

## “Dynamic Vision App for HALCON”

De procesamiento de imágenes HALCON a app de visión artificial IDS NXT en muy pocos pasos



**Ejecute su procesamiento de imágenes desarrollado con HALCON directamente en una cámara basada en apps de visión artificial de IDS sin necesidad de programar una app de visión artificial IDS NXT específica. Ya no es necesario dedicar tanto esfuerzo al desarrollo de apps de visión artificial en C++. La "Dynamic Vision App for HALCON" lo hace por usted. Es el complemento perfecto de su desarrollo del procesamiento de imágenes.**

En este consejo técnico le explicamos cómo crear en unos pocos pasos una biblioteca HALCON adecuada en HDevelop y cargarla y ejecutarla de forma sencilla en un IDS NXT veces utilizando el IDS NXT Cockpit y sin tener que escribir una sola línea de código C++. El sencillo ejemplo de procesamiento de imágenes "*example\_code2d.hdp*" busca códigos de matriz de datos en los datos de imagen guardados y los marca en las pantallas de resultados de la app de visión artificial. Los contenidos de los códigos también se generan como texto. Mediante el desarrollo personalizado de apps de visión artificial el framework IDS NXT, desarrollado por IDS, le permite ejecutar los procesos que desee en las cámaras basadas en apps de visión artificial. El IDS NXT Vision App Development Kit le ayuda a desarrollar y ejecutar su app de visión artificial personalizada; para ello solo necesita tener conocimientos de C++. Sin embargo, el nuevo "Dynamic Vision App for HALCON" se lo pone aún más fácil. Es el complemento perfecto de su desarrollo del procesamiento de imágenes. Si solo desea ejecutar el procesamiento de imágenes con la biblioteca HALCON preinstalada, esta app genérica le ahorra el pesado trabajo de crear la app de visión artificial en C++. Y no solo ejecuta el procesamiento de imágenes HALCON escrito en HDevelop, sino que también proporciona los parámetros de entrada y de salida definidos por medio de la interfaz REST. La app de visión artificial "Dynamic Vision App for HALCON" muestra además estos parámetros en el IDS NXT Cockpit, lo que le permite configurar y comprobar cómodamente su procesamiento de imágenes.

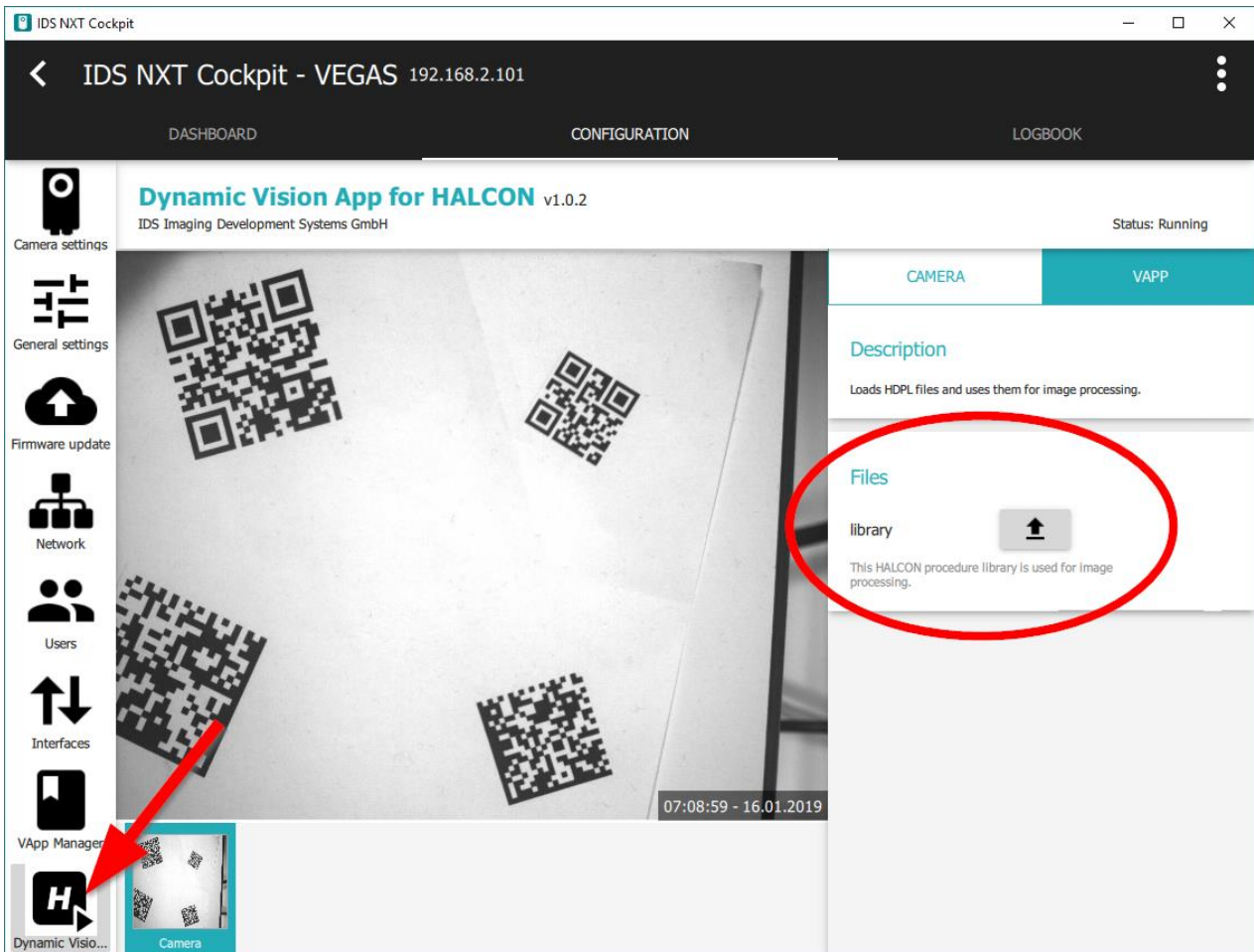


Figura 1 Ejecución directa de bibliotecas HALCON.

## Requisitos

Antes de iniciar el procedimiento descrito aquí y para comprender mejor el concepto IDS NXT es conveniente que consulte el [manual "IDS NXT - Desarrollo de apps de visión artificial"](#) y los [tutoriales de IDS NXT](#). Para poder seguir este consejo técnico necesita lo siguiente:

- IDS NXT vegas (firmware a partir de la versión 1.3)
- PC Windows (conectado por red a IDS NXT vegas)
  - Entorno de desarrollo con licencia HALCON 13
  - IDS NXT Cockpit (a partir de la versión 1.2.4)
- "Dynamic Vision App for HALCON"

Instale y active "Dynamic Vision App for HALCON" en el IDS NXT vegas mediante IDS NXT Cockpit. Para saber cómo instalar una app de visión artificial consulte el capítulo "Gestor de apps de visión artificial" del ["Manual de uso de IDS NXT"](#).

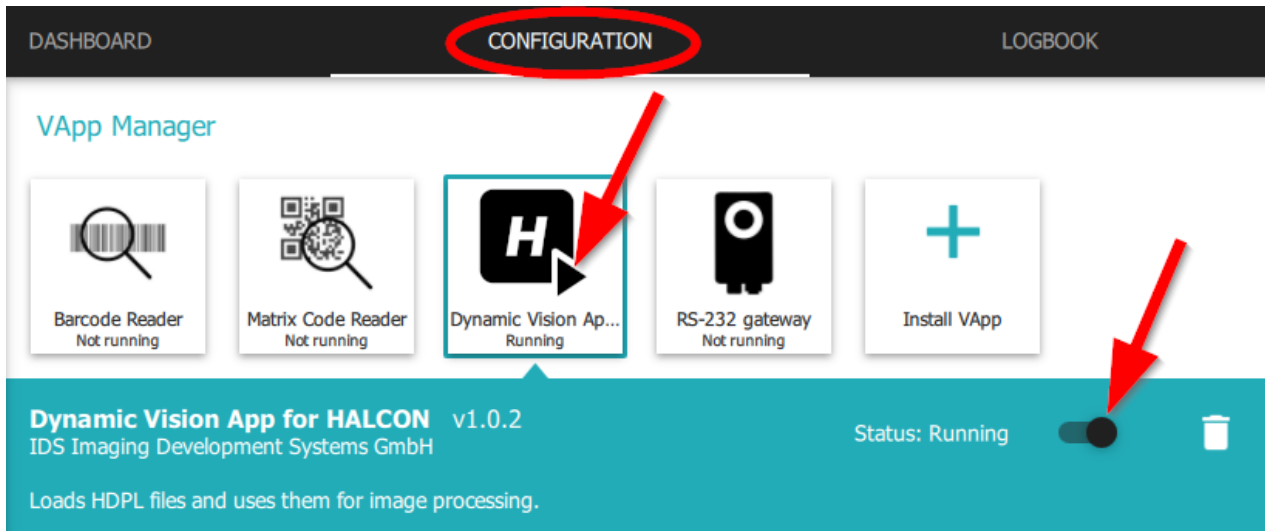


Figura 2 Inicio de "Dynamic Vision App for HALCON" en el VApp Manager

Después de iniciar la app de visión artificial, en el área de configuración del IDS NXT Cockpit aparece el diálogo de configuración. Puede cargar bibliotecas HALCON compatibles cuyo desarrollo se explica a continuación.

**⚠** Recuerde que solo los usuarios administradores pueden instalar y desinstalar apps de visión artificial IDS NXT.

## Desarrollo HALCON

"Dynamic Vision App for HALCON" establece la conexión entre una "HDevelop procedure library" (HDPL) y el framework IDS NXT de una cámara basada en apps de visión artificial de la familia IDS NXT. De ese modo los parámetros de control y los resultados del procesamiento de imágenes se transmiten a la interfaz REST de la cámara y al IDS NXT Cockpit. Esto permite manejar el procesamiento de imágenes directamente como una app de visión artificial completa.

Para garantizar la compatibilidad de la biblioteca HALCON con la app de visión artificial es necesario que existan procedimientos HALCON con nombres determinados, cuyos parámetros de entrada y de salida (interfaz de procedimiento) deben estar perfectamente definidos (tipos semánticos). Solo de ese modo "Dynamic Vision App for HALCON" podrá seleccionar automáticamente las variables de configuración o de resultados correctas que facilitará la interfaz REST.

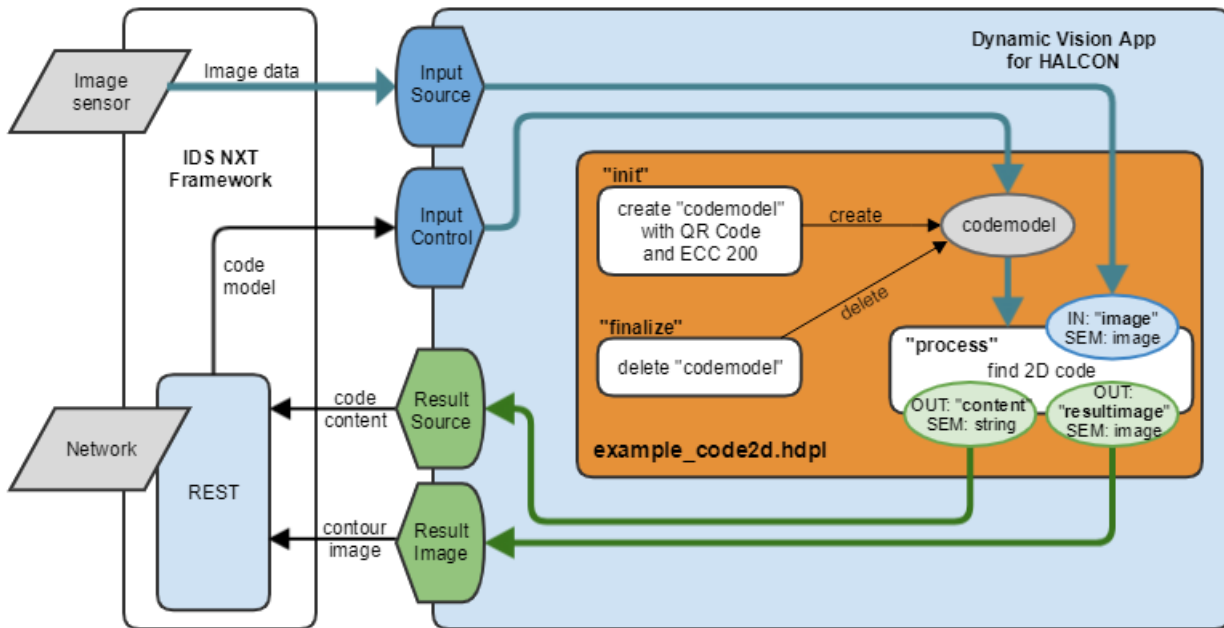


Figura 3 La "Dynamic Vision App for HALCON" establece la conexión entre HALCON y el framework IDS NXT.

## Configuración de la biblioteca HALCON

Para que sea posible la comunicación entre la app de visión artificial y el procesamiento de imágenes HALCON deben crearse los siguientes procedimientos HDPL como puntos de partida:

- **"init"** - se ejecuta al cargar la biblioteca. Aquí se registran InputCtrlParams (variables, parámetros de control) y OutputCtrlParams (visualizaciones, resultados).
- **"finalize"** - se consulta antes de descargar la biblioteca y ordena todas las estructuras de datos creadas en "init".
- **"train"** - entrenamiento en el procesamiento de imágenes HALCON. Este procedimiento es opcional. Si está disponible se llevará a cabo el entrenamiento correspondiente.
- **"process"** - la función de edición del procesamiento de imágenes propiamente dicho.

## Creación de procedimientos

Coloque en HDevelop los procedimientos necesarios del tipo "Procedimiento de biblioteca (.hdpl)" y guárdelos en la misma biblioteca. Indique en cada **procedimiento** los **parámetros necesarios de entrada** o de **salida** (icónicos o de control) y su documentación de parámetro (**semántica**).

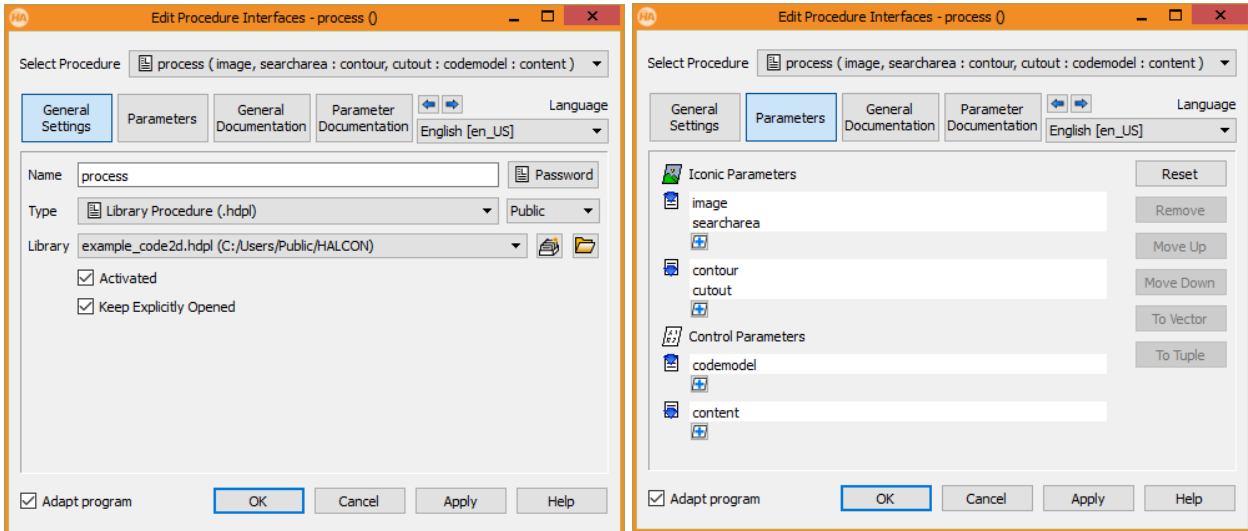


Figura 4 Es imprescindible contar con unas interfaces de procedimiento totalmente definidas, lo que incluye la documentación de parámetro (semántica).

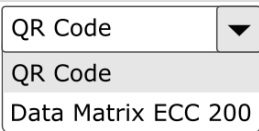



**⚠ Importante:** Deben asignarse los "tipos semánticos" correctos a los parámetros de entrada y de salida de los procedimientos HALCON en la pestaña "Parameter Documentation"!

En el ejemplo de procesamiento de imágenes de este consejo técnico (*example\_code2d.hdp*) debe crear los parámetros siguientes para los procesos:

| Procedimiento | Parámetro  | Tipo                         | Semántica                                    |
|---------------|------------|------------------------------|----------------------------------------------|
| init          | codetype   | Input: Parámetro de control  | cadena (valor: código QR, matriz de ECC 200) |
|               | codemodel  | Output: Parámetro de control | datacode_2d                                  |
| process       | image      | Input: Parámetro icónico     | image                                        |
|               | searcharea | Input: Parámetro icónico     | region                                       |
|               | codemodel  | Input: Parámetro de control  | datacode_2d                                  |
|               | contour    | Output: Parámetro icónico    | xld                                          |
|               | cutout     | Output: Parámetro icónico    | image                                        |
|               | content    | Output: Parámetro de control | cadena                                       |
| finalize      | codemodel  | Input: Parámetro de control  | datacode_2d                                  |
| train         | image      | Input: Parámetro icónico     | image                                        |
|               | firstarea  | Input: Parámetro icónico     | image                                        |
|               | codemodel  | Input: Parámetro de control  | datacode_2d                                  |
|               | content    | Output: Parámetro de control | cadena                                       |
|               | sizemin    | Output: Parámetro de control | integer                                      |
|               | sizemax    | Output: Parámetro de control | integer                                      |

## Semántica de parámetros

La semántica de los parámetros HALCON es muy importante. Con ella usted define qué tipos de parámetros de la app de visión artificial se consultan o se transfieren a los distintos procedimientos HALCON. Mediante la semántica de un parámetro HALCON, la app de visión artificial genera de forma dinámica un parámetro de interacción adecuado para la interfaz REST. En el ejemplo se generan los siguientes parámetros de interacción:

| Parámetro de procedimiento HALCON | IDS NXT Parámetro de interacción                                                    |                                      |
|-----------------------------------|-------------------------------------------------------------------------------------|--------------------------------------|
| codetype                          |    | Combo Box Ctrl                       |
| image                             |    | Imagen                               |
| contour                           |   | Imagen con marca                     |
| content                           | <b>Code content:</b> IDS-Techtip                                                    | Etiqueta                             |
| searcharea                        |  | Región configurable (área de imagen) |

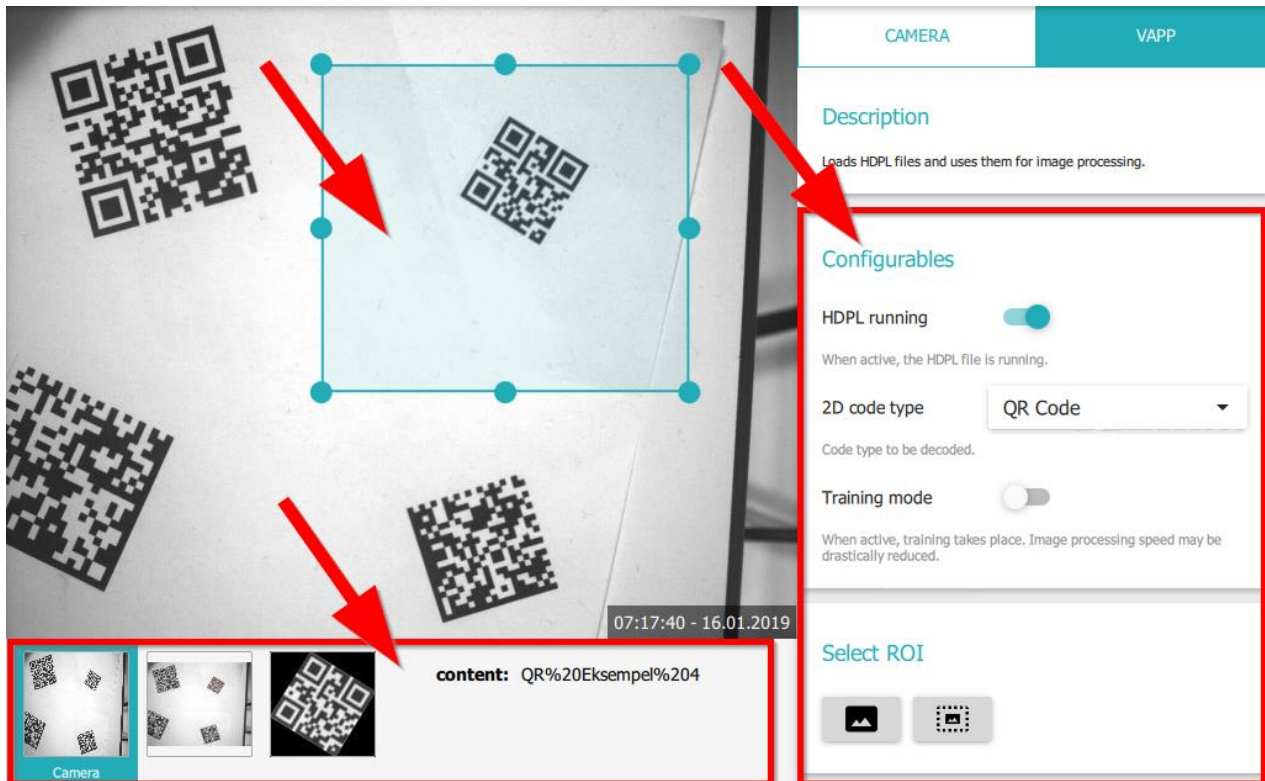


Figura 5 Los parámetros de interacción se muestran como elementos gráficos de la app de visión artificial.

## Procesamiento de imágenes

**i** Este consejo técnico se limita a explicar el intercambio entre la "Dynamic Vision App for HALCON" y la biblioteca HALCON mediante interfaces de procedimientos.

En el procedimiento "init" se crea el parámetro de control "codetype" para seleccionar el tipo de código y se genera un modelo de código 2D, "codemodel", que también se utiliza en otros procedimientos como parámetro de entrada. También se carga o se crea un archivo con datos de entrenamiento para generar un modelo de código detallado.

```
if (FileExistsTraining)
    read_data_code_2d_model(TrainingFilePath, codemodel)
[...]
else
    create_data_code_2d_model(codetype, [], [], codemodel)
```

En el procedimiento "process" primero se reducen los datos de imagen con ayuda de la región ajustada por el usuario "searcharea". Después HALCON busca en esa región códigos 2D utilizando el "codemodel" (entrenado) y transfiere los resultados en la cadena "content" y su posición de imagen como XLD en la imagen "contour". La imagen "cutout" muestra además el código recortado.

```
find_data_code_2d (image, contour, codemodel, [], [], ResultHandles, content)
[...]
reduce_domain(image, Region, cutout)
```

El modelo de código se perfecciona dentro del procedimiento "train" y los datos de entrenamiento se guardan en un archivo.


```
find_data_code_2d(ImageReduced, contour, codemodel, ['train'], ['all'],
ResultHandles, content)
[...]
write_data_code_2d_model(codemodel, TrainingFilePath)
```


El procedimiento "finalize" limpia la memoria borrando el "codemodel".

```
clear_data_code_2d_model(codemodel)
```

## Cargar biblioteca HALCON



Ahora puede utilizar la biblioteca HALCON creada. Utilice el botón "Upload" (  ) de la app de visión artificial para cargar la biblioteca HALCON que se ha creado. Una vez cargada la biblioteca recibirá una notificación. En este caso la app de visión artificial presenta los parámetros de entrada y de salida de la biblioteca HALCON en el IDS NXT Cockpit. El procesamiento de imágenes se puede utilizar inmediatamente.

La biblioteca HALCON sigue siendo residente en memoria después de reiniciar hasta que se elimina del sistema con el botón "Eliminar" (  ) o hasta que se sobrescribe con otra biblioteca. Solo puede haber una biblioteca cargada y activa. No obstante, puede cambiar el procesamiento de imágenes tantas veces como desee.



## Advertencias

### Rendimiento

La combinación de la app de visión artificial genérica y una biblioteca HALCON no está optimizada en materia de velocidad, por lo que no es comparable a una app de visión artificial programada en C++ e implementada específicamente. El ejemplo de este consejo técnico *example\_code2d.hdpi* alcanza alrededor del 70 % de la velocidad de la app de visión artificial especializada "lector de códigos de matriz". No obstante, "Dynamic Vision App for HALCON" ofrece la ventaja de poder probar de forma muy rápida y sencilla el procesamiento de imágenes HALCON en una cámara basada en apps de visión artificial sin necesidad de escribir ni una sola línea de código fuente C++ ni de tener que hacer complejas compilaciones cruzadas para la plataforma de la cámara.

### Exception Handling

Si durante la ejecución de un procedimiento HALCON aparece una excepción no controlada (unhandled exception) el app de visión artificial se reinicia y descarga la biblioteca HALCON. Para evitar la caída asegúrese de que se controlan todos los errores (Exception Handling mediante try...catch) dentro de los procedimientos HALCON.

## Consejo

### Evitar el reinicio

En caso de caída del app de visión artificial durante el desarrollo de la biblioteca de procesamiento de imágenes HALCON conviene evitar que la app se reinicie automáticamente de forma constante. Para ello puede utilizar un comando REST. Después puede decidir manualmente cuándo se debe iniciar la app.

```
curl -u <user credentials> -X PUT  
/vapps/activated/dynamic_vision_app_for_halcon?temporary=true <device URL>
```

**i** Atención: dicho comando REST solo se puede usar una vez en cada instancia de ejecución de la app de visión artificial.

## Mostrar archivo de registro

Escriba la siguiente URL en su navegador para mostrar un registro detallado del "Dynamic Vision App for HALCON" con el fin por ejemplo de hacer el seguimiento de un error de la app de visión artificial o del procesamiento de imágenes:

```
http://<Geräte IP>/vapps/dynamic_vision_app_for_halcon/log
```

```
644-644 [13:31:17] . [dynamic_vision_app_for_halcon] 'Dynamic Vision App for HALCON' up and running.
644-644 [08:58:41] . [MyVApp] Loading engine...
644-644 [08:58:44] . [MyEngine] Updating library...
644-644 [08:58:45] . [MyEngine] Installing Configurables...
644-644 [08:58:45] . [MyEngine] Found Configurable codetype with semantic type "string" and default value "QR Code"
644-644 [08:58:45] . [MyEngine] Configurable codetype processed.
644-644 [08:58:45] . [MyEngine] Running init procedure for the first time...
644-644 [08:58:45] . [MyEngine] Initializing...
644-644 [08:58:45] . [MyEngine] Executing init procedure...
644-644 [08:58:45] . [MyEngine] Init procedure executed. Extracting data...
644-644 [08:58:45] . [MyEngine] Init procedure initialized parameter "codemodel"
644-644 [08:58:45] . [MyEngine] Installing ResultImages...
644-644 [08:58:45] . [MyEngine] Reading ResultImage "contour" ...
644-644 [08:58:45] . [MyEngine] Semantic type of "contour" is "xld"
644-644 [08:58:45] . [MyEngine] ResultImage "contour" processed as Contour Image.
644-644 [08:58:45] . [MyEngine] Reading ResultImage "cutout" ...
644-644 [08:58:45] . [MyEngine] Semantic type of "cutout" is "image"
644-644 [08:58:45] . [MyEngine] ResultImage "cutout" processed as HALCON Image.
644-644 [08:58:45] . [MyEngine] Installing ResultSources...
644-644 [08:58:45] . [MyEngine] Reading ResultSource "content" ...
644-644 [08:58:45] . [MyEngine] Semantic type of "content" is "string"
644-644 [08:58:45] . [MyEngine] Semantic type mapped to "String"
644-644 [08:58:45] . [MyEngine] ResultSource "content" processed.
644-644 [08:58:45] . [MyEngine] Installing Configurables...
644-644 [08:58:45] . [MyEngine] InputCtrlParameter "codemodel" is taken from initialization procedure.
644-644 [08:58:45] . [MyEngine] Installing ROIs
644-644 [08:58:45] . [MyEngine] Found IconicInputParam "image"
644-644 [08:58:45] . [MyEngine] Found IconicInputParam "searcharea"
644-644 [08:58:45] . [MyEngine] Found ROI "searcharea" with semantic type "region"
644-644 [08:58:45] . [MyEngine] Loaded procedures:
644-644 [08:58:45] . [MyEngine] Init: true
644-644 [08:58:45] . [MyEngine] Processing: true
644-644 [08:58:45] . [MyEngine] Training: true
644-644 [08:58:45] . [MyEngine] Finalize: true
```

Figura 6 El extracto del archivo de registro muestra un proceso de carga sin errores de una biblioteca HALCON.

El "Dynamic Vision App for HALCON" analiza los parámetros de procedimiento definidos en la biblioteca y crea los elementos de interacción necesarios de la app de visión artificial (ver marcas amarillas) como "Configurables", "Inputs" y "Results". Este registro de la aplicación permite identificar y solucionar los errores que se producen en la ejecución del procesamiento de imágenes con HALCON.

Utilizando el archivo de salida estándar también se pueden integrar en este registro mensajes procedentes de HALCON.

```
open_file('standard', 'output', stdout)
result := 'NIO'
fwrite_string (stdout, result + '\n')
```

## Conclusión

El "Dynamic Vision App for HALCON" es una forma sencilla y rápida de ejecutar el procesamiento de imágenes basado en HALCON en una cámara basada en apps de visión artificial como IDS NXT vegas. Esto le evita tener que invertir largas horas en la programación de una app de visión artificial propia con C++ y le permite concentrarse en la creación de las rutinas HALCON. Gracias a su sencillo mecanismo de carga puede variar tantas veces como desee en el tiempo de ejecución la tarea de la cámara IDS NXT basada en apps de visión artificial. Para ello no necesita conocimientos ni del código C++ ni de programación de apps de visión artificial. Procesamiento de imágenes embebido, "it's so easy"!

Para obtener más información sobre el concepto de apps de visión artificial IDS NXT y sobre su uso tiene a su disposición el [Manual de desarrollo de apps de visión artificial IDS NXT](#) y los [tutoriales IDS NXT](#). Si desea empezar a programar de forma sencilla e intuitiva [su propia aplicación de visión artificial HALCON](#) consulte las páginas de producto HALCON de MVTec.

Para conocer más consejos técnicos y campos de aplicación [visite la página web de IDS](#). En ella encontrará más información sobre [la nueva familia de cámaras industriales IDS NXT basada en apps de visión artificial](#).

## Autor

Dipl.-Ing. Heiko Seitz  
Redactor técnico

IDS Imaging Development Systems GmbH  
Dimbacher Str. 6-8  
74182 Obersulm  
Alemania

Tel.: +49 7134 96196-0  
E-mail: [h.seitz@ids-imaging.de](mailto:h.seitz@ids-imaging.de)  
Web: [www.ids-imaging.com](http://www.ids-imaging.com)

© 2019 IDS Imaging Development Systems GmbH