

EspAPI

User Manual





READ ME FIRST

Copyright Reserved ©

This document contains proprietary information that is protected by copyright law. All rights are reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form or by any means, that is electronic, mechanical, magnetic, optical, chemical, manual or otherwise, in whole or in part, without the prior written consent of Cronus Consulting (Pty) Ltd.

Disclaimer

Cronus Consulting (Pty) Ltd hereby disclaims any and all guarantees and warranties for the correct use and application of the ESP software.

Cronus Consulting (Pty) Ltd reserves the right to revise and make changes to the software and the content of this document from time to time without obligation to notify any person of the changes.

Ownership

The ESP Products are developed by, and is fully owned by Cronus Consulting (Pty) Ltd.



TABLE OF CONTENTS

READ ME FIRST	2
1. INTRODUCTION	4
1.1 PURPOSE	4
2. ESPAPI.....	5
2.1 USAGE AND PARAMETERS	5
2.2 LOG FILES	6
2.3 API MESSAGES AND RETURN CODES	7



1. Introduction

1.1 Purpose

The EspBatch API consists of a “Korn shell” script that provided a mechanism to submit SCL’s that have been defined within EspBatch.

The API will exit with the corresponding return code produced by the EspBatch SCL. Log files are created in **\$CRONUS/batch/<env>/tmp** that contains additional information pertaining to the submitted SCL.

The parameters supplied via the API are first validated before EspBatch attempts submission of the SCL.



2. EspAPI

2.1 Usage and Parameters

The EspBatch submission API can be called from an external scheduler or directly from the Unix/Linux command line.

The API is located in \$CRONUS/scripts and makes use of the following naming convention: **espapi.<env>.bsh** where **<env>** corresponds to the current EspBatch environment as defined by the environment variable **JBSENV**.

The API must be called using the following parameters:

SCLUSER	-	SCL User Library defined in function JS300.
SCLNAME	-	SCL Name defined in function JS300.
SCLTYPE	-	SCL Type defined in function JS300.
SCLSTEP	-	SCL Step number at which execution must start.
	-	(9999 to flush job from queue.)
EXTNAME	-	Job Name defined in external scheduler.
EXTNO	-	Job Number in external scheduler.
EXTSEQ	-	Sequence No
		<ul style="list-style-type: none"> • Initial submit: 1 • Resubmit: Greater than 1
EXTPARMS	-	Parameter(s) passed to SCL.
		Format: "VAR=VALUE VAR=VALUE" (Max 128 characters)

Usage: **\$CRONUS/scripts/espapi.<env>.bsh SCLUSER SCLNAME**
 SCLTYPE SCLSTEP EXTNAME EXTNO EXTSEQ EXTPARMS

If a SCL with the same SCL Name and Type exists in SCL User Library **OVERRIDE**, the supplied SCL User Library will be ignored and the SCL residing in library **OVERRIDE** is submitted.

The supplied parameters are validated in Natural and a **non-zero** return-code is passed back from the API if any of these are incorrect or invalid.



2.2 Log Files

A log file is created for each job that is submitted via the API. Log files are located in **\$CRONUS/batch/<env>/tmp** and have the following naming convention:

ESPAPI.<ENV>.<\$EXTNO>.<\$EXTSEQ>.<\$EXTNAME>.LOG

Contains the following information:

- Start date and time
- Parameters passed to API
- EspBatch Job No
- Status

Example: ESPAPI.DEV712.41.1.EXTSCHED.LOG

```
* Started at 2016-03-17 09:43:45
Input Parameters - ESPSUBCM:
  1 - SCL User Library: CRONUS
  2 - SCL Name.....: TESTAPI
  3 - SCL Type.....: SCL
  4 - SCL Step No.....: 1
  5 - Job Name.....: EXTJOB
  6 - Job No.....: 41
  7 - Seq No.....: 1
Dynamic Parameters:
  Parm 1 - Name.: PARM1
  Parm 1 - Value: VAL1
Job submitted via EspBatch - Batch No: 943
* Completed at 2016-03-17 09:44:01 - Batch No: 943
```

ESPAPI.<ENV>.<\$EXTNO>.<\$EXTSEQ>.<\$EXTNAME>.MSG

Contains the following information:

- Status

Example: ESPAPI.DEV712.41.1.EXTSCHED.MSG

```
* Completed at 2016-03-17 09:44:01 - Batch No: 943
```


ESPAPI.<ENV>.<\$EXTNO>.<\$EXTSEQ>.<\$EXTNAME>.RC

Contains the following information:

- Return Code

Example: ESPAPI.DEV712.41.1.EXTSCHED.RC

0

ESPAPI.<ENV>.<\$EXTNO>.<\$EXTSEQ>.<\$EXTNAME>.STEPRC

Contains the following information:

- Step name and return code

Example: ESPAPI.DEV712.41.1.EXTSCHED.STEPRC

STEP0001 0000

STEP0002 0000

STEP0003 0000

2.3 API Messages and Return Codes

Return Code	Description
0	Completed Successfully
1	Incomplete Parameter List
2	SCL not defined in EspBatch
3	Invalid Step No
4	Invalid External Job/Seq No
5	External Job No does not exist: Cannot Restart
6	External Job No already exists
7	Restart not allowed on Skipped/Deleted Step
8	Error with Job Submission via SCLSUBJ
9	Incorrect value for \$JBSOBJ
99	ESPAPI Error - Contact Support