

Asignación de un idioma por código de carácter

Si las codificaciones de letras de diferentes idiomas no coinciden, se puede asignar un idioma por letra. Esto resolverá el problema de especificar incorrectamente el idioma en el texto recibido para maquetación, si, por ejemplo, se sabe de antemano que en el archivo solo hay palabras en ruso e inglés, y el marcado del texto es tal que supuestamente está todo en ruso.

Las letras rusas en la búsqueda GREP se definen de la siguiente manera: `[а-яА-ЯёЁ]`; para buscar las letras en inglés se usa `[a-zA-Z]`.

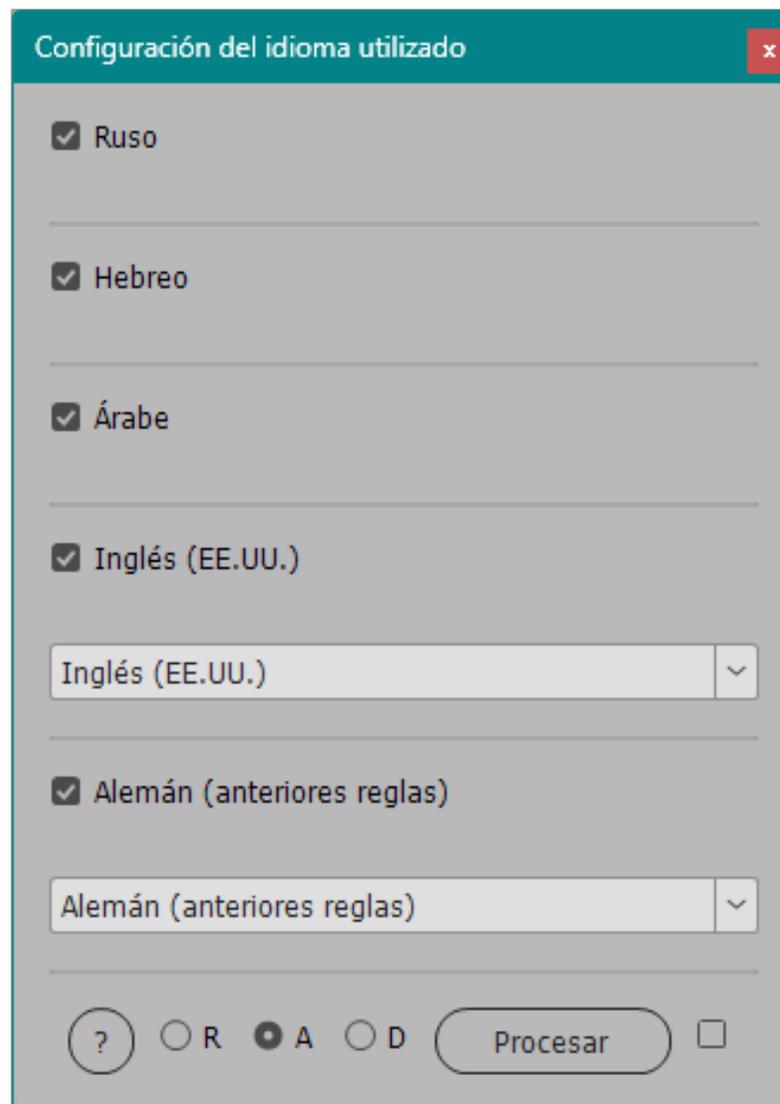
El grupo de caracteres hebreo se encuentra definido con la línea `[\x{0590}-\x{05FF}]` y la escritura árabe se reúne en el siguiente rango: `[\x{0600}-\x{06FF}]`.

Estos espacios no se intersecan, lo que significa que puede buscar letras en ellos espacios y asignarles su propio idioma.

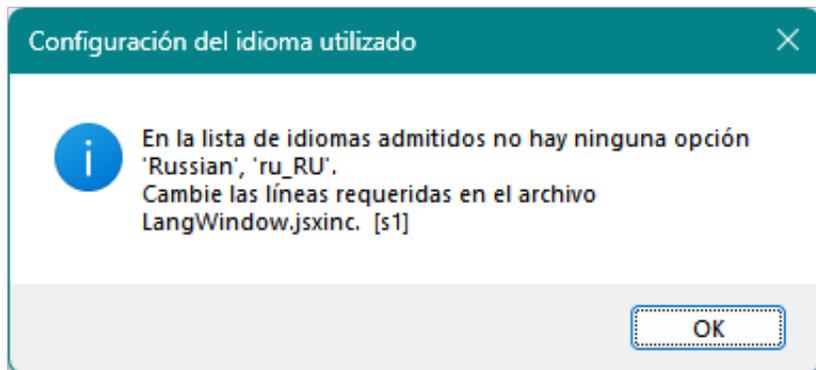
Esta idea es perfectamente adecuada para las letras del alfabeto ruso, hebreo, árabe y devanagari, aunque es obvio que no funciona automáticamente para todos los idiomas europeos. Pero si se sabe de antemano que el texto escrito con caracteres provenientes del latín es, por ejemplo, italiano, entonces sería bueno poder asignarle el atributo de ese idioma a dichas letras.

Hay otro problema con las lenguas europeas: el conjunto de letras mencionado anteriormente no es suficiente para todos. Por lo tanto, para la lengua alemana es necesario tener un conjunto de signos como el siguiente: `[a-zA-ZäöüÄÖÜß]`. El noruego, el checo y otros idiomas tienen sus propios conjuntos de letras especiales, pero si se logra determinar el rango necesario de los signos se podrán identificar las letras de estos idiomas.

La tarea no es fácil, pero está resuelta. La ventana de trabajo del programa **SetLanguageByCharCode.jsx** se encuentra a la derecha.



En la primera ejecución, en lugar de la ventana esperada, puede aparecer el siguiente mensaje en la pantalla:



Este guion, con configuraciones para trabajar con el idioma ruso, fue publicado para la versión en inglés de InDesign y hace falta cambiar su definición en el archivo **LangWindow.jsxinc**.

Así es como se ve el bloque de parámetros para el idioma ruso:

```
18 // русский язык
19 var lng1_name = "Русский";
20 //var lng1_name = "Russian";
21 var lng1_icuLN = "ru_RU";
22 var lng1_untranslatedName = "Russian";
23 var lng1_help = "[а-яА-ЯёЁ]";
```

lng1_name es el nombre del idioma utilizado en la localización de InDesign [en español debe consignarse "Ruso"].

lng1_icuLN es el nombre abreviado.

lng1_untranslatedName es la versión única del nombre del idioma para todas las localizaciones.

lng1_help el espacio de búsqueda de los caracteres de ese idioma.

Hay un bloque de este tipo para cada idioma. Es necesario especificar los valores correctos en todos los bloques para _name y _icuLN.

Hay un guion para obtener estos parámetros para la versión utilizada de InDesign: **All_InDesign_Languages.jsx**.

Después de esto, la ventana del programa aparecerá en la pantalla.

La información sobre el espacio de búsqueda se muestra en los globos de ayuda emergentes para seleccionar el idioma que se va a procesar, seguida de una línea vertical, la variante del nombre de este idioma (que es el mismo en todas las localizaciones de InDesign) y una designación corta del idioma.



✓ German: Old Rules

[a-zA-ZäöüÄÖÜß] | German: Traditional, de_DE

Todos los ejemplos anteriores, que muestran los globos de información, se hicieron en una de las últimas versiones. En todos los casos hay una única versión del nombre del idioma para todas las localizaciones. Antes de la versión de 2015 no existía dicho parámetro en la configuración del idioma.

Debajo se muestra un ejemplo del globo emergente en CS6; solo hay una designación en letras del idioma.

Русский

[a-яА-ЯёЁ] | ru_RU

En el caso de las dos opciones inferiores, el idioma se selecciona en las listas desplegadas. InDesign tiene diferentes variantes del inglés y del alemán, y lo mismo ocurre con el francés o con los idiomas que se utilizan en la India. Es razonable suponer que sea posible que deba asignarle un atributo de idioma específico al texto, y el guion ofrece esta opción.

Como es obvio, el idioma elegido debe coincidir con la codificación utilizada en la versión. Así que en este ejemplo, para la penúltima opción, se puede elegir entre inglés o el latín clásico (y hasta el italiano). Pero en español o en checo hay letras que faltan en la codificación utilizada (en español la cadena sería [a-zA-ZáéíóúñÁÉÍÓÚÑ]).

En la última línea se puede seleccionar cualquiera de las variantes del idioma alemán.

Por lo tanto, la clave para un correcto marcado de un idioma es la codificación correcta de los caracteres. Y esto se puede cambiar en este programa. Por ejemplo, en lugar de buscar letras alemanas puede incluir el alfabeto devanagari, que se utiliza en hindi [o el español, agregando el bloque con la cadena antes indicada]. No es difícil si sabe cómo se edita el código de un guion para InDesign.

Así se ve la instrucción de procesamiento del idioma alemán en el archivo **LangWindow.jsxinc**:

```
46 // немецкий
47 //var lng5_name = "Немецкий: Старые правила"; // rus
48 var lng5_name = "German: Old Rules"; // eng
49 var lng5_icuLN = "de_DE";
50 var lng5_untranslatedName = "German: Traditional";
51 var lng5_help = "[a-zA-ZäöüÄÖÜß]";
```

El bloque comentado para devanagari está en el mismo archivo:

```
53 /*
54 // Подключение обработки деванагари вместо немецкого языка
55 var lng5_name = "Хинди (Индия)"; // rus
56 // var lng5_name = Hindi (India); // eng
57 var lng5_icuLN = "hi_IN";
58 var lng5_untranslatedName = "hi_IN";
59 var lng5_help = "[\x{0900}-\x{097F}]";
60 */
```

Podría esperarse que la vista de la ventana cambie con solo comentar el bloque con el idioma alemán y activar el procesamiento de devanagari. Pero lo más probable es que esto no suceda, sino que en la pantalla aparecerá la antigua ventana, con la opción del idioma alemán. No es un error, sino una característica del programa; al iniciarlo, la configuración de las cinco casillas de verificación se toma del archivo guardado de la configuración de inicio anterior.

La carpeta **#sets** se crea en la del archivo procesado y contiene un archivo **#sets.ini**, del que se toman los datos de diseño de la ventana. Para habilitar la configuración predeterminada (es decir, la incluida en el programa) haga clic en la casilla de verificación a la derecha del botón **Procesar**. Y si hubo un reemplazo del grupo de letras alemanas con el alfabeto devanagari, entonces el botón inferior será así:



Lo que se procesará se determina mediante uno de los tres botones de opción: **R** es el texto resaltado, **A** el artículo y **D** el documento o archivo completo. En la opción **Texto resaltado** el área procesada se extiende hasta los bordes de los párrafos que caen en la selección. Al procesar, el nombre del botón de arranque cambia a **Procesando**. Además, en la pantalla aparecerá una barra de progreso que contiene información sobre lo que se está procesando en ese momento. En un artículo el procesamiento de letras se realiza tantas veces como casillas de verificación haya. La barra de progreso contendrá información sobre qué letras del idioma se están procesando (p.ej., Letras: ruso). Cuando se hayan procesado todas las letras, tendremos la siguiente situación: las letras del artículo, las notas a pie de página y las tablas

tienen un atributo de idioma correspondiente a su codificación, pero el atributo de idioma no ha cambiado para los números y signos de puntuación. Por lo tanto, al final del procesamiento del artículo se realiza otro pasaje por el texto, buscando números y signos de puntuación, y asignándoles el atributo de idioma deseado. Este problema se resuelve de la siguiente manera: en cada párrafo se busca la primera letra y todos los caracteres de este párrafo tendrán su atributo de idioma. Durante este paso habrá un mensaje en la barra de progreso (Puntuación, números, espacios: todos los idiomas).

También hay un mensaje para trabajar con tablas, pero normalmente se procesan tan rápido que uno no se da cuenta.

Cuando se completa el procesamiento reaparece el nombre del botón anterior (**Procesar**).

Si decide complementar el programa con sus propias opciones, lo mejor es contar con diferentes versiones para diferentes idiomas. Es recomendable que tengan sus propios archivos de configuración, como se indica en **SetLanguageByCharCode.jsx** (ver la figura de abajo).

En la versión CS6 no siempre es posible hacer coincidir de inmediato el nombre de los botones inferiores que se están modificando con el idioma seleccionado. Puede resultar que se haya seleccionado un nuevo idioma en la lista, pero que el nombre de la casilla de verificación siga siendo el mismo. En este caso debe desmarcar la casilla de verificación y volver a marcarla, y luego confirmar la selección nuevamente. Esto suele ayudar.

Mikhail Ivanyushin
<https://shop.dotextok.ru/en/>
dotextok@gmail.com

```
129 var rezFileName = "#sets.ini"; // you can have variations of this script
130 // for different sets of processed languages.
131 // And these options should differ by the name of the installation file.
132 | #sets1.ini, #sets2. ini, etc.
```