**BCF**

**MANUAL DE ADMINISTRACION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Control de cambios** | | | |
| Versión | Fecha | Autor | Detalle de cambios |
| 1 | 2015-06-29 | Ariel Dayler Salazar Hinojosa | Elaboración del documento |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Fecha de Aprobación | Firma |
| Elaborado por: | Ariel Dayler Salazar Hinojosa | Especialista en Software de Telecomunicaciones |  |  |
| Revisado por: |  |  |  |  |
| Aprobado por: |  |  |  |  |

CONTENIDO

1. INTRODUCCIÓN 3

2. ARQUITECTURA 3

3. COMPILACION 3

4. INSTALACIÓN 3

5. ESTRUCTURA DE DIRECTORIOS 5

6. INICIO DETENCIÓN DEL SISTEMA 5

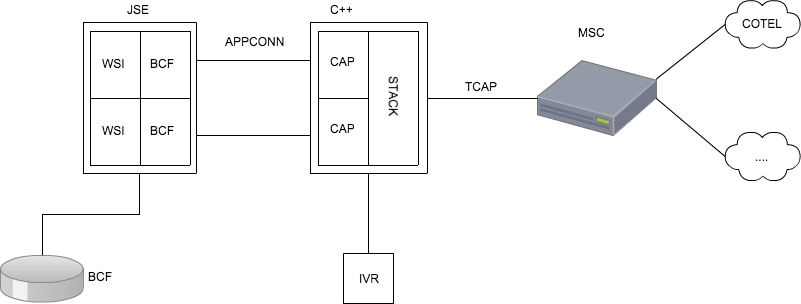
7. ALERTAS 5

8. BASE DE DATOS 5

# INTRODUCCIÓN

El propósito del presente documento describe como instalar, configurar, revisar los registros, y administrar la aplicación BCF.

# ARQUITECTURA



# COMPILACION

En esta sección se describe el procedimiento de compilación de la aplicación BCF.

1. Obtener la última versión del código fuente desde:

<https://10.40.3.230/projects/sptelecom/bcf>

1. En el directorio donde hizo checkout encontrará dos directorios, branches y trunk, diríjase hacia **branches/1.0** (Este documento únicamente es aplicable para la versión 1.0 del BCF).
2. A través de la línea de comandos ejecute: **$ mvn clean install** si necesita obviar los tests de integración **$ mvn –Dmaven.test.skip=true clean install** este último caso no es recomendable.
3. En este mismo directorio (branches/1.0) se habrá generado un directorio llamado build, dentro de este se encuentra un comprimido llamado **bcf-1.0.zip**, que contiene los binarios distribuibles.

# INSTALACIÓN

Obtener los binarios distribuibles, de la aplicación BCF, estos se distribuyen el un comprimido zip llamado **bcf-1.0.zip** que contiene los binarios propios de la aplicación BCF, además necesita los binarios correspondientes al comprimido zip **logger.rotator-1.1.zip**, este último contiene un tool usado para la rotación de las tablas de registro.

* 1. **REQUISITOS**
* Binarios de las aplicaciones BCF y Logger Rotator (bcf-1.0.zip, logger.rotator-1.1.zip)
* Java JDK 8.X o superior
* Acceso a un servidor MySQL 5.6.X o superior
* Un nodo de la aplicación CAP, por cada instancia de la aplicación BCF.
  1. **BCF**

1. Crear el directorio donde se procede a instalar la aplicación. Recomendable usar **/bcf** (que denominaremos a partir de ahora **BCF\_HOME**).

*$ mkdir /bcf*

1. Copiar bcf-1.0.zip al directorio BCF\_HOME.
2. Descomprimir bcf-1.0.zip.

$ cd $BCF\_HOME

$unzip bcf-1.0.zip

1. Añadir permisos de ejecución.

$ chmod +x start\_bcf.10.sh execute\_db\_logger\_rotator.sh

1. Realizar la configuración de las propiedades de la aplicación de acuerdo a lo descrito en sección 10.1.
2. Para inicial la aplicación ejecutar el script **start\_bcf.10.sh.**

**$ ./start\_bcf.10.sh**

1. Verifique el proceso se ha iniciado.

$ ps –fea | grep bcf

* 1. **LOGGER ROTATOR**

1. Crear un directorio donde se instalará la aplicación. Recomendable usar **/log.rot** (que se denominará desde ahora **LOG\_ROT\_HOME**).

$ mkdir /log.rot

1. Copiar logger.rotator-1.1.zip al directorio LOG\_ROT\_HOME.
2. Descomprimir logger.rotator-1.1.zip.

$ cd $LOG\_ROT\_HOME

$ unzip logger.rotator-1.1.zip

1. Crear el directorio **tmp** dentro de **LOG\_ROT\_HOME**. Este es el directorio que contendrá los logs de la aplicación.
2. Configurar la ejecución periódica de los scripts de rotación. Esto se hace a través del cron job de Linux, el script que será ejecutado por esta herramientase encuentra en BCF\_HOME/execute\_db\_logger\_rotator.sh.

i.e. Rotación diaria

$ ln -s $BCF\_HOME/execute\_db\_logger\_rotator.sh /etc/cron.daily/ execute\_db\_logger\_rotator.sh

# ESTRUCTURA DE DIRECTORIOS

|  |  |
| --- | --- |
| /bcf | BCF\_HOME raíz de la aplicación contiene a primer nivel el ejecutable principal, el/los script de iniciostart\_bcf.10.sh, el script de rotación de las tablasexecute\_db\_logger\_rotator.sh. |
| /bcf/lib | Librerías java de las cuales depende la aplicación BCF. |
| /bcf/properties | Propiedades de propósito general para la aplicación BCF. Propiedades para la conexión a la base de datos. |
| /bcf/sql | Script de rotación de la tabla wsi\_record. |
| /bcf/tmp | Contiene los logs del sistema. |

# INICIO DETENCIÓN DEL SISTEMA

Para iniciar la aplicación:

$ ./start\_bcf.10.sh

Para detener el sistema:

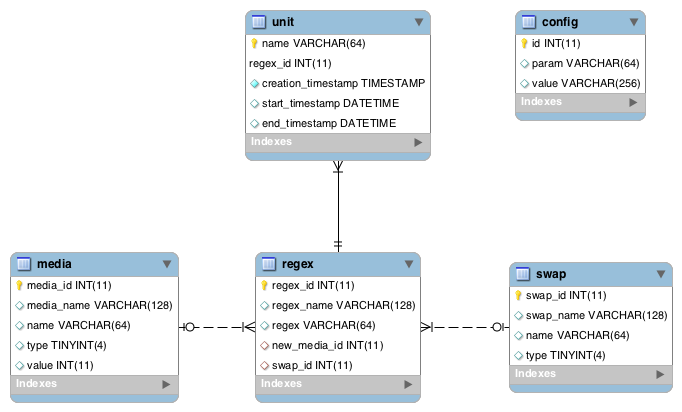
$ kill -9 PID

# ALERTAS

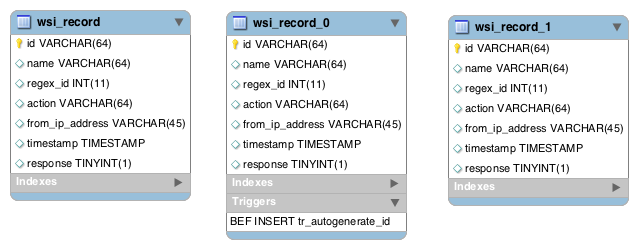
La aplicación puede notificar através de email y SMS cada que se produce un error en el sistema, estps errores son colectados cada 10 minutos. La configuración de los destinos tanto del SMS como del email se realiza en la tabla bcf.record. Para detalles de los campos configurables referirse a la sección 8.1.1.2.

# BASE DE DATOS

* 1. **BCF**



* + 1. **TABLAS**
       1. **UNIT:** Registro de las líneas, con su servicio correspondiente, que se encuentran bloqueadas.
* *name*: número de la línea a bloquear.
* *regex\_id:* Identificador de la regla que se aplicará para el bloqueo (descrito en la tabla regex).
* *creation\_timestamp:*fecha y hora en la que el registro fue insertado en dicha tabla.
* start\_timestamp: Fecha y hora que comienza el bloqueo de la línea (por defecto mismo valor que creation\_timestamp).
* *end\_timestamp:* Fecha y hora en la que termina el bloqueo de la línea. (Por defecto NULL).
  + - 1. **CONFIG:** Registro de las configuraciones necesarias para las alertas.
* *id:* identificador único de la configuración.
* param: Nombre de la configuración.
* value: valor de la configuración.
  + - 1. **MEDIA:** Registro de los audios a reproducirse el caso de bloqueos.
* *media\_id:* identificador único del audio.
* *media\_name:* nombre del audio.
* *name:*número que identifica al ivr donde se encuentra el audio.
* *type:* define que tipo de sesión va a crear.
* *value:* duración en décimas de segundo del audio a reproducirse.
  + - 1. **REGEX:** Registro de las reglas de bloqueo.
* *regex\_id:* identificador único de la regla de bloqueo.
* *regex\_name:* nombre de la regla de bloqueo.
* regex: expresión regular que identifica las reglas de bloqueo.
* *new\_media\_id:* identificador del audio correspondiente a dicha regla.
* *swap\_id:* identificado único para el swap.
  + - 1. **SWAP:** Registro de redirecciones ya sea números o IVR (actualmente se utiliza IVR).
* *swap\_id:* identificador único de la redirección.
* *swap\_name:* nombre de la redirección.
* *name:*número que identifica al ivr a redireccionarse.
* *type:* representa el NAI (nature address indicator) pudiendo tener los valores:3🡪nacional y 4🡪internacional.
  1. **BCF\_RECORD**



* + 1. **TABLAS**
       1. **WSI\_RECORD:** Almacena los registros de bloqueo y desbloqueo.
* *id:* identificador único del registro.
* *name*: numero de la línea que se bloqueo o desbloqueo
* *regex\_id:* identificador de la regla de bloqueo aplicada al name.
* *action*: indicador para saber si la acción que se realizó fue de bloqueo o desbloqueo. Valores:
  + **lock**🡪 bloqueo
  + **unlock**🡪 desbloqueo
* *from\_ip\_address:* dirección ip desde donde se realizo la acción de bloqueo o desbloqueo mediante el web Service.
* timestamp: Fecha y hora en la que se registro el bloqueo o desbloqueo de dicha línea.
* *response:* respuesta obtenida por el web service respecto a si se pudo o no realizar la acción mencionada: Valores:
  + -1🡪Parámetros inválidos
  + -2 🡪Name Inválido
  + -3 🡪 RegexId Inválido
  + -4 🡪 No se puede insertar el Unit
  + -5 🡪 El Unit Name no existe
  + -6 🡪El Unit Name ya existe
  + 1🡪se realizo la acción exitosamente

1. **LOGS**

* bcf-app.log: Logs de la aplicación del día actual. Este archivo va rotando de forma diaria.
* bcf.tmp: En general debe estar vacío, contiene errores de inicialización del sistema. Si la aplicación inicializó de forma correcta este archivo estará vacío.
  1. **CONFIGURACION**

Los logs de la aplicación pueden configurarse a través del archivo *$BCF\_HOME/properties/log4j2.xml,* las propiedades de este archivo se describen en la sección 10.2.

1. **CONFIGURACION**

En esta sección de da una descripción de las propiedades que pueden ser configuradas para la aplicación BCF.

* 1. **PROPERTIES**
     1. **bcf.properties**

Propiedades generales de la aplicación BCF.

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Valor** | **Descripción** |
| bcf.id | 100 | Identificador de la aplicación, debe un número entero y único por cada nodo. |
| bcf.getter | com.nuevatel.bcf.NameGetter10 | Identifica la clase que seusa para obtener el Unit Name del subscriptor, esta clase debe implementar la interface NameGetter y estar disponible en el classpath. |
| bindAddress | all | Lista de IPs, separados por una coma, desde los cuales se permite una conexión hacia el AppServer, las aplicaciones CAP deben estar permitidas. |
| port | 8989 | Puerto por el cual las aplicaciones CAP se comunican con la aplicación BCF. |
| cache.service.concurrencyLevel | 8 | Numero de hilos que simultáneamente pueden acceder al cache de la aplicación. Este número de hilos es definido por servicio. |
| cache.service.expireAfterWrite | 300000 | Tiempo en milisegundos durante el cual un dato esta disponible en la cache, después de una lectura. Aplica para Unit, Regex. |
| cache.service.expireAfterAccess | 900000 | Tiempo en milisegundos durante el cual un dato esta disponible en la cache,despues de ser cargado. Aplica para Unit, Regex. |
| bcf.logrecorder.size | 4 | Número de hilos con los cuales se escribe en el esquema bcf\_record. Actualmente solo válido para la tabla wsi\_record. |
| wsi.connection.bindAddress | all | Lista de IP's desde los cuales se puede iniciar la interface WSI. |
| wsi.connection.port | 8080 | Puerto por el cual se publica la interface WSI |
| wsi.connection.backlog | 32 | Número de conexiones permitidas para la interface WSI. |

* + 1. **jdbc.bcf.properties**

Conexión JDBC hacia el esquema BCF.

|  |  |  |
| --- | --- | --- |
| Nombre | Valor | Descripción |
| datasource.jdbc.driver | com.mysql.jdbc.Driver | Driver usado para la conexión a la base de datos. |
| datasource.jdbc.url | jdbc:mysql://10.40.3.131:3306/bcf | Cadena de conexión JDBC |
| datasource.jdbc.user | bcf | Usuario para conectarse con la base de datos |
| datasource.jdbc.password | \*\*\*\*\*\* | Contraseña del usuario |
| datasource.jdbc.minConnectionsPerPartition | 2 | Número mínimo de conexiones por partición. |
| datasource.jdbc.maxConnectionsPerPartition | 4 | Número máximo de conexiones por partición. |
| datasource.jdbc.partitionCount | 1 | Número de particiones. |

* + 1. **jdbc.bcf\_record.properties**

Conexión JDBC hacia el esquema BCF\_RECORD.

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Valor** | **Descripción** |
| datasource.jdbc.driver | com.mysql.jdbc.Driver | Driver usado para la conexión a la base de datos |
| datasource.jdbc.url | jdbc:mysql://10.40.3.131:3306/bcf\_record | Cadena de conexión JDBC |
| datasource.jdbc.user | bcf | Usuario para conectarse con la base de datos |
| datasource.jdbc.password | \*\*\*\*\*\* | Contraseña del usuario |
| datasource.jdbc.minConnectionsPerPartition | 1 | Número mínimo de conexiones por partición |
| datasource.jdbc.maxConnectionsPerPartition | 2 | Número máximo de conexiones por partición |
| datasource.jdbc.partitionCount | 1 | Número de particiones |

* + 1. **logrotate.properties**

Propiedades para ejecutar la rotación de las tablas de log.

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Valor** | **Descripción** |
| datasource.jdbc.driver | com.mysql.jdbc.Driver | Driver usado para la conexión a la base de datos. |
| datasource.jdbc.url | jdbc:mysql://10.40.3.131:3306/bcf\_record | Cadena de conexión JDBC |
| datasource.jdbc.user | bcf | Usuario para conectarse con la base de datos |
| datasource.jdbc.password | \*\*\*\*\*\* | Contraseña del usuario |
| datasource.jdbc.minConnectionsPerPartition | 1 | Número mínimo de conexiones por partición |
| datasource.jdbc.maxConnectionsPerPartition | 1 | Número máximo de conexiones por partición |
| datasource.jdbc.partitionCount | 1 | Número de particiones |
| logrotate.delimiter | $$ | Delimitador del script de rotación |
| logrotate.fulllinedelimiter | false | Indica si el script de rotación se va ha ejecutar como una única consulta |
| logrotate.schema | bcf\_record | Esquema donde se encuentra la tabla ha ser rotada |
| logrotate.mergedtables | wsi\_record | La tabla tipo merge, compuesta de wsi\_record\_0 y wsi\_record\_1 |
| logrotate.datepattern | yyyyMMdd | Patrón de fecha para identificar por fechas las tablas rotadas |
| logrotate.sqlscript | /bcf/sql/rotation\_script\_for\_wsi\_record.sql | Ruta del script de rotación. |
| debug | false | True para habilitar logs de debug |

* 1. **LOG4J**

Se esta usando la librería de terceros log4j para administrar la ingeniería de logs. La configuración para esta librería se hace mediante el archivo **$BCF\_HOME/properties/log4j2.xml**.

* + 1. **DEFINIR ARCHIVO DE CONFIGURACIÓN**

Para definir una ruta alternativa u otro archivo de configuración diferente (por ejemplo uno para cada nodo BCF), puede editar el script de inicio de la aplicación **BCF start\_bcf.10.sh**(o el que corresponda) y cambiar el valor de la propiedad de sistema **-Dlog4j.configuration** con la nueva ruta del archivo log4j.xml (El archivo de configuración log4j2.xml no necesariamente tiene que llevar este nombre).

* + 1. **PROPIEDADES**

Se procede a listar las propiedades usadas para las actual configuración de log4j para la aplicación BCF. Para mas información acerca de las propiedades de configuración de log4j referirse a su sitio oficial (<http://logging.apache.org/log4j/2.x/>).

|  |  |  |
| --- | --- | --- |
| **Xpath** | **Valor** | **Descripción** |
| //Configuration/Properties/Property/@name = appLogFileName | /bcf/tmp/bcf-app | Ruta y nombre del archivo de registro. |
| //Configuration/Appenders/RollingFile | <RollingFile name="rollingFile" fileName="${appLogFileName}.log" filePattern="$${date:yyyy-MM}/${appLogFileName}-%d{MM-dd-yyyy}-%i.log.gz">  <PatternLayout>  <Pattern>%5p %d{ISO8601} [%t] [%c]#[%M] - %m%n</Pattern>  </PatternLayout>  <Policies>  <!-- <SizeBasedTriggeringPolicy size="10 MB"/> -->  <TimeBasedTriggeringPolicy interval="6" modulate="true"/>  </Policies>  </RollingFile> | Define el como se va ha introducir la información de loggeo en el archivo de finido en “appLogFileName”, además define sus políticas de rotación. |
| //Configuration/Appenders/EmailAndSMSAlert | <EmailAndSMSAlert name="emailAndSmsAlert">  <PatternLayout pattern="%-5level %d{HH:mm} %logger{36} - %msg"/>  <Filters>  <ThresholdFilter level="error" onMatch="ACCEPT" onMismatch="NEUTRAL"/>  <ThresholdFilter level="fatal" onMatch="ACCEPT" onMismatch="DENY"/>  </Filters>  </EmailAndSMSAlert> | Define el como se van colectando los registros que se enviarán por notificaciones email y sms. |
| //Configuration/Root/@level | info | Indica el actual nivel de log. Los posibles niveles son: trace, debug, info, warnig, error, fatal. |
| //Configuration/Root | <Root level="trace">  <AppenderRef ref="console"/>  <AppenderRef ref="rollingFile"/>  <AppenderRef ref="emailAndSmsAlert"/>  </Root> | Define los apenders que están habilitados. |