

Artículo

[Mathew Lambert](#) · Mar 4, 2020 Lectura de 5 min

La importancia de recopilar información precisa sobre la versión y cómo hacerlo (\$zv / \$zversion)

¡ Hola Comunidad!

Esta es una explicación acerca de:

- por qué es importante obtener el \$zv (el POR QUÉ),
- qué significan los componentes de \$zv (el QUE),
- y cómo recopilar \$zv (el CÓMO).

(La guía definitiva sobre \$zv para la galaxia ISC en palabras sencillas y entendibles).

EI POR QUÉ:

La respuesta a la mayoría de los problemas y preguntas depende de la versión que utilice del producto. Algunas veces la mayor version de la release es suficiente (por ejemplo, 2016.1). Sin embargo, la versión de mantenimiento, la build y/o la información sobre la adhoc suelen ser necesarias. La información exacta de la versión también contiene información sobre el SO (el Sistema operativo y su arquitectura) y la codificación de caracteres del producto (8 bits vs Unicode).

Algunas situaciones donde es importante conocer la versión precisa incluyen (pero no se limitan a): observar nuestro código interno para determinar si se produjo algún cambio durante el desarrollo de la versión del cliente (la corrección de un error o la introducción de una nueva función), averiguar la ubicación exacta de un error en el código, o tratar de reproducir un comportamiento que se observó.

La información más completa sobre la versión se obtiene de la cadena \$ZVersion (\$zv).

EI QUÉ:

A continuación, se muestra una lista con ejemplos de la cadena \$zv en diferentes sistemas operativos y sus versiones:

[Cache for UNIX \(Red Hat Enterprise Linux for x86-64\) 2016.1.1 \(Build 108U\) Wed Jul 6 2016 16:02:29 EDT](#)

[Cache for UNIX \(Apple Mac OS X for x86-64\) 2016.1 \(Build 531U\) Tue Aug 11 2015 23:54:30 EDT](#)

[Cache for UNIX \(IBM AIX for System Power System-64\) 2016.1 \(Build 653\) Tue Mar 1 2016 17:25:29 EST](#)

[Cache for UNIX \(Oracle Solaris for SPARC-64\) 2016.1.1 \(Build 108U\) Wed Jul 6 2016 16:13:11 EDT](#)

[Cache for UNIX \(HP HP-UX for Itanium\) 2015.2.1 \(Build 705_0_16197U\) Fri Mar 11 2016 12:33:38 EST](#)

[Cache for Windows \(x86-64\) 2015.2.2 \(Build 811U\) Thu Mar 3 2016 12:55:48 EST](#)

[Cache for OpenVMS/IA64 V8.x \(Itanium\) 2014.1.3 \(Build 775 + Adhoc 14741\) 8-JAN-2015 03:41:01.57](#)

[Cache for OpenVMS/ALPHA V8.x \(Alpha\) 2014.1.3 \(Build 775 + Adhoc 14809\) 29-JAN-2015 21:39:56.22](#)

[Cache for OpenVMS/IA64 V8.x \(Itanium\) 2012.1.2 \(Build 702U + Adhoc 12945\) 24-JUL-2013 17:36:08.03](#)

[Cache for Windows \(x86-64\) 2013.1 \(Build 446_0_13965U\) Thu May 15 2014 18:18:59 EDT](#)

Cómo interpretar los elementos de la cadena \$zv:

[Cache for Windows \(x86-64\) 2015.2.2 \(Build 811U\) Thu Mar 3 2016 12:55:48 EST](#)

La cadena \$zv siempre empieza con el producto, en este caso "Cache for":

[Cache for](#)

Seguida de la información sobre el Sistema Operativo:

[Windows \(x86-64\)](#)

Lo siguiente es el año de lanzamiento:

[2015](#)

La major version, o versión de lanzamiento:

[2](#)

La minor version o versión secundaria/de mantenimiento (solo está presente en las versiones de mantenimiento):

[2](#)

Cada elemento separado por puntos.

Después viene el número de compilación interno junto con el de subcompilación (si existe uno), el número especial y completo del paquete (si existe uno) y la codificación del producto (si es Unicode):

[\(Build 811U\)](#)

NOTA: Solamente Caché puede ser de 8 bits (esta opción se selecciona durante la instalación). Ensemble y HealthShare siempre son Unicode.

Por último, tenemos la fecha en que se compiló el paquete:

[Thu Mar 3 2016 12:55:48 EST](#)

EL CÓMO:

La información de \$zv puede recopilarse de varias maneras. Para cada caso/persona, el método preferido para recabar esta información puede ser diferente y dependerá de la configuración y la experiencia que tenga el usuario, además de otros datos que se recopilen.

1) Desde el [Management Portal](#), vaya al enlace About para obtener información sobre la versión. En la fila Version se muestra la información de la cadena \$zv. Si se trata de una instancia de HealthShare, después de \$zv se encontrará la información sobre la versión de los componentes

Back

System Overview	
Version:	Cache for UNIX (Apple Mac OS X for x86-64) 2017.2 (Build 506U) Tue Sep 20 2016 01:23:55 EDT
Configuration:	/latest/cache.cpf
Database Cache (MB):	256
Routine Cache (MB):	33
Journal file:	/latest/mgr/jrnrest/20161027.001
Superserver Port:	1972
Web Server Port:	57772
License Server Address/Port:	127.0.0.1/4001
Licensed to:	ISC Sales Engineering
Cluster support:	This system is not part of a cluster
Mirroring:	This system is not a mirror member
Enterprise Manager status:	Not managed
Time System Started:	2016-09-20 09:57:33
Encryption Key Identifier:	Not available. Encryption is not activated.
NLS Locale:	ENUW
Preferred language for this session:	English (United States) <input type="button" value="v"/>

2) Para las instancias de Caché y Ensemble, desde una **Caché terminal session**, y desde cualquier namespace, puede ejecutar "write \$zv" o "write \$SYSTEM.Version.GetVersion()".

```
USER>write $zv
```

```
Cache for UNIX (Red Hat Enterprise Linux for x86-64) 2016.1.1 (Build 108U) Wed Jul 6 2016 16:02:29 EDT
```

```
USER>write $SYSTEM.Version.GetVersion()
```

```
Cache for UNIX (Red Hat Enterprise Linux for x86-64) 2016.1.1 (Build 108U) Wed Jul 6 2016 16:02:29 EDT
```

NOTA: Para las instancias de HealthShare, utilice "write ##class(%ZHSLIB.HealthShareMgr).VersionInfo()" para incluir las versiones de los componentes relacionados, así como la cadena \$zv.

```
USER>write ##class(%ZHSLIB.HealthShareMgr).VersionInfo()
```

```
Cache for Windows (x86-64) 2015.2.2 (Build 811U) Thu Mar 3 2016 12:55:48 EST [HealthShare  
Modules:Core:15.0.7865 + Linkage Engine:14.02.7865 + Patient Index:14.02.7865 + Provider Directory:14.09.7865  
+ Active Analytics:14.01.7865]
```

3) Otro método para obtener la información de \$zv es mediante **cconsole.log**, el cual se encuentra en el directorio de la instancia mgr. Sin embargo, para HealthShare, éste no incluye información sobre los componentes. Remitirse al método cconsole.log podría ser particularmente útil si se han realizado actualizaciones recientemente y, por consiguiente, el problema que reporta al WRC ya necesita de varias \$zv o utilizar cconsole.log.

```
...  
09/20/16-09:57:32:186 (83645) 0 Allocated 364MB shared memory: 256MB global buffers,  
35MB routine buffers  
09/20/16-09:57:32:196 (83645) 0 Jrn info from prior WIJ (imflags: 0):  
  wdpass: 0  
  jrnwdpass: 34  
  fspec: /latest/mgr/jrnrest/20160920.007  
  filecnt: 6  
  fileoff: 0  
  prevfcnt: 6  
  prevfileoeff: 0  
  min trans cnt: 6  
  min trans index: 205816  
09/20/16-09:57:32:207 (83645) 0  
CSTART of  
Cache for UNIX (Apple Mac OS X for x86-64) 2017.2 (Build 506U) Tue Sep 20 2016 01:23:  
55 EDT.  
  in /latest/mgr  
  with wij: /latest/mgr/CACHE.WIJ  
  from: /latest/mgr/  
09/20/16-09:57:32:207 (83645) 0  
  OS=[Darwin], version=[Darwin Kernel Version 15.6.0: Thu Jun 23 18:25:34 PDT 2016; r  
oot:xnu-3248.60.10~1/RELEASE_X86_64], release=[15.6.0], machine=[x86_64]  
  nodename=[cgencler.iscinternal.com].  
  numasyncwibuf: 0, swdwrmax: 64, wijdirectio: off, synctype: 3  
  System Initialized.  
...
```

4) Del mismo modo, si el problema que reporta al WRC necesita de **^Buttons**, **^pButtons**, **cstat**, **CacheHung**, **Diagnostic Report**, la información que proporcione la cadena \$zv se incluirá en estos informes (una vez más, en el caso de HealthShare, estos no incluirán información sobre los componentes).

URL de fuente: <https://es.community.intersystems.com/post/la-importancia-de-recopilar-informaci%C3%B3n-precisa-sobre-la-versi%C3%B3n-y-c%C3%B3mo-hacerlo-zv-zversion>