NATIVE style

SnapCenter Software

Nirupama Sriram
June 14, 2021

Table of Contents

NATIVE style

SnapCenter supports non-PERL programming or scripting languages to create plug-ins. This is known as NATIVE style programming, which can be script or BATCH file.

The NATIVE-style plug-ins must follow certain conventions given below:

The plug-in must be executable

- For Unix systems, the user who runs the agent must have execute privileges on the plug-in
- For Windows systems, PowerShell plug-ins must have the suffix .ps1, other windows scripts must have either .cmd or .bat suffix and must be executable by the user
- The plug-ins must react to command-line argument like "-quiesce", "-unquiesce"
- The plug-ins must return exit code 99 incase an operation or function is not implemented
- The plug-ins must use a specific syntax to pass results back to the server

General plug-in handling

Logging error messages

Each operation can send messages back to the server, which displays and stores the content. A message contains the message level, a timestamp, and a message text. Multiline messages are supported.

Format:

```
SC_MSG#<level>#<timestamp>#<message>
SC_MESSAGE#<level>#<timestamp>#<message>
```

Using plug-in stubs

SnapCenter plug-ins must implement plug-in stubs. These are methods that the SnapCenter Server calls based on a specific workflow.

Plug-in Stub	Optional/Required	Purpose
quiesce	required	This stub is responsible for performing a quiesce. It places the application into a state where we can create a Snapshot copy. This is called before storage Snapshot copy operation.
unquiesce	required	This stub is responsible for performing a unquiesce. It places the application in a normal state. This is called after storage Snapshot copy operation.

Plug-in Stub	Optional/Required	Purpose
clone_pre	optional	This stub is responsible for performing pre clone tasks. This assumes that you are using the built-in SnapCenter cloning interface and also is only triggered while performing action "clone_vol or clone_lun".
clone_post	Optional	This stub is responsible for performing post clone tasks. This assumes you are using the built-in SnapCenter cloning interface and also is only triggered while performing "clone_vol or clone_lun" operations.
restore_pre	Optional	This stub is responsible for performing pre restore tasks. This assumes you are using the built-in SnapCenter restore interface and is only triggered while performing restore operation.
restore	optional	This stub is responsible for performing all restore actions. This assumes you are not using built-in restore interface. It is triggered while performing restore operation.

Examples

Windows PowerShell

Check if the script can be executed on your system. If you cannot execute the script, set Set-ExecutionPolicy bypass for the script and retry the operation.

```
if ($args.length -ne 1) {
  write-warning "You must specify a method";
  break;
}
function log ($level, $message) {
  $d = get-date
  echo "SC_MSG#$level#$d#$message"
}
function quiesce {
  $app_name = (get-item env:APP_NAME).value
  log "INFO" "Quiescing application using script $app_name";
  log "INFO" "Quiescing application finished successfully"
}
function unquiesce {
  $app_name = (get-item env:APP_NAME).value
  log "INFO" "Unquiescing application using script $app_name";
  log "INFO" "Unquiescing application finished successfully"
}
  switch ($args[0]) {
    "-quiesce" {
    quiesce;
}
"-unquiesce" {
    unquiesce;
}
default {
    write-error "Function $args[0] is not implemented";
    exit 99;
  }
}
exit 0;
```