

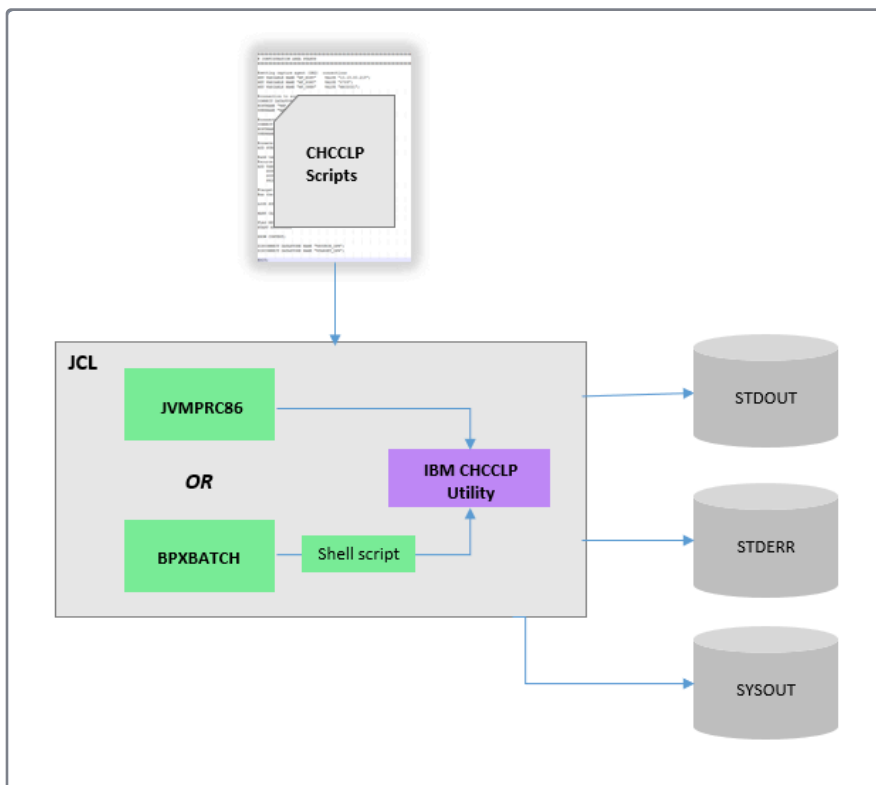
PS - CDC - Solution to address limitations of CHCCLP Utility

Problem

CHCCLP is the Command Line Processor for IBM Data Replication activities, designed to manage and control replication tasks while providing an alternative to the graphical Management Console (MC). It facilitates DevOps automation for various replication activities, such as connecting to data stores, creating, starting, and stopping subscriptions, as well as monitoring and deploying CDC objects. Additionally, it supports an embedded access server, eliminating the need to set up a standalone server to access data stores. CHCCLP operates on z/OS (OMVS/USS), Linux, and Windows platforms.

However, the CHCCLP utility lacks some of the features necessary for automation of CDC maintenance activities in production environments. It is having some limitations, including return code handling, absence of conditional or looping constructs, no delay management for asynchronous commands, and insecure handling of sensitive credentials such as passwords. Without these features, automating CDC operational workflows poses risks. This document discusses solution options to address these current limitations.

Current State



Sample Script :

```
#####;
# CONFIGURATION AREA STARTS
#####;

#setting capture agent (DB2) connections
SET VARIABLE NAME "MF_HOST" VALUE "10.19.80.219";
SET VARIABLE NAME "MF_PORT" VALUE "6789";
SET VARIABLE NAME "MF_USER" VALUE "MXCDCD1";

#connection to source datastore
CONNECT DATASTORE NAME "%SOURCE_DS%"
HOSTNAME "%MF_HOST%" PORT "%MF_PORT%"
USERNAME "%MF_USER%" PASSWORD "%MF_PASS%" CONTEXT SOURCE;

#connection to target datastore
CONNECT DATASTORE NAME "%TARGET_DS%"
HOSTNAME "%KAFKA_HOST%" PORT "%KAFKA_PORT%"
USERNAME "%KAFKA_USER%" PASSWORD "%KAFKA_PASS%" CONTEXT TARGET;

#create subscription
ADD SUBSCRIPTION NAME "%SUBS_NAME%";

#add table mapping for the subscription
#source
ADD TABLE MAPPING
    SOURCESCHEMA "%S_SCHEMA%"
    SOURCETABLE "%S_TABLE%"
    PRIMARYKEYCOLUMNS "%S_PRIMARYKEY%";

#target
#as the target is Kafka, add properties
LOCK SUBSCRIPTION;
MARK CAPTURE POINT;
FLAG REFRESH;
START MIRRORING;
SHOW CONTEXT;
DISCONNECT DATASTORE NAME "%SOURCE_DS%";
DISCONNECT DATASTORE NAME "%TARGET_DS%";
EXIT;
```

Sample job to invoke script:

```
//U432847A JOB SY63,'XXXXXX',CLASS=T,
//      MSGCLASS=X,NOTIFY=&SYSUID
//*
// JCLLIB ORDER=U432847.HLS.TESTJOBS.CDC.CHCCLP.PROC
//*
//STEP0 EXEC PGM=BPXTCAFF,PARM=TCPIP
//JZOS EXEC JVMPRC86,
// JAVACLS='com.ibm.replication.cdc.scripting.Shell',
// ARGS='-ap 10111 -f /u/u432847/chcclp/timeout.cdc',
// LOGLVL='+T'
//STEPLIB DD DISP=SHR,DSN=SYS1.SIEALNKE
//TDIN DD DISP=SHR,DSN=U432847.HLS.TESTJOBS.CDC.CHCCLP.PARM1
//STDENV DD *
```

```

PATH=/bin:/usr/bin
export PATH
CLASSPATH=/usr/lpp/chcclp/chcclp.jar
export CLASSPATH
CLASSPATH=$CLASSPATH:/usr/lpp/java/J8.0_64/bin:/usr/lpp/chcclp
export CLASSPATH
JAVA_HOME=/usr/lpp/java/J8.0_64
export JAVA_HOME
LIBPATH=/lib:/usr/lib:$JAVA_HOME/bin/classic
export LIBPATH
PATH=$PATH:$JAVA_HOME/bin
export PATH
/

```

Major limitations with this utility are:

No	Limitations	Possible workarounds	Impacts
1	<p>Inconsistent return code handling</p> <ul style="list-style-type: none"> In most cases, script processing is halted on errors when invoked by JVMRC86. However, in certain situations (e.g., during a CONNECT statement or when adding a subscription), the job may hang if errors occur. The actual return code (RC) is not propagated back to the job, which could indicate an issue with CHCCLP or BPXBATCH / JVMRC86 	<ul style="list-style-type: none"> Route the outputs to a file and process that file. Set up another job to read the output file, check the return code (RC), and take appropriate actions based on the results 	<ul style="list-style-type: none"> Additional build effort required. Overhead associated with job scheduling.
2	<p>No conditional statements</p> <ul style="list-style-type: none"> There is no facility available to enable conditional processing 	<ul style="list-style-type: none"> Split the script and use job-level conditional statements. 	<ul style="list-style-type: none"> Additional build Not easy to maintain

	within the script.		
3	No looping construct <ul style="list-style-type: none"> There is no looping construct or wait command to check the status of operational CDC commands (e.g., START MIRRORING), which are asynchronous in nature. 	<ul style="list-style-type: none"> Set up monitoring jobs or job steps to track asynchronous operations. 	<ul style="list-style-type: none"> Additional build Less maintainable Manual intervention may be required
4	Password is in clear text	<ul style="list-style-type: none"> Maintain passwords in a secured script file as SET variable commands. Merge this portion of the script with the main scripts before execution. 	<ul style="list-style-type: none"> Tactical solution (requires approval from the Security Architect and Mainframe security teams). Not a complex solution
5	No execution traces available <ul style="list-style-type: none"> There is no execution traces available to identify problems with the execution 	<ul style="list-style-type: none"> Need to enable JVM trace but difficult use it identify issues with the third party utility 	<ul style="list-style-type: none"> Operational and maintenance overhead

Solution

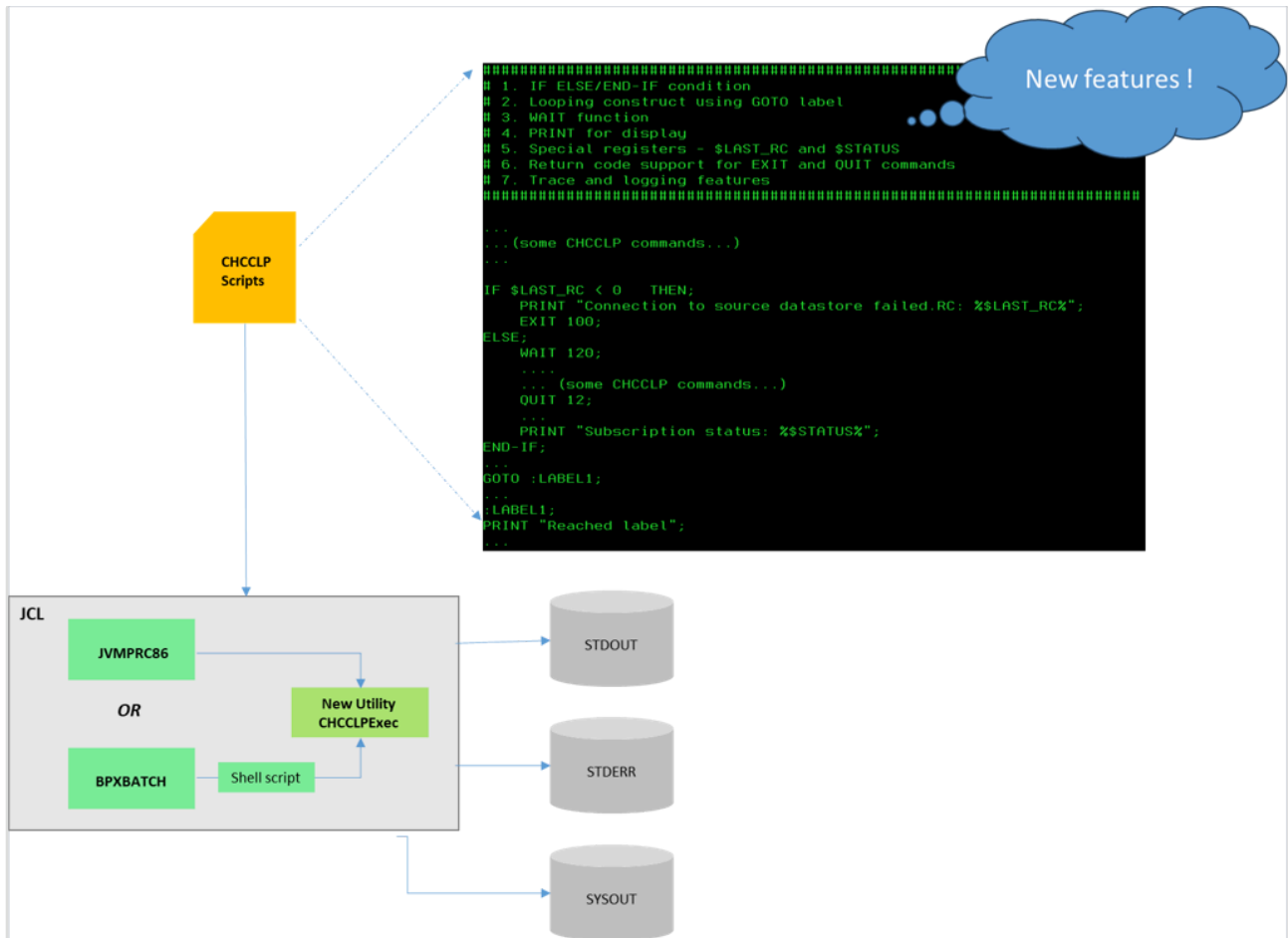
#	Options	Description	Pros	Cons
1	IBM has to enhance CHCCLP utility	<ul style="list-style-type: none"> Raise an enhancement request with IBM to modify the CHCCLP command line processor to support basic operational requirements in DevOps implementation: <ul style="list-style-type: none"> Uplift return code handling Add conditional and looping features 	<ul style="list-style-type: none"> Beneficial to multiple customers Minimize build and maintenance impact Ensure that there is no maintenance overhead, except for necessary 	<ul style="list-style-type: none"> Not expecting any immediate action from IBM team on this Need additional build from customer side to deploy this tool in prod

		<ul style="list-style-type: none"> ◦ Mask passwords in display statements ◦ Allow EXIT and QUIT commands to pass return codes back to the caller • An enhancement request for return code handling improvements has already been raised: <i>DRIMSVSAM-I-208 CHCCLP - Return Code Handling in Mainframe and LUW platform.</i> 	<p>upgrades, to streamline operations and reduce ongoing support efforts.</p>	
2	<p>Develop an in-house utility for CHCCLP command processing (Coding & unit testing completed)</p>	<ul style="list-style-type: none"> • Develop the command line processor in Java, utilizing embedded Java commands to invoke CHCCLP commands. • Ensure compatibility of the utility with both mainframe (OMVS/USS) and LUW platforms. • Leverage LLM capabilities to assist in building and testing the utility effectively. • Establish an Endeavor process to maintain and compile Java programs in the development environment. Additionally, set up the necessary script for deploying Java classes into production 	<ul style="list-style-type: none"> • Better control and allows customization • Enhanced scripting feature (conditional & looping statements, wait, print , logging etc.) • No need of waiting for IBM to provide the enhanced version of the utility • Platform independent • Java skill set is not an issue as LLMs can be used to develop and maintain the program 	<ul style="list-style-type: none"> • Additional effort for build and test • DXC support required to configure Endeavor processes for compilation and deployment • New utility will be a technical debt when IBM provides the enhanced version

3	<p>Use Embedded java for command line processing</p>	<ul style="list-style-type: none"> • Develop standalone embedded Java program when it is really required. In all other cases continue to use existing CHCCLP utility and related scripts for command line processing. No need of uplifting scripting facility or creating an equivalent in-house utility to overcome current issues with return code handling etc. • Instead, follow up with IBM to uplift their solution to support basic operational requirements. Raise P1 incident to IBM in case of any production issues with CHCCLP 	<ul style="list-style-type: none"> • No build and maintenance overhead of in-house utility in Java • No technical debt 	<ul style="list-style-type: none"> • Release dependency with IBM • Challenge to get timely support from IBM to fix prod issues •
---	---	--	--	---

Option 2 - Develop an in-house utility

Solution Context



System Impacts

BUILD CONFIG CHANGE TEST NO CHANGE

#	System	Impacted Area / Component	Change Description	Change Type
1	CD C	Finalize the features to be incorporated	<ul style="list-style-type: none"> • Pass back return code to the script at the end of each CHCCLP commands • Enhance existing EXIT and QUIT statements to pass back return code to JCL • Add provision to use this return code effectively in the script by implementing IF... ELSE and END-IF constructs • Implement basic looping construct (e.g. GOTO or WHILE) • Add WAIT logic 	BUILD

			<ul style="list-style-type: none"> • Add print (display) option 		
2	CD C	Build a new utility	<ul style="list-style-type: none"> • Develop a new utility using Java embedded scripting feature provided IBM • This utility has to accept the enhance script with additional features • It should be compatible in z/OS (USS) and LUW platforms • Test the components • Use LLM to improve the productivity of the testing 	BUILD	
3	CD C	Prepare installation and user guide details	<ul style="list-style-type: none"> • Prepare installation, user guide and operational document for the new CHCCLP utility 	BUILD	

Technical debt

Appendix - Sample job to invoke the in-house utility

```

//U432847A JOB SY63,'XXXXXX',CLASS=T,
//      MSGCLASS=X,NOTIFY=&SYSUID
//*
// JCLLIB ORDER=U432847.HLS.TESTJOBS.CDC.CHCCLP.PROC
//*
//STEP0 EXEC PGM=BPXTCAFF,PARM=TCPIP
//JZOS EXEC JVMPRC86,
// JAVACLS='com.cba.hls.utils.CHCCLPExec',
// LOGLVL='+T',
// ARGS='-ap 10111 -f /u/u432847/chcclp/ibmdemo.cdc'
//STEPLIB DD DISP=SHR,DSN=SYS1.SIEALNKE
//STDENV DD *

PATH=/bin:/usr/bin
export PATH
CLASSPATH=/usr/lpp/chcclp/chcclp.jar

```

```
export CLASSPATH
CLASSPATH=$CLASSPATH:/usr/lpp/java/J8.0_64/bin:/usr/lpp/chcclp
export CLASSPATH
CLASSPATH=$CLASSPATH:/usr/lpp/chcclp/chcclp.jar:/u/u432847/chcclp
CLASSPATH=$CLASSPATH:/usr/lpp/kc4z/kc.war/WEB-INF/lib/gson-2.2.4.jar
export CLASSPATH
JAVA_HOME=/usr/lpp/java/J8.0_64
export JAVA_HOME
LIBPATH=/lib:/usr/lib:$JAVA_HOME/bin/compat:/usr/lpp/chcclp
export LIBPATH
PATH=$PATH:$JAVA_HOME/bin
export PATH
export IBM_JAVA_OPTIONS="-Dlog.level=INFO"
/*
```