

## Changes to the way licenses are consumed when running phantoms

Due to the increasing use of APIs which allow PHANTOMS to interactively solicit user input, we have implemented changes to the way licenses are consumed.

The original intent of PHANTOM licenses was for background processing such as batch file updates, report generation and other similar forms of processing where user intervention was not required. In fact, PHANTOMS will abort if they encounter an INPUT statement. However, with the introduction of the CallHTTP and Sockets interfaces as well as support for secure Sockets, XML, MQ and SOAP in recent versions of UniVerse and UniData, PHANTOMS now have the ability to interactively obtain user input without consuming a database license. This negatively impacts the U2 business and is not consistent with our competitors as they either impose limits or charge for PHANTOM licenses.

We have rolled out patches first to UniVerse 10 and 10.1 followed by UniData 6.0 and 6.1 to address this issue and enhance our current licensing mechanisms. We have defined certain restricted BASIC functions which will affect whether or not a PHANTOM consumes a database license. A PHANTOM which does not use one of the restricted functions will consume no database licenses as usual. A PHANTOM which calls a BASIC program that uses one of the restricted functions will consume a single database license. The license will be consumed as soon as the use of the restricted function is detected. Multiple calls to restricted functions from within a single PHANTOM process will NOT result in multiple licenses being consumed. Once a license is consumed it will be released only when the PHANTOM process exits. PHANTOMS which consume a database license will reduce the available user count just as if a user started a U2 shell process interactively.

The BASIC functions that will cause a PHANTOM process to consume a license are:

Sockets:	openSocket(), openSecureSocket(), initServerSocket(), initSecureServerSocket()
WebSphere MQ:	amInitialize()
callHTTP:	submitRequest()
SOAP:	SOAPSubmitRequest()

There is no additional setup on the part of the user. We provide mechanisms for the user to determine if a license is being consumed by an interactive shell user or an "interactive" PHANTOM.

This change has been implemented in the following patch releases:

- UniVerse 10.0.21 and higher
- UniVerse 10.1.1 and higher
- UniData 6.0.13 and higher
- UniData 6.1.0 and higher

Specific information is provided below for the UniVerse implementation of interactive PHANTOMS. Additional information about the specific implementation of iPHANTOMS in UniData will be provided as it becomes available. If you have any questions, please contact [U2AskUs@rs.com](mailto:U2AskUs@rs.com).

## UniVerse and Phantom Licensing

When UniVerse installs, it will stamp a 'package' called iPHANTOM (which means interactive PHANTOM) so that users can see which processes are traditional PHANTOMs and which processes are licensed PHANTOMs. CONFIG will indicate the license count just as ODBC, NLS, and UVADM do:

```
>CONFIG
Configuration data for license number 1032:
User limit =      5
Expiry date=    1/1/2500
Device licensing is in effect.
-GCI             package is installed.
                 User limit =      5. Expiry date = 1/1/2500
-NLS            package is installed.
                 User limit =      5. Expiry date = 1/1/2500
-UCI            package is installed.
                 User limit =      5. Expiry date = 1/1/2500
-UVCS           package is installed.
                 User limit =      5. Expiry date = 1/1/2500
-UVADM          package is installed.
                 User limit =      5. Expiry date = 1/1/2500
-UVODBC         package is installed.
                 User limit =      5. Expiry date = 1/1/2500
-iPHANTOM       package is installed.
                 User limit =      5. Expiry date = 1/1/2500
>
```

UniVerse tracks the use of these types of PHANTOMs in a new 'package' called iPHANTOM. It will just as UV/ODBC (except device licensing is not available to iPHANTOM processes) and UniObjects appear, one license seat per process. The output of PORT.STATUS and LIST.USER remain unchanged and these processes will appear as normal phantoms in those commands as the commands do not take license count into consideration. uvlictool\_ has been enhanced to report iPHANTOMs just as it does for UniObjects and UV/ODBC. Finally, the command 'smat -x' (ANALYZE.SHM -x) will report this 'package' information as well:

```
>ANALYZE.SHM -x

General System Information:
Login count: 5
Package "-GCI" login count: 0
Package "-NLS" login count: 0
Package "-UCI" login count: 0
Package "-UVCS" login count: 0
Package "-UVADM" login count: 0
Package "-UVODBC" login count: 0
Package "-iPHANTOM" login count: 4
Base address for printer segment: 0x3F7F9000
UniVerse home directory: /usr/ibm/uv/
```

From the above snipped ANALYZE.SHM -x report, one can see that of the 5 database licenses used, 4 are interactive PHANTOMs (iPHANTOMs).

Below is the output of uvlictool:

```
$ uvlictool
Device licensing is in effect.
No. Pid, Package Device name          IP address          Device sub-key
1  17921
```

```

2 17969 iPHANTOM

    Packages in use:  iPHANTOM
3 17970 iPHANTOM

    Packages in use:  iPHANTOM
4 17973 iPHANTOM

    Packages in use:  iPHANTOM
5 17974 iPHANTOM

    Packages in use:  iPHANTOM
5 license seats are in use.
0 license seats are available.

    Packages in use:
    iPHANTOM has 5 licensed seats and 4 in use.
$

```

## UniData and Phantom Licensing

For UniData, LISTUSER has been modified to include the new user type iPhantom. These appear as iPhantom connections and a count of these connections is maintained in the header of the report. A sample report is below:

```

[<code>]:LISTUSER[</code>]

[<code>]Licensed/Effective # of Users          Udt      Sql      iPhantom
Total[</code>]

[<code>]          32 / 32                      1        0        1
2[</code>]

[<code>]UDTNO  USRNBR  UID   USRNAME  USRTYPE  TTY      IP-ADDRESS  TIME
DATE[</code>]
[<code>]  1  2216  197612  ltest1   udt      pts/1    192.168.0.101 13:38:08
May 21 2004[</code>]
[<code>]  2  2096  197612  ltest1   iphantm  pts/2    Console      13:40:36
May 21 2004[</code>>]
[<code>] 129 3668          1 SYSTEM   tool     pts/129  Unknown      07:37:26
May 21 2004[</code>]

:

```

## Notes

1. On UniVerse, if one of the above mentioned functions fails due to all of the database licenses being consumed, STATUS() would return a value of 99, but @SYSTEM.RETURN.CODE will not be updated.

Additionally, the message, "UniVerse user limit has been reached, and license cleanup failed", will be recorded in the &PH& record for the PHANTOM in question. The message only means the licenses have

been consumed. There was no recovery from that situation. Your application will need to trap the return value of 99 or it will continue without successfully executing these functions.

2. When iPhantoms are used to connect to the same U2 instance, if named pipes are used, licenses will not be consumed.

## Summary

- U2 will now charge a database license for each PHANTOM that uses any of the following BASIC functions: openSocket(), openSecureSocket(), initServerSocket(), initSecureServerSocket(), amInitialize(), submitRequest(), SOAPSubmitRequest()
- Once a PHANTOM becomes an iPHANTOM, the license is returned when the iPHANTOM terminates
- Only one license is consumed by any iPHANTOM, regardless of the number of BASIC calls made within the application it runs
- iPHANTOM licenses will consume a database license and report as iPHANTOM licenses, just as UV/ODBC and UniObjects currently do, except device licensing does not apply to iPHANTOM processes
- For UniVerse, smat and uvlictool have been enhanced to display iPHANTOM information (includes their UniVerse shell counterparts)
- One iPHANTOM license equals one database license