

Artigo

[Djeniffer Greffin](#) · Abr. 9, 2021 5min de leitura

Rotina para dump de usuários conectados

Por diversas vezes na empresa na qual trabalho foi necessário guardarmos um histórico dos usuários conectados ao cachê. Para podermos avaliar alguns comportamentos de regionais ou setores da empresa em relação a quem estava conectado e por quanto tempo essa pessoa permaneceu trabalhando no sistema. Até mesmo para realizarmos um controle do licenciamento internamente.

A rotina de exemplo consiste basicamente em rodar a cada 60 segundos, guardar o valor total consumido da licença distribuída calculada no servidor de licenças e a cada 5 minutos realizar um dump dos usuários conectados em um arquivo separado para verificação posterior.

Como por diversas vezes tive que fazer uso desta rotina, acredito que possa ser útil para mais pessoas.

A rotina vai gerar 2 arquivos com os seguintes formatos:

- No primeiro arquivo será gerado os dados seguindo este padrão: 23/02/2021;09:43:05;2228 -> data;hora;quantidade de licenças alocadas no licence server
- No segundo arquivo será gerado os dados no seguinte formato: 16400072@10.10.110.10;1;1;10.10.110.11;CACHE -> id da licença do usuário;unidades de licença consumidas;quantidade de conexões;ip do servidor ao qual usuário está conectado;nome da instância do banco de dados.

A rotina pode ser facilmente adaptada para virar uma task no portal do cachê.

Os diretórios também estão fixos, portanto devem ser ajustados conforme necessidade.

A rotina vai rodar em background e para finalizar a mesma é só setar a global `S ^showLicense = 0` na mesma namespace em que a rotina foi iniciada ou finalizar o processo em background iniciado através do portal do cachê.

Após os arquivos gerados normalmente importamos em alguma ferramenta de análise para gerarmos gráficos setorizados a respeito do consumo.

Para iniciar a rotina:

```
Do ##class(zUtil.VerificaUnidadeLicenca).IniciaLog()
```

```
Class zUtil.VerificaUnidadeLicenca Extends %RegisteredObject
{
```

```
ClassMethod IniciaLog(){
    Job ##class(zUtil.VerificaUnidadeLicenca).VerificaUnidadeLicenca2()
    Write "Processo iniciado: ", $ZChild
}
```

```
ClassMethod VerificaUnidadeLicenca2() As %Status
{
    Set Status = $System.Status.OK()
```

```

Set FullFilename = ##class(%Library.File).NormalizeFilename(
"/tmp/license/license_"_$ZDate($Piece($Horolog,"",1),8)_"_log")
Set arquivo1 = ##class(%Library.File).%New(FullFilename)
Do arquivo1.Open("NSRW")
Set rset = ##class(%ResultSet).%New("%SYSTEM.License.Summary")
If 'rset Set Status="Could not form query %SYSTEM.License.Summary."
Kill ^showLicense
Set ^showLicense = 1
For
{
Quit:^showLicense=0
Set retorno = rset.Execute()
If retorno
{
Set ncol = rset.GetColumnCount()
If (rset.Next())
{
Set qtdDistribuida = rset.GetData(3)
Set linha = $ZDate($Piece($Horolog,"",1),4)_"_"$ZTime($Piece(
$Horolog,"",2))_"_"_qtdDistribuida
Set Status = arquivo1.WriteLine(linha)
Set ultDispatch = $Get(^showLicense("ultdispatch"))
If (ultDispatch '= "")
{
Set ultdata = $Piece(ultDispatch,"",1)
Set ulthora = $Piece(ultDispatch,"",2)
Set data = $Piece($Horolog,"",1)
Set hora = $Piece($Horolog,"",2)
If (ultdata = data)
{
If ((hora - ulthora) < 300)
{
Hang 60
Continue
}
}
Else
{
If (((86400 - ulthora) + hora) < 300)
{
Hang 60
Continue
}
}
}
Set FullFilename2 = ##class(%Library.File).
NormalizeFilename("/tmp/license/licensedispatch_"_$Translate($ZDateTime(
$Horolog,8)," :-","")_"_log")
Set arquivo2 = ##class(%Library.File).%New(FullFilename2)
Do arquivo2.Open("NSRW")
Set rset2 = ##class(%ResultSet).%New("%SYSTEM.License.ConnectionList")
If 'rset2 Set Status=
"Could not form query %SYSTEM.License.ConnectionList."
Set retorno2 = rset2.Execute()
If retorno2
{
Set ncol = rset2.GetColumnCount()
For i=1:1:ncol Set headings(i)=rset2.GetColumnName(i)
s nrec=0

```

```

        While rset2.Next()
        {
            Set nrec = nrec + 1
            Set ncol = rset2.GetColumnCount()
            For i=1:1:ncol s data(nrec,i)=rset2.GetData(i)
        }
    }
    Do rset2.Close()
    Set linha = headings(1)_" ";"_headings(2)_" ";"_headings(3)_" ";"_
headings(4)_" ";"_headings(5)
    Set Status = arquivo2.WriteLine(linha)
    For i=1:1:nrec
    {
        Set linha = data(i,1)_" ";"_data(i,2)_" ";"_data(i,3)_" ";"_data(i
,4)_" ";"_data(i,5)
        Set Status = arquivo2.WriteLine(linha)
    }
    Do arquivo2.Close()
    Set ^showLicense("ultdispatch") = $Horolog
    }
}
    Hang 60
}
Do rset.Close()
Do arquivo1.Close()
Quit Status
}
}

```

[#Controle de acesso](#) [#Caché](#) [#Ensemble](#)

URL de origem: <https://pt.community.intersystems.com/post/rotina-para-dump-de-usu%C3%A1rios-conectados>