaluates the expression for transition going from activity "r3" to activity "r2" - this expression checks if the value of the "counter" variable is smaller than the number of "game_cycles" variable. If it is, the new game cycle is started and otherwise process finishes.

Administrator has to perform some steps so process can run as described. First, it has to create new user that will participate in the game, then he can perform some mappings among XPDL application definitions and its real Tool Agent implementations (if he does not perform Application->Tool agent

application mappings example will still work because XPDL contains extended attributes which are used by DefaultToolAgent implementation to start appropriate Tool Agent application - see documentation about Tool agents). And finally, he has to instantiate the process, and set the number of the game cycles.

- Go to the "User management" section - "Users" sub-section, and press "New" button.

Select "admin" for Group name, enter "john" for Username, "j" for Password and its confirmation, "John" for the First name, and Doe" for the Last name, and press "OK" button.

Now you have created new account in User/Group structure that is queried by Shark's User/Group plugin when assigning tasks to users. This account will be used to represent the "Player 1" of "The Game" process.

- Go to the "User management" section - "Mapping" sub-section, and press "Add" button.

In the left pane select "Player 1 - rol", in the right pane select "john - John Doe", and press "Apply" button and then "Close" button.

Now you have mapped XPDL definition of the "Player 1" to account that you've just created. All activities that are performed by "Player 1" participant in XPDL, are going to be placed in the "john's" worklist when they come to execution.

If mapping does not exist, activity is placed into the worklist of the user who created the process. In the "The Game" process, admin will perform "Player 2" activities.

- Go to the "Application mapping" section, and press "Add" button.
 - In the left panel Select "random_no_generator" , and in the right panel select "org.enhydra.shark.toolagent.JavaClassToolAgent". Populate "Application name" field in the right pane with "RandomNoProc" (do not enter quotes). Finally, press the "Apply" button.
 - In the left panel Select "addition" , and in the right panel select "org.enhydra.shark.toolagent.JavaScriptToolAgent". Populate "Application name" field in the right pane with "AdditionProc.js" (do not enter quotes), and for "Application Mode" enter 0 (zero). Finally, press the "Apply" button.

Now the XPDL application definitions are mapped to real applications performed by two different Tool Agent applications. You can press "Close" button to close the mapping dialog.

NOTE: this step can't be done in community version. Even if you skip this step, everything will work as supposed to, because shark (default) tool agent will use static mapping information already defined in XPDL.

- Now that everything is prepared for the process execution, you can return to the "Process instantiation management" and press "Instantiate" button after you select "The Game" process.

You will be asked if you want to update some process variables. Answer yes, and enter i.e. "3" for "game_cycles" (you will play the guessing game 3 times).

- Now, go to 'Worklist management' section, press CTRL+R to refresh the content of combo-box, and select 'john' from the box. You will see the workitem for the activity "Enter 1. No". Accept it, and double-click on its row, or press "Complete" button. Now, you will be asked if you want to update variables. Press "Yes", and the following variable "player1_no" will be shown. You have to enter an integer value from 0 to 100. The number you enter will be compared to the number entered by the player2, to see which is closer to randomly generated number

You can see the variable descriptions if you press the button at the right in the variable row. Enter the value for player1_no, and press "OK".

Now, the activity is completed, and shark waits while Player2 enters his number.

- Select 'admin' user from the combo-box.

Now, the worklist contains the "Enter 2. No." activity - repeat the actions taken in previous step from Worklist Handler application.

After the activity is completed, shark performs the random number generation, decision which score to increment based on the number entered by the players, and creates two new activities for both players to see the result.

- New activity "View score" will appear in admin's and john's worklist (if it doesn't, press CTRL-R). You have to execute it both from Admin's and from john's worklist, and client application will ask you if you want to update process variables. Press OK, and you'll see the list of variables and its values in a dialog (none of this variables can't be updated, but they are here to show you the result):
 - random_no - the random number generated by the activity "Generate Random No"
 - player1_no - the number that "Player 1" (which is john) entered (the one that is compared to the number of the player2, to see which is closer to random generated number).
 - player2_no - the number that "Player 2" (which is admin) entered (the one that is compared to the number of the player1, to see which is closer to random generated number).
 - player1_score - the score of the "Player 1" (which is john). Every time the player1 guess is closer to the random number, this score is incremented.
 - player2_score - the score of the "Player 2" (which is admin). Every time the player2 guess is closer to the random number, this score is incremented.

When both players finish their "View score" activity, counter increments, and shark performs the check if it is the end of the game. If you've entered "3" for the number of game cycles, the game will proceed, and you will have to repeat previous actions from Worklist Handler and Shark Admin applications.

- If using demo/professional version, while executing the workitems, you can switch to the "Process monitor" section of Shark to graphically monitor process flow. Go to this section and select Package 'test', refresh view by selecting appropriate menu item, or pressing CTRL-R and then select the instantiated process to see which activities from process flow are currently active (in this example you will be able to see only "Enter 1. No", "Enter 2. No" and "View score" activities active, because these are only manual activities which are waiting for user input, and not automatically executed by shark.

You might also want try to execute some processes from XPDl called workflow_patterns.xpdl.

To do so, upload this XPDl into shark, read description of the process, and while executing it, read the description of each activity in worklist.