Garbage collection in platform services in the WRT 1.1 environment

Overview

The garbage collection of JavaScript service objects has recently changed. This change was done to improve performance and memory efficiency. Even though JavaScript objects have a small memory footprint, the memory consumed by underlaying native objects must be released.

Description

Previously, bindings to native code protected the JavaScript objects obtained through platform services. The current implementation does not protect objects anymore, and one must be careful not to let them fall out of the scope.

How to reproduce

The following code shows the situation that **must** be avoided. The correct way is included in the code comments.

```
 var media=null;
 var garbage=null;

 function init(){
     var so = device.getServiceObject("Service.MediaManagement", "IDataSource");

     //the right way
     //this.media = so;

     //this worked on previous releases
     this.media = so.IDataSource;

     garbage = new Array();
}

function getImages(){
    //ensure that garbage collector runs and invalidates so
    for(var i=0; i< 50000; i++){
        garbage[i]= new Object();
    }

    try {
        //getList() is no longer defined
 alert("Media: "+ typeof media +" Media.getList(): "+typeof media.GetList);

    //the right way
        //var result = media.IDataSource.GetList(criteria, callback);

        //does not work anymore since so falls out of scope and becomes invalid
        //var result = media.GetList(criteria, callback);
   }
   catch (e) {
       alert ("getImages: " + e);
   }
}
```

Solution

Make sure that reusable service objects and their parents do not fall out of the scope. If they fall out of the scope, the garbage collector will clean them up.

(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.