**How to configure and test**

**“Dynamic Layout” in the NDL client**

**February 2015**

Radu Nitescu

# The “Dynamic Layout” feature

The idea of this feature is that based on a set of rules (triggers) at some point of a player’s presence in the casino, he can be presented with an alternative layout (different lobby and different games in the arenas). This can be used for testing the games performance, replacing some more generous game types with ones that have lower RTPs, etc.

Example of rules could be wethear the player has been in the casino long enough (a certain amount of time has passed since his initial registration date), or maybe if he excedded a certain amount of rounds played in various game types.

At the time of writing of this document, there are 2 defined stages (player statuses) this feature is based upon and therefore we can trigger 2 layout changes for one player with one client configuration.

**NOTE**:

* This feature was implemented first in version 6.6 in such a way (both NDL code and configuration files) that it is backwards compatible in any scenario: An older version of the NDL client will work with updated configuration files but ignore content regarding this feature, and also a newer version of the NDL client (containing this feature) will work with older versions of this configuration files (prior to this feature).
* This feature is configurable per skin/brand (configured in the skin.xml, per each skin of a given brand and in the navigation\_plan\_ndl.xmm file of that brand)
* It is important to note that the values of the “playerStatus” attributes on which this feature is based have been generically named “new1” and “new2”. It’s mandatory that this naming conventions are respected by the configuration files. The next paragraph will explain this in detail.

# How to configure the feature

The files that require changes in configuration to enable this feature are:

1. …\build\versionX\brand\brand\_XXX\skin\_XXX\**skin.xml**

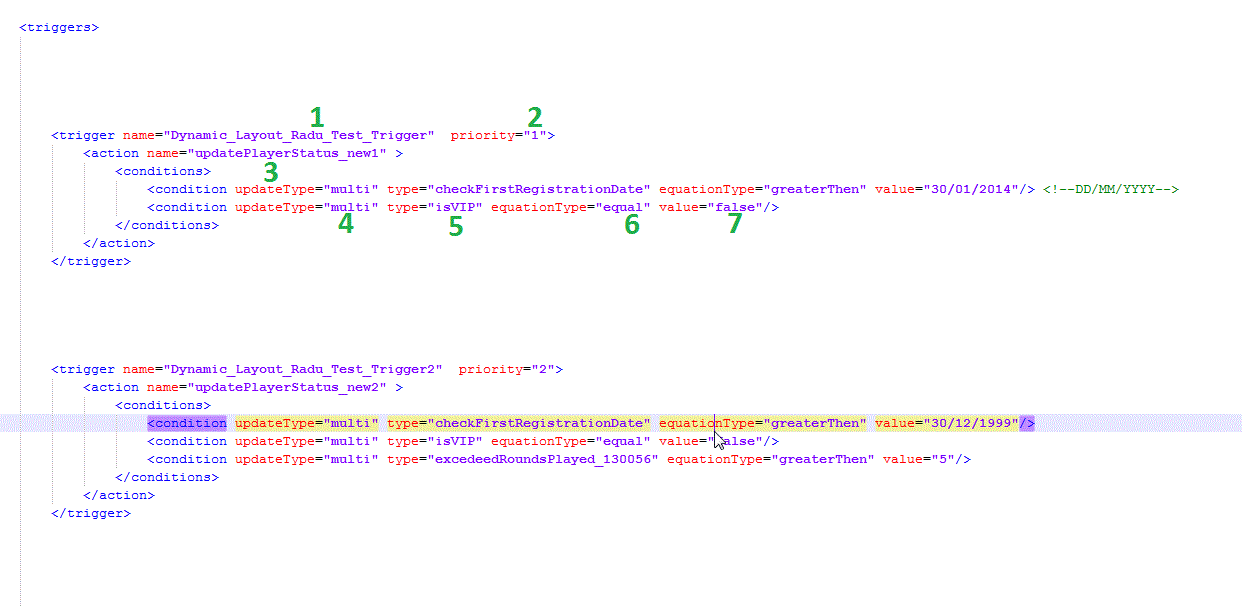
**AND**

1. …\build\versionX\navigation\plan\XXX\**navigation\_plan\_ndl.xmm**

## skin.xml configuration

In the <triggers/> node of this file, new <trigger/> nodes will be added according to the desired settings of this feature. Below is a detailed example of how to achieve this:

***Example:***



1. **Trigger name:**

This is not important; It’s only a desired name given by the configurator to identify triggers by

1. **Priority**

This attribute must be an integer higher than 0. It represents the order in which the triggers will be fired (actions will be performed). Note that for different priority values, if all triggers are to be fired, this will happen in batches as such: first all the triggers with priority=1 will be fired, then all triggers with priority=2…etc. until all triggers are fired. The lower the priority number is, the sooner the trigger will be fired.

***NOTE***: It is very important to set the correct priority for triggers that are logically supposed to get fired at different moments in time: For this feature for example, a player will **ALWAYS** achieve a playerStatus=”**new1**”, and only then may become a new player of type “**new2**”. (after having played also an established number of rounds per desired slot game type). Ergo the trigger meant to update the playerStatus to “**new2**” **MUST** have a higher priority (higher number value) that that for “**new1**”, thus making sure that the actions will be performed in the desired order.

1. **Action name**

This is the name of the action that the client will perform when all conditions for fiering a trigger will be satisfied. For this feature we will have an action name="**updatePlayerStatus\_new1**" for playerStatus=”**new1**” and an action name="**updatePlayerStatus\_new2**" for playerStatus=”**new2**”.

1. **Update type**

This attribute will take only the values of “**single**” and “**multi**” depending on whether the condition should be checked only once or multiple times.

1. **Type**

Regarding dynamic layout, this attribute will take only the following values (this may be subject to change in the future):

* + “**isVIP**”
  + “**checkFirstRegistrationDate**”
  + “**excedeedRoundsPlayed**\_” + “desired **gameType** number”

***NOTE***: It is mandatory that this naming conventions are respected exactly as described in this document, also be aware of upper/lower case.

1. **Equasion type**

The value of this attribute can only take one of the following values and as the name suggests it specifies the equasion type used when comparing a reference value to the value specified by configuration in the “**value**” attribute (**7.**):

* + var EQUATION\_TYPE\_EQUAL:String = "**equal**";
  + var EQUATION\_TYPE\_CHANGE:String = "**change**";
  + var EQUATION\_TYPE\_LESS\_THEN:String = "**lessThen**";
  + var EQUATION\_TYPE\_GREATER\_THEN:String = "**greaterThen**";
  + var EQUATION\_TYPE\_GREATER\_THEN\_OR\_EQUAL:String = "**greaterThenOrEqual**"

1. **Value**

This attribute holds the desired value to which we compare certain player traits (this feature); It can ONLY hold a value of one of the two following types:

1. **A number**
2. **A date**

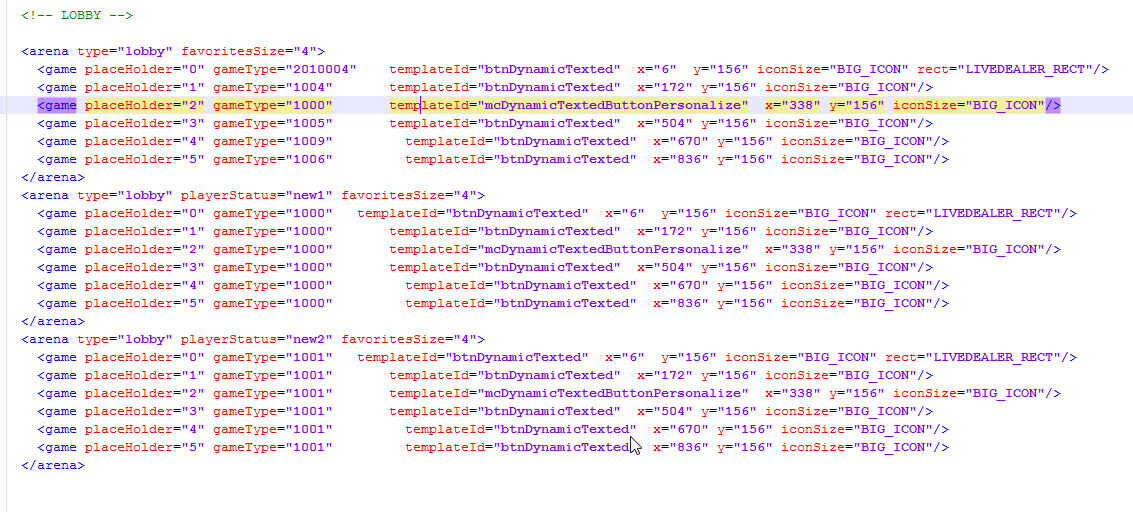
**NOTE**: A date value **must** be represented only in this format: **DD/MM/YYYY**

## navigation\_plan\_ndl.xmm configuration

In the **<arenas/>** node we can add new **<arena/>** child nodes with the desired values set to the attribute “**playerStatus**” according to the way we want to configure our dynamic layout. The values used will be “**new1**” and “**new2.** This can be applied to all arena types (lobby,1000,1001,1002, etc).

Below there is an example for the arena type=lobby, set to work with the 2 triggers we have already establised in the previous section in the skin.xml:

***Example:***



Basically we have 2 duplicates for the arena type lobby in the navigation\_plan\_ndl.xmm file, one with an attribute “ playerStatus=”new1” ” and one with another attribute “ playerStatus=”new2” ”. This will work together with the 2 triggers we have set: when the conditions imposed by a trigger are met, at the next login the player will have his lobby changed with the “new1” or “new2” one, depending on which set of rules have been satisfied.

**NOTE:** In the exact same way as we did above with the lobby, we can also “duplicate” all other arena types with “playerStatus=new1(new2)” and have them changed as well according to our desired configuration.

# Testing this feature – some ideas

1. Before anything else, the navigation\_plan\_ndl.xmm and the skin.xml files should be inspected (easily via Chrome’s built-in developer tools -> network panel (F12)) and see that they are configured exactly in a proper manner as described in the aforementioned sections of this doc.
2. Testing that triggers respect priority: Verify that indeed a trigger marked “new1” with priority=”1” is triggered before the second one marked “new2” with priority=”2” if indeed the conditions are satisfied in this order
3. Testing that the rules inside the triggers are indeed respected: testing all equation types individually (login and logout to test if a new layout is triggered always, it will NOT happen in real time). This way we can be sure that the rules are respected as configuration intends to and not by some accident. The layout change according to a said trigger will only happen when all equations defined within are satisfied.
4. Testing more sets of triggers by editing the 2 config files (navigation\_plan\_ndl.xmm and the skin.xml) with various types of conditions (Ex: exedeed some rounds played in more game types, testing more users against their registration date, etc)
5. After conditions are met and a trigger is fired (a player achieves a status of “new1” or “new2”) test to see if indeed the lobby and arenas “look” exactly like they are defined inside the navigation\_plan\_ndl.xmm in the according section and that no error has occurred.
6. When testing triggers for exceeded rounds played in a certain gameType, always be aware that this information is stored in the flash cookies that the client saves on the local machine. You can “reset” this (and revert the trigger) just by deleting the flash cookies at this URL: <http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager07.html>

by clicking “Delete all sites”. In this manner you can revert to any layout previous to a trigger that used an “excedeedRoundsPlayed\_” rule.