ISXEVE - Bug #2197

Item 'IsRepackable' attribute is FALSE when it should be TRUE

2021-12-27 06:20 AM - smokemonkey

Status:	Resolved
Priority:	Normal
Assignee:	
Category:	
Target version:	
vbulletin_issue_id:	

Description

repro:

dock at a station with some (used) drones which means there is no quantity displayed on their icon, and run script

```
    variable index:item items

    variable iterator itemIterator

    ; Repackage unloaded drones.

        EVEWindow[Inventory].ChildWindow[${Me.Station.ID}, StationItems]:GetItems[items]

        items:GetIterator[itemIterator]

        if ${itemIterator:First(exists)}

        {

            do

            {

                This:LogInfo[ ${itemIterator.Value.Name} ${itemIterator.Value.Group} is repackageable
${itemIterator.Value.IsRepackable}]

                if ${itemIterator.Value.Group.Find[Drone]}

                {

                    echo repackaging ${itemIterator.Value.Name}

                    itemIterator.Value:Repackage
                    ; equivalent of wait 30

                }

            }

            while ${itemIterator:Next(exists)}

        }
```

you will find the IsRepackable is always FALSE, and nothing happens when you Repackage them. This is not expected.

In short:
IsRepackable attribute
and Repackage method are both not working, at least for drones.

History

#1 - 2022-05-14 11:49 PM - Amadeus

- Status changed from New to Resolved

This is fixed in version 20220510.0012:

```
May 15, 2022
[20220510.0012]
* Added new datatype 'evemessageboxwindow' that inherits all methods/members of the 'evewindow' datatype.
* The EVEWindow TLO will now return an 'evemessageboxwindow' datatype object if you use "MessageBox" as a
parameter and
  a Message Box exists.
* Fixed the IsRepackable MEMBER of the 'item' datatype
* Fixed the Repackage METHOD of the 'item' datatype
```

<u>**However**</u>**, you will need to manage the popup windows that appear now**, including the windows that state that the item is damaged and cannot be repackaged.   Here is an updated version of the script you provided that works like a charm in my testing:

```
function main()
{
    variable index:item items
    variable iterator itemIterator

    ; Repackage unloaded drones.
    EVEWindow[Inventory].ChildWindow[${Me.Station.ID}, StationItems]:GetItems[items]

    items:GetIterator[itemIterator]

    if ${itemIterator:First(exists)}
    {
        do
        {
            if ${itemIterator.Value.Group.Find[Drone]}
            {
                if (${itemIterator.Value.IsRepackable})
                {
                    echo "-- Initiating repackaging for ${itemIterator.Value.Name}"

                    itemIterator.Value:Repackage
                    wait 20
                    if (${EVEWindow[MessageBox](exists)})
                    {
                        if (${EVEWindow[MessageBox].Text.Equal["Are you sure that you want to repackage this
item?"]})
                        {
                            echo "--- Clicking YES to repackage ${itemIterator.Value.Name}."
                            EVEWindow[MessageBox]:ClickButtonYes

                            wait 20
                            if (${EVEWindow[MessageBox](exists)})
                            {
                                if (${EVEWindow[MessageBox].Text.Equal["This item can not be repackaged as it
or one of its components is damaged. Please take the item to a repair shop and have it fixed before trying
again."]})
                                {
                                    echo "---- ${itemIterator.Value.Name} is DAMAGED...skipping."
                                    EVEWindow[MessageBox]:ClickButtonOK
                                }
                            }
                            else
                            {
                                wait 20
                            }
                        }
                        else
                        {
                            echo "--- ERROR --> Something happened that we did not expect"
```

```
                        wait 20
                    }
                }
            }
        }
    }
    while ${itemIterator:Next(exists)}
    }
}
```

You can mess around with the routine above to see which waits you can remove or reduce, if any.