

C.U.F.
(C)ATRE (U)pgrade (F)ormat

Operaciones necesarias para especificar un
cambio de versión en CATRE

Versión: 0.9

3 de marzo de 2004

Índice general

1. Introducción	3
1.1. Estructura de la clasificación	3
1.2. Genealogía de áreas	4
2. Operaciones permitidas	6
2.1. Operaciones básicas con recursos	6
2.1.1. Mover recursos a otra área (no recursivo)	
rec_mv	6
2.1.2. Eliminar recursos	
rec_rm	6
2.2. Operaciones simples con áreas	7
2.2.1. Añadir un área hoja	
area_leaf_add	7
2.2.2. Eliminar un área	
area_simple_rm	7
2.2.3. Cambio de área	
area_maincode_mv	7
2.2.4. Cambiar de tía un área	
area_nephew_mv	7
2.2.5. Eliminar un área sobrina	
area_nephew_rm	8
2.3. Operaciones complejas con áreas	8
2.3.1. Eliminar un área hoja	
area_leaf_rm	8
2.3.2. Mover un área hoja	
area_leaf_mv	9
2.3.3. Dividir un área hoja en varias áreas	
area_leaf_split	9
2.3.4. Eliminar una rama que cuelga de un área	
area_tree_rm	10
2.3.5. Mover una rama que cuelga de un área	
area_tree_mv	10
2.3.6. Agrupar áreas hoja hermanas	
area_leaf_merge	11

<i>ÍNDICE GENERAL</i>	2
3. Migración	13
3.1. Versiones de CATRE	13
3.2. Proceso a seguir para una actualización	13

Capítulo 1

Introducción

En este documento pretendemos dar una serie de especificaciones sobre el proceso a llevar a cabo en una migración de versión de la clasificación CATRE de RedIRIS.

La persona encargada de generar la nueva versión de CATRE deberá proporcionar un fichero donde se indiquen las relaciones existentes entre la clasificación en curso y la que propone. Este fichero debe contener una serie de operaciones que indiquen cómo se pasa de una versión a otra. Las operaciones que se pueden realizar se detallan en este documento.

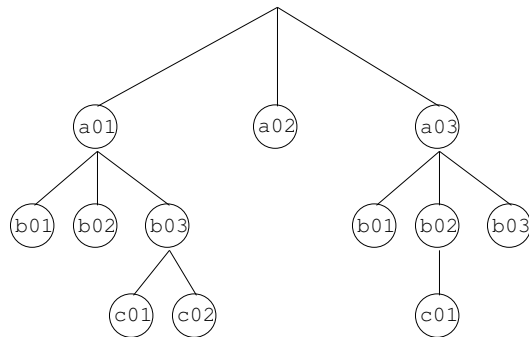
Para evitar inconsistencias en la base de datos de áreas/recursos hay que seguir estas especificaciones al pie de la letra.

1.1. Estructura de la clasificación

La clasificación CATRE se ha estructurado, al igual que las ramas del conocimiento, de manera jerárquica en un árbol invertido. De esta forma las áreas más generales están en los niveles superiores y las más específicas en los inferiores. Cada área en el árbol es llamada nodo y los nodos finales, que no tienen más hijos, son denominados hojas.

Para representar esta clasificación jerárquica de manera textual hemos definido unos códigos especiales basados en grupos de tres caracteres, una letra y dos números (por ejemplo a01b21c02 ...). Cada letra identifica un nivel en el árbol de áreas temáticas (a para el nivel superior, b para el nivel inferior a a y así sucesivamente) y el número identifica un área entre las del mismo nivel.

A continuación vamos a mostrar un ejemplo de árbol CATRE. Hemos puesto un texto dentro de cada nodo. El código de un área determinada será la composición de los nombres que se encuentran en los círculos de los nodos si recorremos el árbol desde la parte superior hasta el nodo a tratar.



Según lo expuesto anteriormente el nodo a01 es un nodo intermedio y a02, a01b02 y a03b02c01 son nodos hoja.

1.2. Genealogía de áreas

Para familiarizarnos con la notación empleada en el nombrado de nodos de la estructura jerárquica damos, a continuación, una breve explicación de cada término.

- | | |
|----------|---|
| madre | área que tiene nodos que cuelgan de ella. |
| hija | área que cuelga del área madre. Tendrá un <code>catreMainCode</code> igual que el área madre añadiendo una letra y dos números. |
| tía | área que tiene en su campo <code>catreMainCode</code> el mismo código que otras áreas (sobrinas) tienen en sus campos <code>catreCode</code> . En el mundo real no sería un área tía sino una madrina ya que las supuestas sobrinas no son hijas de sus hermanas. |
| sobrina | área que tiene en su campo <code>catreCode</code> el código <code>catreMainCode</code> de otra área. En el mundo real no sería un área sobrina ya que no es la hija de un área hermana. Sería más bien un área ahijada. |
| hermanas | áreas que se encuentran bajo una misma madre. Su <code>catreMainCode</code> se diferencia en los tres últimos caracteres. |

En el presente gráfico mostramos algunas de las relaciones que un nodo puede tener.

Capítulo 2

Operaciones permitidas

2.1. Operaciones básicas con recursos

2.1.1. Mover recursos a otra área (no recursivo)

rec_mv

Formato: `rec_mv catreCode catreCodeNew`

Ejemplo: `rec_mv a02b2 a01b01c01`

- Cambiar el código `catreCode=a02b02` por `catreCode=a01b01c01` en todos los recursos clasificados en `a01b02`. Si hay recursos clasificados por debajo de `a02b02` (por ejemplo `a02b02c03`) no serán movidos.

Nota: No es un movimiento recursivo

2.1.2. Eliminar recursos

rec_rm

Formato: `rec_rm catreCode`

Ejemplo: `rec_rm a01b01`

- Eliminamos los recursos que tengan únicamente el valor `a01b01` en campo `catreCode`.
- Borramos el valor `a01b01` del campo `catreCode` de aquellos recursos que lo tengan multivaluado.

2.2. Operaciones simples con áreas

2.2.1. Añadir un área hoja `area_leaf_add`

Para crear una nueva área se utiliza la operación `area_leaf_add`.

Formato: `area_leaf_add catreMainCode`

Ejemplo: `area_leaf_add a01b20`

Notas: Habrá que tener en cuenta si algún recurso del área a01 puede ser clasificado en la nueva área a01b20

Si algún área puede estar relacionada con la que acabamos de crear hay que añadir a su campo `catreCode` el código del área tía.

2.2.2. Eliminar un área `area_simple_rm`

Usamos esta operación cuando deseamos eliminar un área

Formato: `area_simple_rm catreMainCode`

Ejemplo: `area_simple_rm a01b01c01`

- Eliminamos el área con `catreMainCode=a01b01c01`.

Nota: No se hace nada más, no se comprueba si tenía sobrinas ni si había recursos catalogados en esta área

2.2.3. Cambio de área `area_maincode_mv`

Formato: `area_maincode_mv catreMainCodeOld catreMainCodeNew`

Ejemplo: `area_maincode_mv a01b01 a02b10`

- Cambiamos el campo `catreMainCode=a01b01` por el `catreMainCode=a02b10`.

2.2.4. Cambiar de tía un área `area_nephew_mv`

Lo que vamos a hacer es cambiar un valor del campo `catreCode` por otro en un área determinada. Con esto hemos cambiado de tía a un área.

Formato: `area_nephew_mv [catreMainCode] catreCodeOld catreCodeNew`
El campo `[catreMainCode]` es opcional

Ejemplo: `area_nephew_mv a01b01c01 a01b01 a02b02`

- Buscamos el área con `catreMainCode=a01b01c01` y cambiamos el `catreCode= a01b01` por `catreCode=a02b02`.

Ejemplo: `area_nephew_mv a01b01 a02b02`

- Buscamos cualquier área sobrina de `a01b01` (que tenga `catreCode=a01b01`) y cambiamos dicho valor por `a02b02`.

2.2.5. Eliminar un área sobrina `area_nephew_rm`

Realmente no estamos eliminando el área. Lo que hacemos es eliminar el lazo familiar que existe entre el área tía y la sobrina, es decir, el campo `catreCode` del área sobrina.

Formato: `area_nephew_rm [catreMainCode] catreCode`
El campo `[catreMainCode]` es opcional

Ejemplo: `area_nephew_rm a01b01c01 a01b02c02`

- Buscamos el área con `catreMainCode=a01b01c01` y eliminamos el `catreCode=a01b02c02`

Ejemplo: `area_nephew_rm a01b02c02`

- Buscamos cualquier área sobrina de `a01b02c02` (que tenga `catreCode=a01b02c02`) y eliminamos dicho valor.

2.3. Operaciones complejas con áreas

2.3.1. Eliminar un área hoja `area_leaf_rm`

Usaremos esta operación cuando queramos eliminar un área de último nivel.

Formato: `area_leaf_rm catreMainCode`

Ejemplo: `area_leaf_rm a01b01c01`

- Eliminar el área con `catreMainCode=a01b01c01` mediante `area_simple_rm a01b01c01`
- Eliminar `catreCode=a01b01c01` de todas las áreas sobrinas que pudieran tener ese código en su campo `catreCode` así que si esta área era tía de algunas otras se romperá esa relación. Usaremos la operación `area_nephew_rm a01b01c01`

- Cambiar `catreCode=a01b01c01` por `catreCode=a01b01` a todos los recursos que tuviesen `catreCode=a01b01c01` mediante
`rec_mv a01b01c01 a01b01`

Nota: Si el área a borrar es de nivel 1 (por ejemplo a04) los recursos catalogados en esta área han de ser eliminados si es que no contienen otro código `catreCode`.

2.3.2. Mover un área hoja `area_leaf_mv`

Usamos esta operación cuando vamos a mover un área hoja de sitio. Este movimiento lleva consigo un cambio del código `catreMainCode`.

Formato: `area_leaf_mv catreMainCodeOrig catreMainCodeNew`

Ejemplo: `area_leaf_mv a01b01 a22b33c44`

- Cambiar el código `catreMainCode=a01b01` por `catreMainCode=a11b33c44` en el área que estamos tratando mediante
`area_maincode_mv a01b01 a22b33c44`
- Cambiar `catreCode=a01b01` por `catreCode=a11b33c44` en las áreas sobrinas mediante
`area_nephew_mv a01b01 a11b33c44`
- Cambiar los `catreCode=a01b01` por `catreCode=a11b33c44` en los recursos clasificados en el área original mediante
`rec_mv a01b01 a11b33c44`

Nota: Siempre se ha de mover a una localización nueva, es decir, el campo `catreMainCode` debe no existir previamente.

2.3.3. Dividir un área hoja en varias áreas `area_leaf_split`

Cuando queremos dividir un área de tipo hoja podemos usar la operación `area_leaf_split`. Esta división se puede realizar tanto bajo el área que estemos tratando como en otra localización.

Formato: `area_leaf_split catreMainCodeOrig catreMainCodeNew1
... catreMainCodeNewN`

Ejemplo: `area_leaf_split a04 a04b01 a04b02 a04b03`

- Creamos las nuevas áreas a04b01 a04b02 y a04b03 mediante
`area_leaf_add a04b01`
`area_leaf_add a04b02`
`area_leaf_add a04b03`

- Los recursos que estaban clasificados en a04 se dejan ahí.

Notas: Habrá que ver, posteriormente, qué se hace con esos recursos. Lo interesante sería que fuesen reclasificados por los expertos/documentalistas.

Otra opción puede ser la catalogación de todos los recursos que tenían `catreCode=a04` en las tres áreas que se han creado hasta la espera de la reclasificación, es decir, les añadiríamos `catreCode=a04b01`, `catreCode=a04b02` y `catreCode=a04b03`.

2.3.4. Eliminar una rama que cuelga de un área `area_tree_rm`

Usaremos esta operación cuando queramos eliminar un área y las que cuelguen de ella.

Formato: `area_tree_rm catreMainCode`

Ejemplo: `area_tree_rm a01b01c01`

- Eliminar el área con `catreMainCode=a01b01c01` mediante `area_simple_rm a01b01c01`
- Eliminar todas las áreas hijas de `a01b01c01`, las que tengan `catreMainCode=a01b01c01*` mediante `??????`
- Añadir `catreCode=a01b01` a todos los recursos que tuviesen `catreCode=a01b01c01*` mediante `??????`
- Eliminar `catreCode=a01b01c01*` de todas las áreas sobrinas mediante `?????? area_nephew_rm ??????`

Nota: Si el área a borrar es de nivel 1 (por ejemplo a04) los recursos catalogados en esta área han de ser eliminados si es que no contenían otro código `catreCode`.

2.3.5. Mover una rama que cuelga de un área `area_tree_mv`

Usamos esta operación cuando vamos a mover un área de sitio en la estructura jerárquica de áreas. Este movimiento lleva consigo un cambio del código `catreMainCode`.

Formato: `area_tree_mv catreMainCodeOld catreMainCodeNew`

Ejemplo: `area_tree_mv a01b01 a22b33c44`

- Cambiar `catreMainCode=a01b01` por `catreMainCode=a11b33c44` en el área que estamos tratando mediante
`area_maincode_mv a01b01 a11b33c44`
- Cambiar `catreMainCode=a01b01*` por `catreMainCode=a11b33c44*` en todas las áreas por debajo de la que estamos tratando mediante
`??????`
- Cambiar `catreCode=a01b01` por `catreCode=a11b33c44` en las áreas sobrinas mediante
`area_nephew_mv a01b01 a11b33c44`
- Cambiar `catreCode=a01b01*` por `catreCode=a11b33c44*` en las áreas sobrinas mediante
`?????? area_nephew_mv a01b01* a11b33c44* ??????`
- Cambiar los `catreCode=a01b01*` por `catreCode=a11b33c44*` en los recursos mediante
`?????? rec_mv a01b01* a11b33c44* ??????`

Nota: Siempre se ha de mover a una localización nueva, es decir, el campo `catreMainCodeNew` no debe existir previamente.

2.3.6. Agrupar áreas hoja hermanas `area_leaf_merge`

Mediante esta operación podemos agrupar varias áreas de tipo hoja en una sola

Formato: `area_leaf_merge catreMainCode1
... catreMainCodeN catreMainCodeNew`

Ejemplo: `area_leaf_merge a10b10 a10b11 a10b12 a10b20`

- Crear el área con `catreMainCode=a10b20` mediante
`area_leaf_add a10b20`
- Eliminar las áreas con `catreMainCode=a10b10`, `a10b11` y `a10b12` mediante
`area_leaf_rm a10b10`
`area_leaf_rm a10b11`
`area_leaf_rm a10b12`
- Cambiar los campos `catreCode=a10b10`, `a10b11` y `a10b12` de las áreas sobrinas por `catreCode=a10b20` mediante
`area_nephew_mv a10b10 a10b20`
`area_nephew_mv a10b11 a10b20`
`area_nephew_mv a10b12 a10b20`

- Cambiar los campos `catreCode=a10b10`, `a10b11` y `a10b12` de los recursos por `catreCode=a10b20` mediante
`rec_mv a10b10 a10b20`
`rec_mv a10b11 a10b20`
`rec_mv a10b12 a10b20`

Capítulo 3

Migración

3.1. Versiones de CATRE

Una versión de CATRE viene identificada por la fecha de generación, en formato año, mes, día (YYYYMMDD), el caracter '_' y los códigos x.y.z donde

- x Modificaciones que supongan cambios importantes en el software que usamos para el proyecto RTR. Por ejemplo la adición de un campo nuevo que tenga relación con otros campos existentes (por ejemplo cuando se incorporó `catreCode` hubo que cambiar los `catreCode` antiguos por `catreMainCode`)
- y Modificaciones que supongan cambios estructurales como movimientos de áreas
- z Modificaciones de contenido

Por ejemplo la versión 1.0.6 se generó el día 25 de marzo de 2003. La versión de `catre` será la `2030325_1.0.6`. Si usamos notación urn tendríamos `urn:mace:rediris.es:copa:catre:20030325_1.0.6`

3.2. Proceso a seguir para una actualización

Hay que tener presente que una migración de CATRE de la versión `x1` a la `x2` implica dos procesos principales:

1. Comprobación de la consistencia de la migración `x1-x2`
 - a) Aplicación del fichero de especificaciones PUF `x2.puf` al fichero con el esquema CATRE en formato CSV `x1.csv`
Obtenemos el fichero `x1-x2.csv` con el esquema CATRE después de la migración y el fichero `rec-x1-x2.puf` con las operaciones a aplicar a los recursos para finalizar el proceso de migración

- b) Comprobación de que el fichero `x1-x2.csv` es compatible con el esquema `x2.csv` propuesto

2. Migración

- a) Parada del servidor LDAP para un proceso de mantenimiento
- b) Eliminación del directorio de las áreas de CATRE x1
- c) Volcado de las áreas de `x2.csv`
- d) Actualización de los recursos existentes tomando como referencia el fichero `rec-x1-x2.puf`

Índice alfabético

area_leaf_add, 7
area_leaf_merge, 11
area_leaf_mv, 9
area_leaf_rm, 8
area_leaf_split, 9
area_maincode_mv, 7
area_nephew_mv, 7
area_nephew_rm, 8
area_simple_rm, 7
area_tree_mv, 10
area_tree_rm, 10

rec_mv, 6
rec_rm, 6