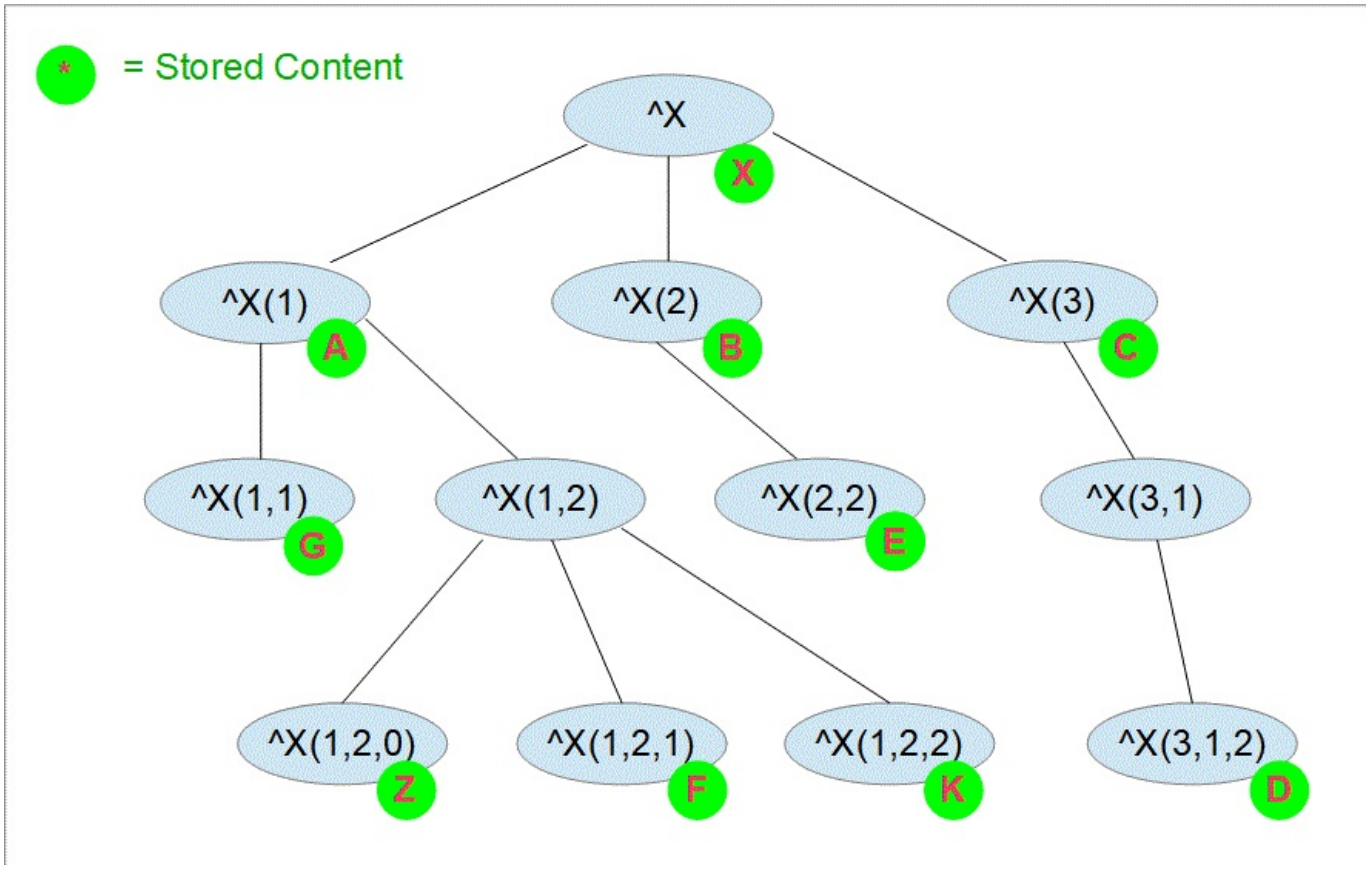
Artículo
Robert Cemper · 7 feb, 2022 Lectura de 1 min

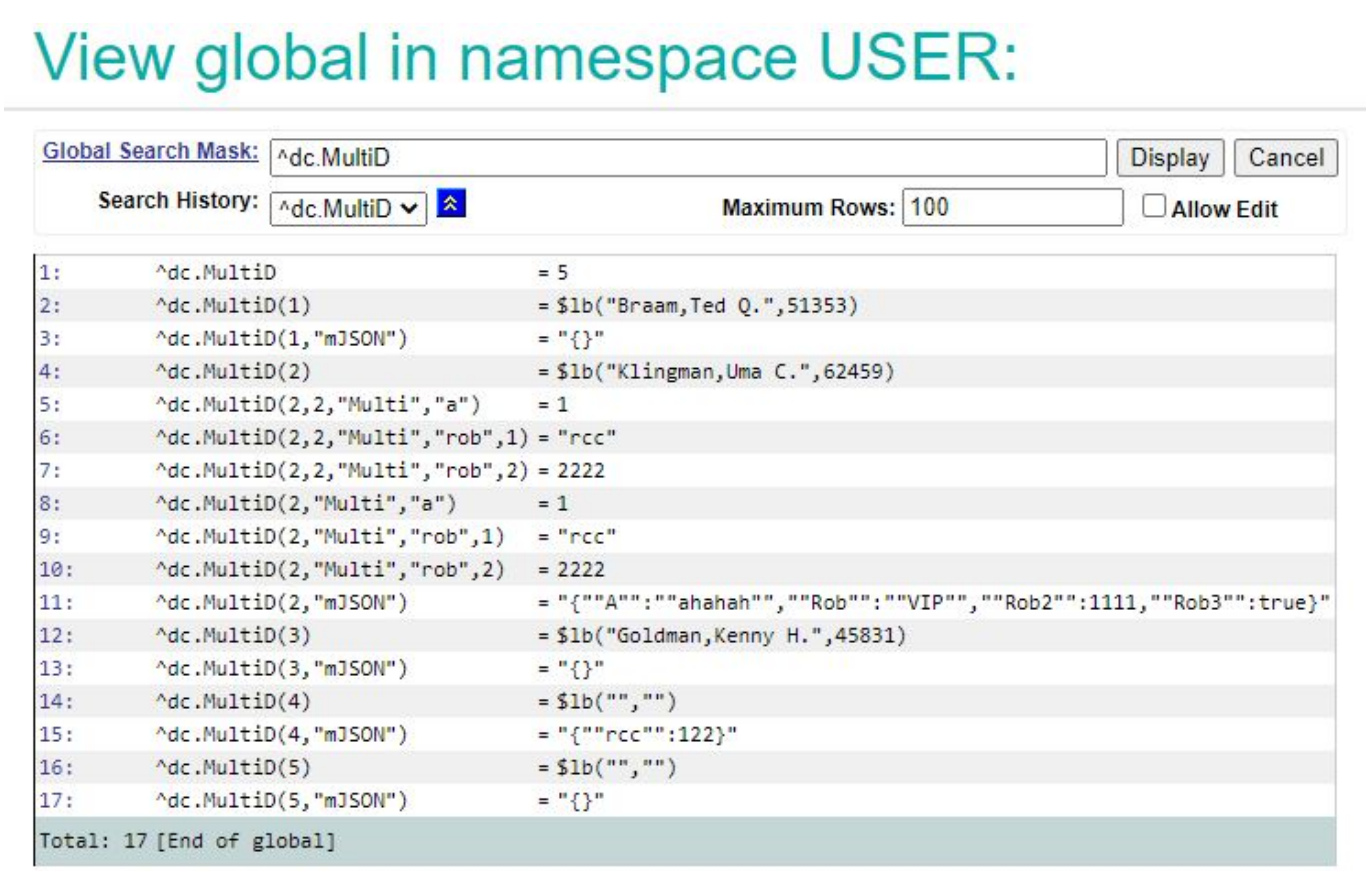
Open Exchange

GlobalToJSON-embedded-Python

Este es un paquete para exportar un Global a un archivo de objeto JSON y volver a crearlo recargando desde este archivo
embeddedPython se refiere a las nuevas tecnologías disponibles. Debe entenderse como un ejercicio de aprendizaje de
cómo manejar las diferentes interfaces. Solo los nodos de Global que contienen datos se presentan en el archivo JSON generado.



Exportamos este Global



Este es el contenido del archivo.

```
{"gbl":[
{"node": "^dc.MultiD", "val": "5"}
{"node": "^dc.MultiD(1)", "val": "$lb(\"Braam,Ted Q.\",51353)"}
{"node": "^dc.MultiD(1,\"mJSON\")", "val": "{}"}
{"node": "^dc.MultiD(2)", "val": "$lb(\"Klingman,Uma C.\",62459)"}
{"node": "^dc.MultiD(2,2,\"Multi\",\"a\")", "val": "1"}
{"node": "^dc.MultiD(2,2,\"Multi\",\"rob\",1)", "val": "rcc"}
{"node": "^dc.MultiD(2,2,\"Multi\",\"rob\",2)", "val": "2222"}
{"node": "^dc.MultiD(2,\"Multi\",\"a\")", "val": "1"}
{"node": "^dc.MultiD(2,\"Multi\",\"rob\",1)", "val": "rcc"}
{"node": "^dc.MultiD(2,\"Multi\",\"rob\",2)", "val": "2222"}
{"node": "^dc.MultiD(2,\"mJSON\")", "val": "{\"A\":\"ahahah\",\"Rob\":\"VIP\",\"Rob2\":1111,\"Rob3\":true}"}
{"node": "^dc.MultiD(3)", "val": "$lb(\"Goldman,Kenny H.\",45831)"}
{"node": "^dc.MultiD(3,\"mJSON\")", "val": "{}"}
{"node": "^dc.MultiD(4)", "val": "$lb(\"\",\"\")"}
{"node": "^dc.MultiD(4,\"mJSON\")", "val": "{\"rcc\":122}"}
{"node": "^dc.MultiD(5)", "val": "$lb(\"\",\"\")"}
{"node": "^dc.MultiD(5,\"mJSON\")", "val": "{}"}
]}
```

El Loader relacionado crea exactamente el mismo Global

El ejemplo es una clase en la que se mezclan Intersystems Object Script y Embedded Python.
Así que es un código híbrido. Intersystems Object Script se usa para iterar a través de Global con $QUERY().
También escribe datos con una estructura $LISTBUILD(). El objeto JSON corresponde así al modelo **Eficiente.**

Embedded Python se utiliza para escribir y leer el archivo de datos. Y también se genera el objeto JSON
y su resolución se realiza con Embedded Python.

Una tarea especial es la conversión correcta de una estructura de subíndice convencional **(sub,sub,sub,..)**
en una lista de Python **[sub,sub,sub, ...]**. Esta lista de Python es obligatoria para cualquier acceso directo
a los nodos de Global.
¡Y al nodo superior del Global, que por definición no tiene subíndice, se accede con esta lista especial **[None]**!

Puede ser beneficioso mirar el código en detalle.

Vídeo (en inglés)

#Globals #JSON #InterSystems IRIS
Ir a la aplicación en InterSystems Open Exchange