

EspRoutines

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	5
UTILITIES OVERVIEW	5
ESP ERROR MESSAGES	6
2. ESP ROUTINES	8
ESPDSBCK – BACKUP DATASETS.....	8
ESPDSCLR – CLEAR OUT A DATASET	9
ESPDSCPY – COPY A DATASET	10
ESPDSREN – RENAME DATASETS.....	11
ESPDSDEL – DELETE DATASETS.....	12
ESPDSVRT – PRINT DATASETS	13
ESPDSVRM – PRINT MULTIPLE DATASETS.....	14
ESPPFTP – FTP INTERFACE (ONLINE INTERFACE).....	16
ESPPFTPb – FTP INTERFACE (BATCH INTERFACE)	18
ESPPDF – TRANSFORM TEXT TO PDF	21
ESPGPATH – SUBSYSTEM PATH	24
ESPMail – EMAIL INTERFACE	25
ESPMailD – EMAIL INTERFACE FOR BATCH USING DATASET.....	27
ESPSORT – SORT DATASETS USING LINUX/UNIX SORT.....	35
ESPSORTC – SORT DATASETS USING CoSORT.....	36
ESPSORTS – SORT DATASETS USING SYNC SORT	37
ESPDSMV – MOVE DATASETS	38
ESPUZIP – COMPRESS FILES (ZIP).....	40
ESPSMS – SEND SMS TEXT MESSAGES.....	44
ESPDSMOD – MODIFY A DATASET	47
ESPGZIP – GZIP A DATASET	48
ESPPAUSE – WTO CONSOLE (WRITE MESSAGE TO OPERATOR)	49
3. EXECUTION OF OPERATING SYSTEM COMMANDS	57
ESPUTIL – EXECUTE AN EXTERNAL UTILITY OR SCRIPT	57
ESPUTILN – EXECUTE AN EXTERNAL UTILITY OR SCRIPT	58
ESPUTILR – EXECUTE AN EXTERNAL UTILITY OR SCRIPT ON REMOTE MACHINE	59
4. OTHER ROUTINES	61
ESPXCOM – XCOM INTERFACE	62
ESPRFTS – RFTS INTERFACE.....	63
ESPMQCLR – CLEAR MQSERIES QUEUE	64
ESPGVCPY – COPY A DATASET TO ANOTHER FORMAT	65



ESPGVEXP – EXPORT AN ISAM FILE	66
ESPGVRES – RESTORE AN ISAM FILE	67



1. Introduction

The EspRoutines are designed to replace the Mainframe or IBM standard utilities like: IDCAMS, IEBGENER, etc. for Linux/Unix implementations.

Most of the routines can be customised and modified to suite the client's specific requirements.

Utilities overview

Provides the following functionality:

- Brief description and function of utility.
- Lists required parameter(s) - Most of these routines can be called from both RJE and batch jobs but some of them cannot be called from an online nucleus. **Please note that the parameters are case sensitive.**
- ESP Error messages * More detail on the exact error is available within the log file produced by the job step.

* ESP error messages are loaded using the standard Natural SYSERR interface.



ESP Error Messages

ESPSOFT0001	Invalid Run ID
ESPSOFT0002	Error with Print of Workfile
ESPSOFT0003	The specified Workfile does not exist
ESPSOFT0004	Error refreshing Workfile
ESPSOFT0005	No Workfiles specified
ESPSOFT0006	Error Copying Workfile
ESPSOFT0007	Error Deleting Workfile
ESPSOFT0008	SCLSUBJ - No STEPS defined for SCL
ESPSOFT0009	No Parameter defined
ESPSOFT0010	Cycle Job does not exist
ESPSOFT0011	Maximum GDG Version not Defined
ESPSOFT0012	Invalid Sub System
ESPSOFT0013	Input Parameter Not Supplied
ESPSOFT0014	Invalid Parameter Supplied
ESPSOFT0015	Conversion of Workfile Failed
ESPSOFT0016	Error with Adabas Backup
ESPSOFT0017	Error with UNIX sort
ESPSOFT0018	Cannot stop users in database
ESPSOFT0019	Problem stopping Adabas Database
ESPSOFT0020	Invalid Disposition Supplied
ESPSOFT0021	Problem Creating Database Report
ESPSOFT0022	Error with Restore of File
ESPSOFT0023	Error with Restore Command
ESPSOFT0024	Problem starting Adabas Database
ESPSOFT0025	Error creating backup dataset
ESPSOFT0026	Error changing file permissions
ESPSOFT0027	Invalid IP Address
ESPSOFT0028	FTP Server not Running
ESPSOFT0029	Invalid Home Directory
ESPSOFT0030	Error with FTP of dataset
ESPSOFT0031	Job not defined in ESPAUTO - Use Function JS405
ESPSOFT0032	Invalid Parameter
ESPSOFT0033	Invalid or Inaccessible Directory/File name Supplied
ESPSOFT0034	Error with the execution of a Utility
ESPSOFT0035	Invalid Natural Security User-Id/Password
ESPSOFT0036	Natural Security User-Id does not have access to Library
ESPSOFT0037	Report Information not defined
ESPSOFT0038	Software not available
ESPSOFT0039	INFILE for creating PDF file not supplied
ESPSOFT0040	OUTFILE for creating PDF file not supplied
ESPSOFT0041	Error creating PDF file



ESPSOFT0042 Demand of EspAuto jobs restricted
ESPSOFT0043 Job not Scheduled
ESPSOFT0044 Invalid parameter specification
ESPSOFT0045 SCL not defined in ESPBATCH
ESPSOFT0046 Error during Dynamic Parameter Expansion
ESPSOFT0047 Dynamic parameter does not exist
ESPSOFT0048 Error Generating EspAuto Job No
ESPSOFT0049 Invalid Environment supplied for job step
ESPSOFT0050 Invalid file permissions - Cannot create Workfile
ESPSOFT0051 Invalid Unix/Linux group
ESPSOFT0052 Work File disposition (N)ew - Work File Exists
ESPSOFT0053 WF disposition (O)ld - Work File does not Exist
ESPSOFT0054 Error setting ACL on Work File
ESPSOFT1234 Error with submit to Unix Scheduler
ESPSOFT7777 Job Terminated with function JS011
ESPSOFT7778 Job Terminated by EspBatch - No API process
ESPSOFT8888 Abnormal termination of batch process.



2. Esp Routines

Note that some of these routines require customisation to suit a specific operating environment and/or associated software.

ESPDSBCK – Backup Datasets

ESPDSBCK is used to “**backup**” datasets to a predefined backup destination. The backup destination must be defined as a subsystem on the ESP Codes file function **JS002**. The routine will create a backup copy of the supplied dataset(s) in the target directory. The dataset names and file attributes (permissions) of the backed-up datasets will be the same as the source datasets.

E.g: **OPSBACK** was defined in JS002 with Linux/Unix filesystem path /backups/mybackups

This utility can only be called in “**Batch mode**”

Required parameters:

- Parameter 1 - Subsystem Name to copy datasets to e.g. **OPSBACK**
This parameter cannot be the same as the WF's subsystem.
- Workfile 1-96 - Dataset Name to backup
- Subsystem 1-96 - Subsystem Name
- Disposition 1-96 - Disposition (can only be “**O**” OLD)

Wildcards are allowed as part of the dataset name e.g. MYFILE*.*

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0013** – Input Parameter Not Supplied
- ESP Error: **0012** – Invalid Sub System
- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0025** – Error creating backup dataset



ESPDSCLR – Clear out a Dataset

ESPDSCLR is used to “**clear** the **contents**” of datasets. This utility first validates that all of the supplied workfiles exist before any of them are cleared.

This utility can only be called in “**Batch mode**”

Required parameters:

- Workfiles 1-96 - Dataset name to refresh (Wildcards **NOT** allowed)
- Subsystem 1-96 - Subsystem Name
- Disposition 1-96 - Disposition (Must be “**O**” Old or “**M**” Modify)

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0003** – The specified Workfile does not exist
- ESP Error: **0004** – Error refreshing Workfile
- ESP Error: **0005** – No Workfiles specified



ESPDSCPYP – Copy a Dataset

ESPDSCPYP is used to “**copy**” datasets. One or more input workfiles may be specified but only one output file may be provided. The last file specified will always be assumed to be the output file. The optional parameter **NEWPAGE** can be used to insert a “page break” between datasets.

This utility can only be called in “**Batch mode**”

Required parameters:

- | | | |
|--------------------|---|--|
| • Workfiles 1-96 | - | Input workfile name |
| • Subsystem 1-96 | - | Subsystem Name |
| • Disposition 1-96 | - | Disposition |
| • Workfile 2-96 | - | Output workfile name (Last workfile) |
| • Subsystem 2-96 | - | Subsystem Name |
| • Disposition 2-96 | - | Disposition |

Optional parameter:

- | | | |
|------------------|---|-------------------------------------|
| • NEWPAGE | - | Inserts form feed between datasets. |
|------------------|---|-------------------------------------|

Wildcards are allowed as part of the dataset name. E.g. MYFILE*.*

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0003** – The specified Workfile does not exist
- ESP Error: **0005** – No Workfile specified
- ESP Error: **0006** – Error Copying Workfile



ESPDSREN – Rename Datasets

ESPDSREN is used to “**rename**” datasets. An equal number of source-files and target-files must be specified. Where workfile 1 will be renamed to workfile 2, workfile 3 to workfile 4, etc. The datasets are validated according to the DISPOSITION supplied on each of the files.

The **source-files** DISPOSITION must be defined as OLD (**O**) and it has to exist - the utility fails if an empty source-file is detected. This can be overruled with optional parameter **ALLOW_EMPTY** as parameter 1 – this parameter applies to **all** source-files specified in the job step. **Do not** change the source-file DISPOSITION to **NEW** (**N**) as this will **delete** the contents of the source-file prior to the rename command. The **ALLOW_EMPTY** parameter will automatically **create** an **empty file** if it does not already exist.

The **target-file** can have a DISPOSITION of NEW (**N**) or OLD (**O**). If a DISPOSITION of OLD (**O**) is specified, the utility will validate the Linux/Unix file size and if it is greater than zero (i.e. already exists and is not empty) the utility fails.

Required parameters: This utility can only be called in “**Batch mode**”

- | | | |
|--------------------|---|--------------------------------------|
| • Workfiles 1-96 | - | Input workfile name (Source) |
| • Subsystem 1-96 | - | Subsystem Name |
| • Disposition 1-96 | - | Disposition |
| • Workfile 1-96 | - | Output workfile name (Target) |
| • Subsystem 1-96 | - | Subsystem name |
| • Disposition 1-96 | - | Disposition |

Optional parameter:

- | | | |
|----------------------|---|---|
| • ALLOW_EMPTY | - | Allow empty source-files to be renamed. |
|----------------------|---|---|

Wildcards are **NOT** allowed as part of the dataset name. E.g. MYFILE*.*

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0005** – No Workfiles specified (Source/Target files do not match)
- ESP Error: **0020** – Invalid Disposition Supplied – Target-file not empty
- ESP Error: **0033** – Invalid or Inaccessible Directory/File name on source-file
- ESP Error: **0034** – Error with execution of utility - Could not move file(s) – verify permission.



ESPDSDEL – Delete Datasets

ESPDSDEL is used to “**delete**” the supplied **dataset** or range of **datasets**. This utility validates that all of the supplied workfiles exist before any of them are deleted, unless input parameter **ABORT=N** is supplied. Workfiles that contain the wildcard character “*” are not validated. The dataset(s) supplied are permanently removed from disk and cannot be recovered without restoring a file-system backup.

This utility can only be used in “**Batch mode**”

Required parameters:

- Workfiles 1-96 - **Input** workfile name to delete (Wildcards can be used)
- Subsystem 1-96 - Subsystem Name
- Disposition 1-96 - Disposition of workfile **(M) Modify or (O) Old**
- Parameter 1 - ABORT=(Y/N). Optional parameter – default is Y.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0003** – The specified Workfile does not exist
- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0007** – Error Deleting Workfile



ESPDSPRT – Print Datasets

ESPDSPRT is used to “**print**” the contents of datasets as individual reports. The content of the dataset is routed via the ESP print scripts (PRT01-31) to the Linux/Unix print spooler. All workfiles specified must at least have a corresponding printer name assigned if the optional parameter REPINFO is supplied. This utility does not support report distribution. This utility can only be used in “**Batch mode**”

Required parameters:

- | | | |
|---------------------|---|-------------------------------------|
| • Workfiles 1-30 | - | Input workfile name to print |
| • Subsystem 1-30 | - | Subsystem |
| • Disposition 1-30 | - | Disposition |
| | | |
| • Printer Name 1-30 | - | Output Pinter Name |
| • Copies 1-30 | - | Printer Copies |
| • Form Type 1-30 | - | Form Type |
| • Report Name 1-30 | - | Report Name |

Optional parameter:

- | | | |
|--------------------|---|---|
| ALLOW_EMPTY | - | Don't fail if the workfile is empty (must exist)
If this parameter is specified, the routine will not abort is the dataset is empty. |
| REPINFO | - | Fails if the corresponding report info is not supplied. |
| ALLOW_NONE | - | Don't fail if the workfile doesn't exist.
If this parameter is specified, the routine will not abort is the dataset doesn't exist. |
| PRINTER | - | Uses the ESPUX002 “user exit” to translate printer name supplied as parameter. Can be used if “long” print names are defined and needs remapping to “short” (A8) printer name. |

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0002** – Error with Print of Workfile
- ESP Error: **0003** – The specified Workfile does not exist
- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0037** – Report Information not defined
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied (Print scripts)



ESPDSPRM – Print Multiple Datasets

ESPDSPRM is used to “**print**” multiple datasets defined and to spool them as “one” dataset to the printer. The concatenated datasets are routed via the ESP print script PRT01 to the Linux/Unix print spooler. Workfile 1 must have a corresponding printer name assigned if the optional parameter **REPINFO** is specified. This utility does not support report distribution. The optional parameter **NEWPAGE** inserts breaks after consecutive datasets.

(Dataset 1 is reserved for printer definitions/setup commands)

This utility can only be used in “**Batch mode**”

Required parameters:

- | | | |
|---------------------|---|-------------------------------------|
| • Workfiles 1-30 | - | Input workfile name to print |
| • Subsystem 1-30 | - | Subsystem Name |
| • Disposition 1-30 | - | Disposition |
| | | |
| • Printer-Name 1-30 | - | Output Pinter-Name |
| • Copies 1-30 | - | Printer Copies |
| • Form Type 1-30 | - | Form Type |
| • Report-Name 1-30 | - | Report Name |

Optional parameter:

- | | | |
|--------------------|---|---|
| ALLOW_EMPTY | - | Don't fail if the workfile is empty (must exist)
If this parameter is specified, the routine will not abort is the dataset is empty. |
| REPINFO | - | Fails if the corresponding report info is not supplied. |
| NEWPAGE | - | Automatically inserts “page breaks” after dataset 2 |
| ALLOW_NONE | - | Don't fail if the workfile doesn't exist.
If this parameter is specified, the routine will not abort is the dataset doesn't exist. |
| PRINTER | - | Uses the ESPUX002 “user exit” to translate printer name supplied as parameter. Can be used if “long” print names are defined and needs remapping to “short” (A8) printer name. |

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.



Error Messages:

ESP Error: **0002** – Error with Print of Workfile

ESP Error: **0003** – The specified Workfile does not exist

ESP Error: **0005** – No Workfiles specified

ESP Error: **0037** – Report Information not defined

ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied (Print scripts)



ESPPFTP – FTP Interface (Online interface)

ESPPFTP is used to transfer datasets to and from the Linux/Unix operating system. This utility offers Natural users an interface to the standard Linux/Unix Perl FTP protocol. The **normal** FTP parameters are passed to ESPPFTP. The **Subsystem** contains the logical location of the dataset.

Required INPUT parameters:

- Parameter 1 - IP address for remote server
- Parameter 2 - FTP user-id
- Parameter 3 - FTP user-id password or **SecureFTP** to run as SFTP.
- Parameter 4 - Subsystem (Location of workfile on Linux/Unix Server)
- Parameter 5 – 90 - FTP commands e.g. **PUT,GET,DIR**,etc.

Optional INPUT parameters.

- Parameter 4 – 90 - FTP **TRIM** Command **TRIM=YES or NO**
- Parameter 4 – 90 - FTP **PASSIVE** Command **PASSIVE=YES or NO**
- Parameter 4 – 90 - FTP **SIZECHECK** Command **SIZECHECK=YES or NO**
- Parameter 4 – 90 - FTP **TIMEOUT** Command **TIMEOUT=nnn (seconds)**

Optional INPUT parameter TRIM=YES or TRIM=NO

Override the GLOBAL TRIM parameter set in JS002

Please note: The TRIM parameter is **not** supported for the **MPUT** command.

Optional INPUT parameter PASSIVE=YES or PASSIVE=NO

Start the FTP process in **PASSIVE** mode – the default is ACTIVE mode.

Use **PASSIVE=YES** to solve firewall filtering problems from the incoming data port connection to the client from the server.

Optional INPUT parameter SIZECHECK=YES or SIZECHECK=NO

This will disable the file-size check option that is not generally available on MF computers.

Optional INPUT parameter TIMEOUT=nnn (seconds)

This will set the default TIMEOUT parameter to **nnn** seconds for the FTP session.



GLOBAL FTP parameters:

TRIMFTP: This parameter can be used to remove trailing spaces from all **outbound** datasets.

Function **JS002** must be used to update this parameter:

CODETYPE=**BATCHVAR**

CODEVALUE=**TRIMFTP**

SETTING=**Y** or **N**

SKIPFTP: This parameter can be used to prevent execution of the ftp utility. If variable **SKIPFTP** has a value of **SKIP** the ftp is not executed and a report is sent to the print spooler that contains all ftp parameters.

Function **JS002** must be used to define/set this parameter:

CODETYPE=**BATCHVAR**

CODEVALUE=**SKIPFTP**

SETTING=**SKIP** or **SEND**

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

ESP Error: **0002** – Error with Print of Workfile

ESP Error: **0012** – Invalid Subsystem

ESP Error: **0027** – Invalid IP Address

ESP Error: **0028** – FTP Server not Running

ESP Error: **0029** – Invalid Home Directory

ESP Error: **0030** – Error with FTP of dataset

ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied (Print scripts)

ESP Error: **0038** – Software not available - **espftp** script in \$CRONUS/scripts.

**** Multiple PUT/GET statements are NOT allowed for the online interface****



ESPPFTPB – FTP Interface (Batch Interface)

ESPPFTPB is used to transfer datasets to and from the Linux/Unix operating system. This utility offers Natural users an interface to the standard Linux/Unix Perl FTP protocol. The **normal** FTP parameters are passed to ESPPFTPB. The **normal** FTP parameters are passed to ESPPFTP. The **Subsystem** contains the logical location of the dataset.

Required INPUT parameters:

- Parameter 1 - IP address for remote server
- Parameter 2 - FTP user-id
- Parameter 3 - FTP user-id password or **SecureFTP** to enable SFTP.**
- Parameter 4 - Subsystem (Location of workfile on Linux/Unix Server)
- Parameter 5 – 90 - FTP commands e.g. **PUT,GET,DIR**,etc.

Optional WORKFILE parameters for output logfile.

- Workfiles 1-96 - Workfile for output of FTP command
- Subsystem 1-96 - Subsystem Name
- Disposition 1-96 - Disposition
- Label 1-96 - Label must be **OUTPUT**

Optional INPUT parameters.

- Parameter 5 – 96 - FTP **TRIM** Command **TRIM=YES** or **NO**
- Parameter 5 – 96 - FTP **PASSIVE** Command **PASSIVE=YES** or **NO**
- Parameter 5 – 96 - FTP **SIZECHECK** Command **SIZECHECK=YES** or **NO**
- Parameter 5 – 96 - FTP **TIMEOUT** Command **TIMEOUT=nnn (seconds)**
- Parameter 5 – 96 - FTP **VERBOSE** Command **VERBOSE=n (level 1-9)**
- Parameter 5 – 96 - **SKIPFTP=SEND** or **SKIP**

Optional INPUT parameter TRIM=YES or TRIM=NO

Override the GLOBAL TRIM parameter set in JS002

Please note: The TRIM parameter is **not** support for **MPUT** command.

Optional INPUT parameter PASSIVE=YES or PASSIVE=NO

Start the FTP process in **PASSIVE** mode – the default mode is ACTIVE mode.

Use **PASSIVE=YES** to solve firewalls filtering problems form the incoming data port connection to the client from the server.



Optional INPUT parameter SIZECHECK=YES or SIZECHECK=NO

This will disable the file-size check option that is not generally available on MF computers.

Optional INPUT parameter TIMEOUT=nnn (seconds)

This will set the default TIMEOUT parameter to **nnn** seconds for the FTP session.

Optional INPUT parameter VERBOSE=n (LEVEL 1 to 9)

This will set the DEBUG mode to level **n** – 1 (minimum) and 9 (maximum).

GLOBAL FTP parameters:

TRIMFTP: This parameter can be used to remove trailing spaces from all **outbound** datasets.

Function **JS002** must be used to update this parameter:

CODETYPE=**BATCHVAR**

CODEVALUE=**TRIMFTP**

SETTING=**Y** or **N**

SKIPFTP: This parameter can be used to prevent execution of the ftp utility. If variable **SKIPFTP** has a value of **SKIP** the ftp is not executed and a report is sent to the print spooler that contains all ftp parameters.

Function **JS002** must be used to define/set this parameter:

CODETYPE=**BATCHVAR**

CODEVALUE=**SKIPFTP**

SETTING=**SKIP** or **SEND**

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code.

**** Executing FTP using the Secure Protocol:**

The User Password should be set to keyword "**SecureFTP**" – this automatically enables the "espftp" Perl program to use the secure FTP interface. Note that ssh key pairs must be defined for the specified User Id.



Error Messages:

ESP Error: **0002** – Error with Print of Workfile

ESP Error: **0012** – Invalid Subsystem

ESP Error: **0027** – Invalid IP Address

ESP Error: **0028** – FTP Server not Running

ESP Error: **0029** – Invalid Home Directory

ESP Error: **0030** – Error with FTP of dataset

ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied (Print scripts)

ESP Error: **0038** – Software not available - **espftp** script in \$CRONUS/scripts.

**** Multiple PUT/GET statements are allowed for the batch interface****



ESPPDF – Transform Text to PDF

ESPPDF is used to transform text datasets to PDF documents. This utility provides Natural users with an interface to the Linux/Unix operating system to a 3rd party PDF (txt2pdf) utility. Both the input workfile and target workfile names must be supplied. Each of these files must have a corresponding LABEL defined as INFILE for the input file and OUTFILE for the output or target file. The subsystem and disposition information are required for each of the workfiles. (See JS002 Code Maintenance – SUBSYSTEM for more details on subsystems).

Required Workfile parameters.

- Workfile 1 - **Input** workfile name (Text file)
- Label 1 - INFILE
- Subsystem 1 - Subsystem Name
- Disposition 1 - Disposition (must be Old)

- Workfile 2 - **Output** or target workfile name with extension (PDF file)
- Label 2 - OUTFILE
- Subsystem 2 - Subsystem Name
- Disposition 2 - Disposition (must be New)

Optional parameters to customize the PDF document format.

The following parameters can be modified to change the format the PDF document. The values below are automatically set as defaults and can be replaced with user specifics.

- npage=1 - 0 Disables page nr's in PDF doc and 1 enables page nr's
- landscape=1 - 0 for Portrait or 1 Landscape - Orientation of PDF document
- pointsize=10 - Point Size (Font) of PDF document. (Default is 10)
- vertspace=12 - Vertical spacing factor (Default is 12)
- zoom=1.00 - 1.00 means 100% and 1.5 means 150% - Display size when PDF document is opened
- paper=a4 - Paper Type (Default = A4)
- border=1 - 0 Disables page border and 1 enables page border



Protecting the PDF document

By using any or all of the following parameters the PDF document attributes can be manipulated. The document can for be protected with a user defined password before opening the PDF document. The PDF document can further be protected from editing options such as 'text-copy', 'print options' or 'read only' settings.

PASSWORD=<passwd>	Password to open PDF document. No PASSWORD is the default – exclude the keyword if you don't want to protect the document.
PASSWORD=UEXIT	Password to open PDF document is obtained from the customizable user-exit <i>ESPUX011</i> . The user-exit is called whenever the UEXIT keyword is defined.
PROTECT=YES/NO	Protect the PDF document with option YES. The reader will not be able to modify, copy or print the PDF document. The default is NO protect.

Predefined PDF format parameters:

1) LH133V8

o npage=0	-	No page nr in PDF doc
o landscape=1	-	Landscape orientation of PDF document
o pointsize=9	-	Point Size (Font) of PDF document.
o vertspace=9	-	Vertical spacing factor
o zoom=1.00	-	1.00 means 100%. Display size when opened
o paper=a4	-	Paper Type
o border=0	-	No page border

2) LH1150V8

o npage=0	-	No page nr in PDF doc
o landscape=1	-	Landscape orientation of PDF document
o pointsize=8	-	Point Size (Font) of PDF document.
o vertspace=9	-	Vertical spacing factor
o zoom=1.00	-	1.00 means 100%. Display size when opened
o paper=a4	-	Paper Type
o border=0	-	No page border



3) LH1190V8 and LH195V8

- npage=0 - No page nr in PDF doc
- landscape=1 - Landscape orientation of PDF document
- pointsize=8 - Point Size (Font) of PDF document.
- vertspace=9 - Vertical spacing factor
- zoom=1.00 - 1.00 means 100%. Display size when opened
- paper=a4 - Paper Type
- border=0 - No page border

4) PH80V8

- npage=0 - No page nr in PDF doc
- landscape=0 - Portrait orientation of PDF document
- pointsize=11 - Point Size (Font) of PDF document.
- vertspace=9 - Vertical spacing factor
- zoom=1.00 - 1.00 means 100%. Display size when opened
- paper=a4 - Paper Type
- border=0 - No page border

**** Only one translation INFILE/OUTFILE is allowed ****

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

ESP Error: **0001** – Error with Run ID

ESP Error: **0003** – Error with Workfile

ESP Error: **0013** – Input Parameter Not Supplied

ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied

ESP Error: **0034** – Error with the execution of a Utility

ESP Error: **0039** – Invalid INFILE Workfile name supplied

ESP Error: **0040** – Invalid OUTFILE Workfile name supplied



ESPGPATH – Subsystem Path

Sub-program ESPGPATH is used to return the complete filesystem path for the supplied subsystem code. The sub-program appends a trailing "/" at the end of the path:

Eg: CALLNAT 'ESPGPATH' #SUBSYSTEM #PATH

PATH is returned as: **/data/wf/mypath/**

Note: This is a **sub-program** and should be called with the correct parameters.

The **Subsystem** contains the logical location on operating system. (See **JS002** Code Maintenance – SUBSYSTEM for more details).

Required parameters:

- Parameter 1 - ESP Subsystem (A50)
- Parameter 2 - Linux/Unix filesystem path (A80)

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

Batch - ESP Error: **0012** – Invalid Subsystem

Online - ESP Error Window. **The subsystem name supplied is invalid**



ESPMAIL – Email Interface

Online interface ONLY. Some mail view programs restrict the MIME syntax and will not allow attachments.

ESPMAIL is used as an **online** interface to email messages from Linux/Unix. The ESPMAIL utility must be called as sub-program with parameters as defined below. This program executes via Linux/Unix SMTP mail program.

Required parameters:

- Parameter1 #MESSAGE - Email message
- (Max 1600 chars – split in 20 pieces) (A80/20)
- Parameter 2 #DATASET - Dataset that contains the message or data to be send.
(A128)
- Parameter 3 #SUBSYSTEM - JS002 Subsystem of Dataset on Disk (A20)
- Parameter 4 #IN-EMAIL-ADD - Email addresses of recipients (“,” delimited A500)
- Parameter 5 #CC-EMAIL-ADD - Email addresses of CC recipients (“,” delimited A500)
- Parameter 6 #FORMAT - Format of email message – (E , A , P, X , B or T) (A1)
- Parameter 7 #SUBJECT - Subject of Email Message (A50)
- Parameter 8 #RETURN-ADD - Sender Return Address (if applicable) (A50)

FORMAT of EMAIL MESSAGE:

FORMAT is supplied as **E** - the MESSAGE will be HTML formatted and DATASET (if not empty) embedded in the actual email message.

FORMAT is supplied as **A** - the MESSAGE will be HTML formatted and DATASET (if not empty) supplied will be sent as an attachment.

FORMAT is supplied as **P** the MESSAGE will be HTML formatted and DATASET converted to a PDF file and attached to the mail message.

FORMAT is supplied as **B** the MESSAGE will be HTML formatted and DATASET attached to the mail message as binary.



FORMAT is supplied as **T** - the message will be TEXT formatted and DATASET (if not empty) embedded in the actual email message as TEXT.

When FORMAT is NOT supplied - the message will be HTML formatted - ESPMAIL will produce an Error report if any part of the email call to the Linux/Unix SMTP Server is unsuccessful.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code.

Example programs: ESPTSTMA, ESPTSTME and ESPTSTMT in Library ESPSOFT

Error Messages:

ESP Error: **0012** – Invalid Sub System

ESP Error: **0003** – The specified Workfile does not exist

ESP Error: **0013** – Input Parameter Not Supplied

ESP Error: **0034** – Error with the execution of a Utility

ESP Error: **0033** – Invalid or Inaccessible Directory/File name Supplied



ESPMAILD – Email Interface for Batch using Dataset

Batch interface ONLY.

ESPMAILD is used as batch interface utility to initiate email messages from Natural on a Linux/Unix platform. ESPMAILD should be called using the definition(s) described below.

ESPMAILD formats Linux/Unix commands for the Linux/Unix SMTP mail program “sendemails” and relies on special **keywords** embedded in an input workfile. This input workfile should contain the email address(es), return/from address, subject line, location of message dataset and the actual dataset name with message lines that can either be embedded in the email message or attached to the email message; depending on the specific keyword used.

The email information is supplied to ESPMAILD via an **INPUT** workfile that should be defined as workfile 1, if you wish to use a different workfile number the workfile label must be “MAILCFG”. The INPUT workfile should contain the **keywords**.

The **keywords** “*EMAILTO*” and “*SUBJECT*” are required as minimum input for ESPMAILD to initiate a Linux/Unix Email message.

If the **keywords** *DATASET* or *ATTACH_DATASET* are specified, the **keyword** *SUBSYSTEM* is mandatory as this indicates the location of the applicable dataset. The location (i.e. *SUBSYSTEM*) of the datasets/workfiles are defined in the EspBatch product under code maintenance function (JS002). A logical variable is defined in JS002 that points to the actual Linux/Unix file system directory where the workfiles/datasets are located. EspBatch automatically translates the logical *SUBSYSTEM* name to the complete file system path for each dataset.

ESPMAILD will complete with a ZERO return code if the INPUT workfile is empty but will terminate with a non-ZERO return code if the dataset supplied in the *DATASET* or *ATTACH_DATASET* **keyword(s)** could not be located or if an error occurred during the initialization of the Linux/Unix message.

Separate Linux/Unix email messages are generated for each *EMAILTO* **keyword** (followed by the rest of the *CC*, *BCC*, *SUBJECT*, *SUBSYSTEM* and *DATASET* or *ATTACH_DATASET* **keywords**) detected in the INPUT workfile. The default email format (i.e. embedded message lines or attached dataset) depends on the *DATASET* or *ATTACH_DATASET* **keywords** specified in the INPUT workfile.



Wildcards (*) can be used for the *DATASET* or *ATTACH_DATASET* keywords if the datasets are all located in the **same** SUBSYSTEM. e.g.

See Example 2 for more detail.

SUBSYSTEM(SYSPROD)

DATASET(UID.MYFILE.2007010*.TXT) or

ATTACH_DATASET(UID.MYFILE.2007010*.PDF)

Multiple datasets can be supplied as attachments *and/or* as embedded message lines using the *DATASET* and/or *ATTACH_DATASET* parameter associated to the preceding *EMAILTO* keyword.

A **MIME** table is defined in **\$CRONUS/software** directory, this table keeps an inventory of the allowed attachment extension types. Standard attachment types are defined, and the administrator can add or remove entries from this text file.

This table is used during the automatic encoding of the attachment based on the file extension. A file that is attached and not specified in the MIME table can result in a corrupt attachment being sent.

ESPMAILD is configured to abort if an unknown file extension is detected and not defined in the [\\$CRONUS/software/mime.txt](#) file.

See [Section 1.10](#) for more detail on error messages.

Sending of emails can be blocked by changing the value of Code Type/Value: **BATCHVAR/SKIPEMAIL** to "**SKIP**" using function JS002. To enable sending of email the value must be set to "**SEND**".

Detail of the parameters and email can be found in the EspBatch Logfile for the step and also as report in the spooler.

**Required Parameters for EspBatch:**

- Workfile 1 - Input workfile name
- Subsystem 1 - Subsystem Name
- Disposition 1 - Disposition (must be 'O' Old)
- Label - Workfile label "**MAILCFG**" – Only if workfile 1 is not used.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code.

Keywords and Keyword Syntax:

The keyword must start in the 1st character of a line.

- a) The **EMAILTO** keyword (**Required parameter**)

EMAILTO(value value value)

Example:

EMAILTO(email1@domain.com email2@domain.com email3@domain.com)

- b) The **CC** keyword (**Optional parameter**)

CC(value value value)

Example:

CC(ccemail1@domain.com ccemail2@domain.com ccemail3@domain.com)

- c) The **BCC** keyword (**Optional parameter**)

BCC(value value value)

Example:

BCC(bccemail1@domain.com bccemail2@domain.com
bccemail3@domain.com)

- d) The **FROM** keyword (**Optional parameter**)

FROM(value)

Example:

FROM(espmail@domain.com)



- e) The **SUBJECT** keyword (**Required parameter**)

SUBJECT(value)

Example:

SUBJECT(Unix Email Message)

- f) The **SUBSYSTEM** keyword (**Required parameter** if **DATASET** or **ATTACH_DATASET** keywords are supplied)

SUBSYSTEM(value)

Example:

SUBSYSTEM(SYSREPORT)

The SUBSYSTEM value is required for each *DATASET* or *ATTACH_DATASET* keyword supplied.

- g) The **DATASET** keyword (**Option parameter**)

DATASET(value)

Example:

DATASET(MY.WORKFILE.TXT)

or

DATASET(MY.WORKFILE.*)

Multiple Datasets:

Repeat the **SUBSYSTEM** and **DATASET** keywords for each dataset that should be embedded in the email message. If the wildcard character "*" is used; only one SUBSYSTEM keyword is required to indicate the location of the dataset group.

- h) The **ATTACH_DATASET** keyword (**Option parameter**)

ATTACH_DATASET(value)

Example:

ATTACH_DATASET(MY.WORKFILE.PDF)

or

ATTACH_DATASET(MY.WORKFILE.*)

Multiple Attachments:

Repeat the **SUBSYSTEM** and **ATTACH_DATASET** keywords for each dataset that should be attached to the email message. If the wildcard character "*" is used; only **one** SUBSYSTEM keyword is required to indicate the location of the datasets group.



Example 1 (Multiple Addresses)

Email message to multiple EMAILTO, CC and BCC addresses.

EMAILTO(email1 @domain.com email2 @domain.com)

CC(ccemail1 @domain.com cccemail2 @domain.com cccemail3 @domain.com)

BCC(bccemail1 @domain.com bccemail2 @domain.com bccemail3 @domain.com)

SUBJECT(My Subject Line)

SUBSYSTEM(SYSDA)

→ Location of dataset

DATASET(MY.DATASET1.LINES)

→ Dataset Name to embed in mail message.

AND/OR

SUBSYSTEM(SYSDA)

→ Location of dataset

ATTACH_DATASET('MY.PDF.DATASET.200701*')

→ Dataset Name(s) to attach to mail message.

Example 2 (Multiple Datasets)

Email message to multiple datasets used as attachments and embedded message lines.

EMAILTO(email1 @domain.com)

CC(ccemail1 @domain.com)

BCC(bccemail1 @domain.com bccemail2 @domain.com)

SUBJECT(Message with multiple attachments)

SUBSYSTEM(SYSPROD)

DATASET(MY.MESSAGE.LINES)

SUBSYSTEM(SYSPDF)

ATTACH_DATASET(UID.MYFILE.PDF)

SUBSYSTEM(SYSREPORT)

ATTACH_DATASET(UID.REPORTS.ACCOUNT.ZIP)



EMAILTO(email3@domain.com)

CC(ccemail1@domain.com)

BCC(bccemail2@domain.com)

SUBJECT(You have a new email message)

SUBSYSTEM(SYSPROD)

DATASET(MESSAGE1.TXT)

EMAILTO(email4@domain.com)

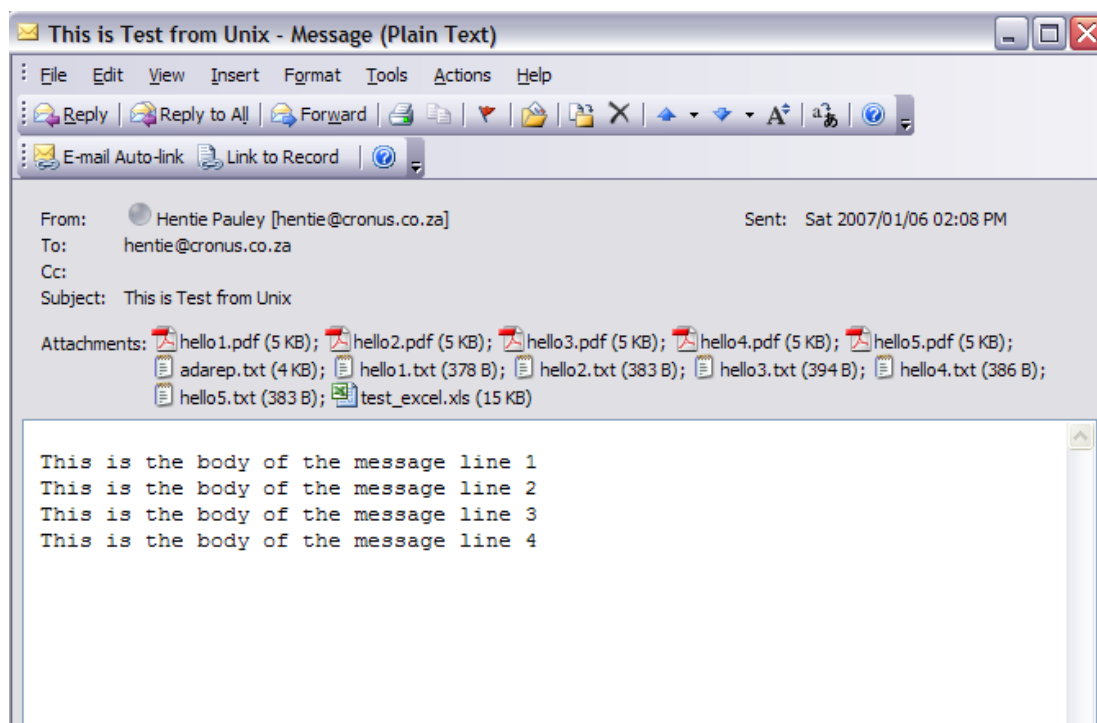
SUBJECT(You have a new email message)

SUBSYSTEM(SYSPROD)

DATASET(LAST.MESSAGE)

Example 3 (MS Outlook example)

Email message with body text and multiple attached datasets in PDF, EXCEL, and TXT format.





Special Parameters

ESPMAILD supports parameters for when it is used in combination with **ESPPDF** and **ESPPMAIL**. The following parameters can be passed if the workfile name, location and/or attachment file names are not defined within the INPUT workfile. These “special keywords” should replace the normal values defined in the INPUT file.

Example dataset with special variables:

EMAILTO(email1@domain.com)
CC(ccemail1@domain.com)
BCC(bccemail1@domain.com bccemail2@domain.com)
SUBJECT(Message with multiple attachments)
SUBSYSTEM(\$WFSUBSYS)
ATTACH_DATASET(\$WFATTACH)

Special Parameter 1:

\$WFATTACH=<attachment file name> (Max 50 characters)

Special Parameter 2:

\$WFSUBSYS=<Subsystem Name>

Special Parameter 3:

\$FROMNAME=<From email name>

Example usage of special parameters:

Actual values passed to ESPMAILD – **including** the INPUT file name that should be defined as workfile 1.

Parameters 1, 2 & 3

\$WFATTACH=UI-A4000P0-01_Closed_Acc_Report.PDF
\$WFSUBSYS=TEMP
\$FROMNAME= Linux/Unix_ESPMAILD



ESP EMAIL SETUP:

The following email parameters are defined on EspBatch function **JS002 Maintain - EspBatch Codes**. These codes are validated and used in the ESPMAILD program to formulate the message(s).

CODE TYPE : ESPBATCH

CODE VALUES : SETUP

```
E-Mail
Sender Name...: espmail_____ Domain.....: @cronus.co.za_____
Reply Address: info@cronus.co.za_____
Parameters...: sendmail -t -oi -oem -r_____
Notify On....: Y Address: support@cronus.co.za_____
Remove Logs...: N Server Type: SMTPSERVER
SMTP Server...: cronus-co-za.mail.protection.outlook.com:25_____ (IP/Domain)
```

Values for the "Server Type":

SMTPSERVER	Uses the available SMTP server to relay messages.
LOCALMAIL	Uses "Linux/Unix sendmail" as mail transport medium.

ESP Error Message

ESP Error: **0012** – Invalid Sub-System

Explanation: The SUBSYSTEM supplied is either blank or not found on the EspBatch codes file.

ESP Error: **0003** – The specified Dataset does not exist

Explanation: The dataset name supplied in the INPUT workfile for keywords DATASET or ATTACH_DATASET is either blank or not found on the Linux/Unix operating system.

ESP Error: **0033** – Invalid or Inaccessible Directory/File name Supplied

Explanation: The complete dataset path is invalid (SUBSYSTEM + dataset name) or a file used in the mail program could not be located.

ESP Error: **0034** – Error with the execution of a Utility

Explanation: The email message could not be initiated – problem with Linux/Unix mail settings.



ESPSORT – Sort Datasets Using Linux/Unix Sort

ESPSORT is used to “**sort**” datasets. This utility uses the standard Linux/Unix **sort** engine and relies on the characteristics of the Linux/Unix sort module.

Required parameters:

- | | | |
|------------------------|---|--|
| • Workfiles 1-96 | - | Input workfile name (SORTIN) |
| • Subsystem 1-96 | - | Subsystem Name |
| • Disposition 1-96 | - | Disposition |
| • Workfile 2-96 | - | Output workfile name (SORTOUT), last workfile |
| • Subsystem 96 | - | Subsystem Name |
| • Disposition 96 | - | Disposition |
| • Parameters... | - | SORT Fields e.g. 1,1,A (Parm1) |
| • Parameters... | - | SORT Fields e.g. 14,3,A (Parm3) |

Where the Mainframe syntax would be: SORT FIELDS=(**1,1,A**,14,3,A) the Linux/Unix sort routine expects the sort fields to be passed in separate parameter fields.

Note:

SUM,OMIT syntax not available using ESPSORT.

The last workfile specified will always be the output file.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

ESP Error: **0003** – The specified Workfile does not exist

ESP Error: **0005** – No Workfiles specified

ESP Error: **0017** – Error with UNIX sort



ESPSORTC – Sort Datasets Using CoSort

ESPSORTC is used to “**sort**” datasets. This utility uses the 3rd party sort engine **Cosort** and relies on the **characteristics** of the **CoSort** module.

Required Workfile parameters:

- Workfiles 1-96 - **Input** workfile name(s) SORTIN01 & SORTIN02
- Subsystem 1-96 - Subsystem Name
- Disposition 1-96 - Disposition
- Label 1-96 - SORTIN01 & SORTIN02
- Workfile 2-96 - **Output** workfile name (SORTOUT), last workfile
- Subsystem 96 - Subsystem Name
- Disposition 96 - Disposition

Required parameters (Must be defined using the syntax described in the CoSort manual)

- Parameters... - /SORT
- Parameters... - /INFILES=(\$SORTIN01,\$SORTIN02)
- Parameters... - /FIELD=(FIELD1, POS=01, SIZE=1)
- Parameters... - /FIELD=(FIELD2, POS=14, SIZE=3)
- Parameters... - /KEY=(FIELD1,ASCENDING)
- Parameters... - /KEY=(FIELD2,ASCENDING)
- Parameters... - /OUTFILE=\$SORTOUT

Where the Mainframe syntax would be: SORT FIELDS=(**1,1,A,14,3,A**) the Linux/Unix sort routine expects the sort fields to be passed in separate parameter fields.

The **LABEL** defined per workfile and not the workfile Name must be used in the parameter specifications to identify the input and output files. The label must be preceded by a '\$'.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0003** – The specified Workfile does not exist
- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0012** – Invalid Sub System
- ESP Error: **0017** – Error with UNIX sort
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied



ESPSORTS – Sort Datasets Using SyncSort

ESPSORTS is used to “**sort**” datasets. This utility uses the 3rd party sort engine **Syncsort** and relies on the **characteristics** of the **SyncSort** module.

Required Workfile parameters:

- Workfiles 1-96 - **Input** workfile name to sort (SORTIN)
- Subsystem 1-96 - Subsystem Name
- Disposition 1-96 - Disposition
- Workfile 2-96 - **Output** workfile name (SORTOUT), last workfile
- Subsystem 96 - Subsystem Name
- Disposition 96 - Disposition

Required parameters (Must be defined using the syntax described in the CoSort manual)

- Parameters... - /INFILE \$SORTIN fixed 67
- Parameters... - /FIELDS FIELD1 1 1 char
- Parameters... - /FIELDS FIELD2 14 3 char
- Parameters... - /KEYS FIELD1, FIELD2
- Parameters... - /OUTFILE \$SORTOUT fixed 67

Where the Mainframe syntax would be: SORT FIELDS=(**1,1,A,14,3,A**) the Linux/Unix sort routine expects the sort fields to be passed in separate parameter fields.

The **LABEL** defined per workfile and not the workfile Name must be used in the parameter specifications to identify the input and output files. The label must be preceded by a '\$'.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0003** – The specified Workfile does not exist
- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0012** – Invalid Sub System
- ESP Error: **0017** – Invalid Sub System
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied



ESPDSMV – Move Datasets

ESPDSMV is used to “**move**” datasets. An equal number of source-files and target-files must be specified. Whereas workfile 1 will be renamed to workfile 2, workfile 3 to workfile 4, etc. The datasets are validated according to the DISPOSITION supplied on each of the files.

The **source-files** DISPOSITION must be defined as OLD (**O**) and it has to exist - the utility fails if an empty source-file is detected. This can however be overruled with an optional parameter **ALLOW_EMPTY** as parameter – this parameter applies to **all** source-files specified in the job step. **Do not** change the source-file DISPOSITION to **NEW** (**N**) as this will **delete** the contents of the source-file prior to the rename command. The **ALLOW_EMPTY** parameter will automatically **create an empty file** if it does not already exist.

The **target-file** can have a DISPOSITION of NEW (**N**) or OLD (**O**) or MOD (**M**). If a DISPOSITION of OLD (**O**) is specified, the utility will validate the Linux/Unix file size and if it is greater than zero (i.e. already exists and is not empty) the utility fails. A DISPOSITION of OLD (**O**) should be specified if you wish to ensure that the target-file is not overwritten if it exists and contains data. DISPOSITION of MOD (**M**) will append to the target file.

Required parameters: This utility can only be called in “**Batch mode**”

- Workfiles 1-96 - **Input** workfile name (Source-file)
- Subsystem 1-96 - Subsystem Name for source-file
- Disposition 1-96 - Disposition of workfile,
- Workfile 1-96 - **Output** workfile name (Target-file)
- Subsystem 1-96 - Subsystem Name for target-file
- Disposition 1-96 - Disposition of output workfile

Optional parameter:

- **ALLOW_EMPTY** - Allow empty source-files to be renamed.
- **ERROR_PATH=<subsystem>** Location for error trapped files.

Wildcards are not allowed as part of the dataset name. e.g. MYFILE*.*

ERROR_PATH

If the move from the source to the target fails and an **ERROR_PATH** parameter was specified, the source file(s) will be moved to the **ERROR_PATH** subsystem. Should the move to the ERROR_PATH fail, the ESPDSMV utility will terminate, and an error message is written to the batch logfile.



An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

ESP Error: **0005** – Invalid number of workfile supplied

ESP Error: **0020** – Invalid disposition supplied – Target-file not empty

ESP Error: **0033** – Invalid or Inaccessible Directory/File name on source-file

ESP Error: **0034** – Error with execution of utility - Could not move file(s) – verify permission.



ESPUZIP – Compress Files (ZIP)

ESPUZIP is used to compress files on Linux/Unix. This utility provides an interface to the Linux/Unix operating system ZIP utility. Input and target file names can be supplied. Each of these files must have a corresponding LABEL. The subsystem and disposition are required for each of the files. (See JS002 Code Maintenance – SUBSYSTEM for more details on subsystems).

FORMAT=TEXT vs FORMAT=BINARY

When processing an existing ZIP archive (ARCHINFL or ARCHUPD) you should use FORMAT=BINARY and **cannot** combine TEXT files.

By specifying FORMAT=TEXT, ESPUZIP automatically enables Windows compatibility mode on all TEXT documents supplied in ZINFLnn.

Previously created ZIP that is associated with this ZIP archive will become unusable!

ECHO=YES enables the highest verbose mode and the report is automatically spooled to LPT1.

Required WORKFILE and LABEL assignments.

ARCHUPD = Zip Archive to be updated

ZIP Archive to be updated (adding new files to existing zip archive)

The **ARCHUPD** label specifies the archive name to be read in and updated.

Action: If ARCHUPD is supplied action -u (update) is automatically added.

Fails with **RC=12** if newer files are not available (does not fail if archive doesn't exist or is empty)

Embedded Linux/Unix command: **zip -u \$ARCHUPD \$INFLnn**

(Equivalent MVS variable ARCHIVE_DSN alias ARCHIVE, ARCHIVE_DSNAME)

ARCHINFL = Zip archive to be used - create new ZIP in ARCHOUT + files.

Zip archive to be used to created new ZIP together with files (not updated)

The **ARCHINFL** label specifies the ZIP archive to be read in for processing.

Use this command when the archive is **not** to be updated and the processed file is to be written to another destination using **ARCHOUT**.

Also use this label when processing GDG's. Do not use this command in conjunction with the **ARCHUPD** command (ARCHUPD updates the archive).

(Equivalent MVS variable ARCHIVE_INFILE or ARCHIVE_INDD alias ARCHINDD, ARCHIFILE, ARCHINFILE, ARCHIVE_IFILE)



Input files are: **ARCHINFL**

Output file is: **ARCHOUT**

Fails with **RC=12** if nothing is found to do. (Does not fail if archive doesn't exist or is replaced)

Embedded Linux/Unix command: **zip \$ARCHOUT \$ARCHINFL \$INFLnn**

ZINFLnn = File(s) to add to ZIP

The ZINFLnn label identifies the file(s) to be compressed.

File(s) to add to ZIP

Action should be (ADD, FRESHEN or UPDATE)

(Equivalent MVS variable INFILE alias INDD, IFILE, INFILE_DD)

ARCHOUT = Archive name to create (not updated)

The ARCHOUT command specifies the ZIP archive to be written.

Use this command when the input archive is not to be updated with new information. This command is mainly used when processing GDG's. Do not use this command in conjunction with the ARCHUPD command (ARCHUPD updates the archive)

(Equivalent MVS variable ARCHIVE_OUTFILE (alias ARCHIVE_OUTDD, ARCHIVE_OFILE, ARCHOUTDD, ARCHOFIL, ARCHOUTFILE))

Uses ARCHINFL/ZINFLnn

Archive name to be written (**not** updated)

Action should be **ADD**

Fails with **RC=12** if nothing is found to do. (Does not fail if archive doesn't exist or is replaced)

Embedded Linux/Unix command: **zip \$ARCHOUT \$ARCHINFL \$ZINFLnn**

RZINFLnn = Old dsn-name (Rename file to new zip file name)

RZOUTFLnn = New dsn-name (Rename file to new zip file name)

Rename input file (RZINFLnn) to zip file name (RZOUTFLnn) if you want the original_name saved as new_name in ZIP file)

The RZINFLnn & RZOUTFLnn label specifies the original file name and how it is to be renamed for the associated ZIP file. An equal number of **RZINFLnn** and **RZOUTFLnn** must be specified. One file at a time. Wildcards are not supported.

(Equivalent MVS variable ZIPPED_DSN)



Note that the workfile DISP=(,DELETE,DELETE) will automatically remove the temporary file created on Linux/Unix (RZOUTFLnn), if DISP=(,KEEP,KEEP) is specified the file name in RZOUTFLnn will not be removed.

Parameters for ESPUZIP:

The default parameters passed to the Linux/Unix ZIP routine are **zip -Dj** these cannot be modified (-Dj specifies not to include sub-directories and save just the name of file in the ZIP)

Optional parameters are: -l -d -g -m -T -n(0-9) -u -f and can be specified using the keywords below.

ECHO=Y/N

N add -q (quiet) to zip command

Y add -v (verbose)

Y is Default

ACTION=ADD/DELETE/FRESHEN/TEST/UPDATE/LIST/MOVE

ADD Don't pass -u or -f

-d (deletes)

-f (freshen)

-g (grow and not new)

-m (move to archive and delete) (only on Linux/Unix)

-T (test integrity of zip)

-u (replace entry in zip if newer version is available)

unzip -l <zipname> to LIST

ADD is default

FORMAT=BINARY or TEXT

TEXT add -l parm (to convert LF to CR/LF Dos)

BINARY don't pass -l

TEXT is default

METHOD=NORMAL/MAXIMUM/FAST/SLOW

STORE add -0 (zero no compression just add)

FAST add -1 (fastest compression – less compression)

NORMAL add -6 (default)

MAXIMUM add -7 (slow and high compression level)



SLOW add -9 (slowest and highest compression level)

NORMAL is default

An ESP error message will be displayed if the ESP routine completes with a non-zero completion
More detail on the exact error is available within the log file produced by the job step.

General Rules:

- At least one LABEL must be supplied for ESPUZIP: ARCHOUT or ARCHUPD or ARCHINFL.
- ARCHINFL cannot have a DISPOSITION of NEW – this will “refresh” the ZIP file supplied.
- ARCHUPD with disposition of NEW will delete the contents of ARCHUPD (only a warning message is displayed)
- ARCHOUT with disposition of NEW will delete the contents of ARCHOUT.
- ARCHOUT and ARCHUPD and ARCHINFL cannot be supplied together.
- ARCHUPD and ARCHINFL cannot be used together.
- RZINFLnn and RZOTFLnn must have an equal number of definitions supplied.
- RZINFLnn and RZOTFLnn do not support “wildcards”.
- ACTION=LIST or ACTION=TEST supports only one ZIP file at a time – ARCHOUT or ARCHUPD can be used to identify the ZIP to process.
- RC=12 is returned if you use ACTION=UPDATE or ACTION=FRESHEN and no new files are added to the ZIP.

Additional return-codes from ESPUZIP

- 11 read or seek error**
- 12 zip has nothing to do**
- 13 missing or empty zip file**
- 14 error writing to a file**
- 15 zip was unable to create a file to write to**
- 16 bad command line parameters**
- 18 zip could not open a specified file to read**

Error Messages:

- ESP Error: **0001** – Error with Run ID
- ESP Error: **0003** – Error with Workfile
- ESP Error: **0013** – Input Parameter Not Supplied
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied
- ESP Error: **0034** – Error with the execution of a Utility
- ESP Error: **0039** – Invalid INFILE Workfile name supplied
- ESP Error: **0040** – Invalid OUTFILE Workfile name supplied



ESPSMS – Send SMS Text Messages

ESPSMS and **ESPSMSB** can be used as online or batch interface to initiate SMS Text messages from Natural on a Linux/Unix platform. Sub-program **ESPSMS** must be called using the definition(s) described in [section 2](#) and utility **ESPSMSB** must be called using the definitions described in [section 3](#).

ESPSMS uses the Cronus *Email to SMS* API for single SMS messages. *Email to SMS* allows you to extend the power of SMS to your desktop. By using your email client to send SMS's you can easily communicate with one person or groups of people while using a familiar interface.

ESPSMS can be used for multiple messages but is not recommended for large volumes (use ESPSMSB for bulk messages). With ESPSMS the input-message is limited to 160 characters.

ESPSMSB uses a HTTP call to the Cronus SMS server. Using a direct HTTP call, multiple SMS messages can be sent simultaneously. This is a very popular and fast way of sending large volume messages to multiple recipients.

Function Description ESPSMS and ESPSMSB

The SMS text message, cellular nr and Cronus SMS server password are supplied to the ESPSMS sub-program via parameter definition. See [section 2.1](#).

The user (sender email address) and password must be defined on www.sms.cronus.co.za and the *Email to SMS* option enabled. This information is passed to ESPSMS via the email-subject line which authenticates the user on the Cronus SMS server.

A warning message is displayed if ESPSMS could not initiate the text email message.

ESPSMS – Message Interface

The **ESPSMS** sub-program should be called with the following parameters.

Required Parameters

- | | | |
|---------------------|---|--|
| ▪ #MESSAGE (A160) | - | Text SMS message (Max 160 characters) |
| ▪ #SUBJECT (A50) | - | Cronus SMS server password |
| ▪ #CELL-NR (A500) | - | The Cellular telephone number (international format) |
| ▪ #RETURN-ADD (A50) | - | Senders Email Address (used for replies) |
| ▪ #SENDER-ADD (A50) | - | Senders Email Address (used for authentication) |

An error message will be displayed if the ESPSMS routine completes with a non-zero completion code.



ESPSMSB – Message Interface

The **ESPSMSB** utility must be executed with the following parameters.

- The SMS text messages and cellular nr's are supplied to the ESPSMSB in workfile format. Multiple messages and/or cellular numbers can be defined.
- The user and password must be defined on www.sms.cronus.co.za. The numeric user-id as per Cronus SMS server is used for authentication.
- The ESPSMSB utility uses the Perl script **\$CRONUS/scripts/espsmsb** to establish communication to the Cronus SMS server.
- The messages defined in the input workfile must be in the following format and pipe (|) delimited:

Cell Nr | Scheduled Date & Time | Message

For example:

0825550123|2009-09-23 09:22:26|TEST NR: 1 BULK test sms from ESPSMSB using API.
 0825554567|2009-09-23 09:22:26|TEST NR: 2 BULK test sms from ESPSMSB using API.
 0829124567|2009-09-23 09:22:26|TEST NR: 3 BULK test sms from ESPSMSB using API.
 Etc.

The **ESPSMSB** utility should be executed with the following parameters.

Required Parameters

- Workfile Name - Input workfile name (predefined or inline)
- Subsystem 1 - Subsystem Name
- Disposition 1 - Disposition of input workfile (Must be 'O' Old)

- Parameter 1 - User credentials as per www.sms.cronus.co.za
SMSACC=<User-id>:<user name>:<password>
 (The user-id is the numeric id for user-name)

ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

NOTE: The ESPSMSB API requires certain PERL packages to be installed. Please refer to installation documentation for detail on required packages.



Error Messages:

ESP Error: **0001** – Error with Run ID

ESP Error: **0003** – Error with Work file

ESP Error: **0013** – Missing Account Information / Input Parameter Not Supplied

ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied

ESP Error: **0034** – Error with the execution of a Utility

ESP Error: **0039** – Invalid Work file name supplied

More detail on the exact error is available within the log file produced by the job step.

ESP Error: **0034** – Error with the execution of a Utility

Not enough credits

ESP Error: **0034** – Error with the execution of a Utility

Error sending request: 23: invalid credentials



ESPDSMOD – Modify a Dataset

ESPDSMOD is used to “**modify**” datasets. One or more input workfiles may be specified but only one output file may be provided. The last file specified will always be assumed as the output file.

The optional parameters **NEWPAGE** can be used to automatic insert a “page break” between datasets.

This utility can only be called in “**Batch mode**”

Required workfile parameters:

- | | | |
|--------------------|---|--|
| • Workfiles 1-96 | - | Input workfile name to copy |
| • Subsystem 1-96 | - | Subsystem Name |
| • Disposition 1-96 | - | Disposition |
| • Workfile 2-96 | - | Output workfile name (Destination) Last workfile |
| • Subsystem 2-96 | - | Subsystem Name |
| • Disposition 2-96 | - | Disposition |

Optional parameters:

- | | | |
|------------------|---|---|
| • SKIP | - | SKIP= nn records when copying to output file. |
| • STOPAF | - | STOPAF= nn Stop after nn records |
| • NOSPACE | - | NOSPACE Skip “empty” records (Positions 1-256 equal spaces) |

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

ESP Error: **0003** – The specified Workfile does not exist / Invalid Input File

ESP Error: **0005** – No Workfile specified

ESP Error: **0006** – Error Copying Workfile / Dataset modification failed



ESPGZIP – GZIP a Dataset

ESPGZIP is used to “**gzip**” datasets. Only one input workfile may be specified and must have a label of INFILE. Optionally the output workfile may be specified and must have a label of OUTFILE. If the output file is not specified, the .gz file extension is automatically added to the file specified as input.

This utility can only be called in “**Batch mode**”

Required parameters:

- | | | |
|-----------------|---|-----------------------------|
| • Workfile 1 | - | Input workfile name |
| • Subsystem 1 | - | Subsystem Name |
| • Disposition 1 | - | Disposition |
| • Label 1 | - | INFILE |
| • Workfile 2 | - | Output workfile name |
| • Subsystem 2 | - | Subsystem Name |
| • Disposition 2 | - | Disposition |
| • Label 2 | - | OUTFILE |

Optional parameter:

- **MODE=BEST/FAST**
BEST – Maximum compression
FAST – Fastest compression

Wildcards are allowed as part of the dataset name. E.g. MYFILE*.*

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0003** – The specified Workfile does not exist
- ESP Error: **0005** – No Workfile specified
- ESP Error: **0006** – Error Copying Workfile



ESPPAUSE – WTO console (Write message To Operator)

ESPPAUSE is used to log informational and/or action messages to a Linux/Unix console interface. This utility provides Natural online or batch programs an interface to log additional messages viewable directly from the Linux/Unix operating system. Action and informational messages can be logged depending on the supplied **keyword**.

Batch Interface Options:

Required Parameters/Keywords:

The ESPINF Keyword

The ESPINF keyword indicates that this message should be treated as an “informational” message and that an acknowledgement is **not** required. The program that initiated the EspPause message will not wait for a reply and will continue its normal processing.

The ESPPSE Keyword

The ESPPSE keyword will **pause** the job stream until acknowledged but does not expect input response from the operator.

The ESPMSG Keyword

The ESPMSG keyword forces the job stream to **pause** and it will remain in a paused state until the operator replies to the message. The operator “reply” is written to a pre-defined workfile located under the \$CRONUS/data/wf/replies directory structure as: ESPMSG.<JOBNAME>.<BATID>.

Note:

If more than one **ESPMSG** message for a batch job is used, the output file is overwritten by subsequent replies.

Optional Workfile Definitions:

A predefined or generated file containing the message(s) can be supplied to EspPause using the Workfile definition as described below.

- Workfile 1 - **Input** workfile with message
- Label 1 - INFILE
- Subsystem 1 - Subsystem Name
- Disposition 1 - Disposition of workfile (Should be “O” Old)

(Maximum length of message for a workfile is 750 characters)



ESPPAUSO – WTO console (Online interface)

Online Interface Options: (Sub-Program ESPPAUSO)

Required Parameters/Keywords:

#LEVEL (A1)
#KEYWORD (A1)
#WTO-LINE (A500)

The ESPINF Keyword

The ESPINF keyword indicates that this message should be treated as an “informational” message and that an acknowledgement is **not** required. The program that initiated the EspPause message will not wait for reply and will continue its normal processing.

The ESPPSE Keyword

The ESPPSE keyword will **pause** the job stream until acknowledged but does not expect input response from the operator.

The level on which ESPPAUSO writes messages to the “console” is controlled via the **#LEVEL** parameter. This parameter is validated in ESPPAUSO and only if it is defined in function **JS002 – Maintenance – EspBatch Codes** with CODE-TYPE: **ESPBATCH**, CODE-VALUE: **SETUP** parameter “**EspPause Flags**” has a value of “**P**”, will the message be submitted to the Linux/Unix Console interface.

Codes file definitions:

Informational messages are logged during various execution stages in EspBatch. This code can be used to set the different log levels for messages to the ESP console Linux/Unix utility.

Additional logon must be added to “**EspPause Flags**” on function JS002:

- Values **B E C R F T P** (case sensitive) **or NONE**

B = WTO at start of Job

E = WTO on Error of Job

C = WTO on completion of Job

R = WTO if job was restarted after error

F = WTO if job was “force completed”

T = WTO if job was manually terminated

P = WTO for Program Requests

NONE = No messages to Linux/Unix console.



Online Interface

Important directories and files used by EspPause:

`$CRONUS/data/<ENV>/wf/reply`

This internal directory is used by EspPause for detail on messages received and are contained in the logfile **replug**

(eg: /opt/softwareag/cronus/data/dev712/wf/reply)

Internal temporary storage for messages to and from EspPause

`$CRONUS/data/<ENV>/wf/replies` (Defined as **SUBSYSTEM REPLIES** in function JS002)

A "reply" message from the operator (when the keyword ESPMSG is used) is written to a file in this directory. The file name or location thereof cannot be changed. The file name used is: **ESPMSG.<ESPJOBNAME>.<BATCHID>** and can be accessed by the application to retrieve the operators reply.

Note:

This file will be overwritten by subsequent messages if more than one ESPMSG message is defined within the batch job.

Executable Scripts

`$CRONUS/scripts/esppause`

Location of the **esppause** script. The esppause script is called by the ESP application using the keywords and message info described above.

`$CRONUS/scripts/espreply`

Location of the **espreply** script. Messages from ESP application are listed in descending sequence. This script is used to **reply** or **acknowledge** messages generated from the ESP application or **esppause** script. Messages will remain on this screen(s) until replied or acknowledged. To reply to a message, enter the <Message Id> (01, 02, 03, etc.) and press <ENTER>. The message format will vary depending on the keyword used. When a **ESPPSE** message is generated an additional "**WAITING JOB**" message is displayed together with the actual message. If an **ESPPSE** message is listed the EspBatch job stream is **paused** until the message is replied to or acknowledged.



Note:

An Adabas timeout might occur if these message(s) are left un-replied for a time greater than the Adabas default timeout parameters specified in the database configuration file.

\$CRONUS/scripts/espconsole

Location of the **espconsole** script.

This script is used to **list** all messages generated from the ESP application or **esppause** script. Messages will remain on this screen(s) until replied to or acknowledged via the **espreply** script. "espconsole" **cannot** be used to **reply to** or **acknowledge** messages - it is a display **only** function.

The message format will vary depending on the keyword used. When an **ESPPSE** message is generated an additional "**WAITING JOB**" message is displayed together with the actual message. If an **ESPPSE** message is listed the EspBatch job stream is **paused** until the message is replied to or acknowledged.

To end the "espconsole" script – press Control C

Note:

An Adabas timeout might occur if these message(s) are left un-replied for a time greater than the Adabas default timeout parameters specified in the database configuration file.



Examples - WTO console (Online interface) /Outst/Out/

Example ESPINF message – as viewed from “espreply”

A screenshot of a terminal window titled "Cronus Consulting". The window has a menu bar with "File", "Edit", "Options", "Send", "Receive", "Window", and "Help". The main text area displays the following message:

```
Outstanding Replies on cronus-sun. Total: 1 Page 1/1
01) 175105 hentie ESPINF MYTSTACL.4578 07/06/07 17:51:05 The job MYTSTACL
      executed pgm ESPPAUSE as user hentie and a report
      should be in the spooler called MYREPORT.
Select a reply, Pn for page number or Q to Quit:
```

The status bar at the bottom shows "VT220 TCP/IP 17:56" and a series of status icons.

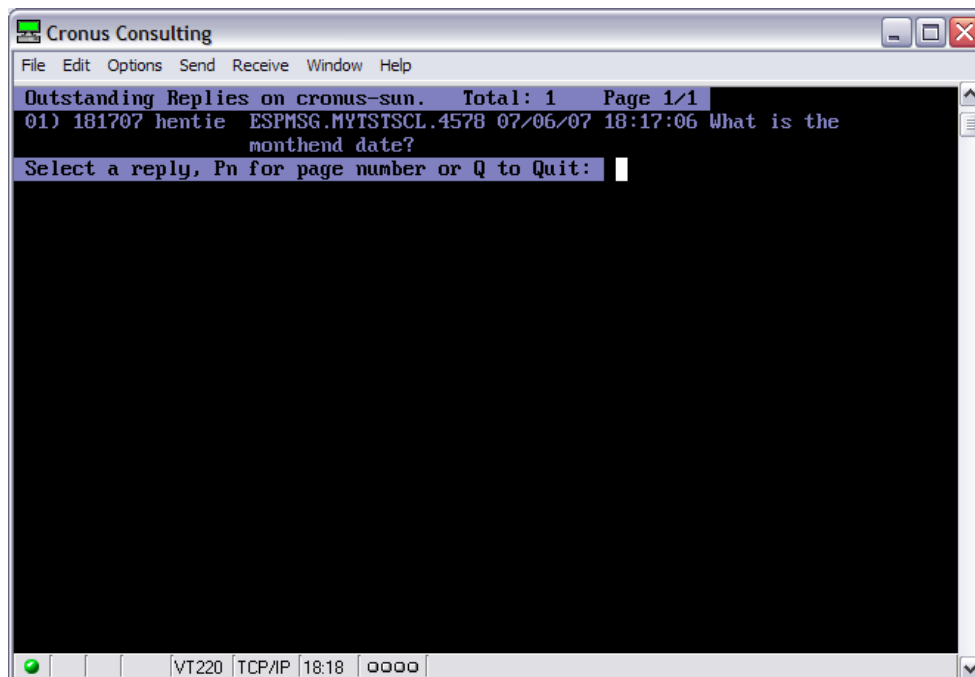
Screen 1: ESPINF Message (informational only)

Example ESPPSE message – as viewed from “espreply”

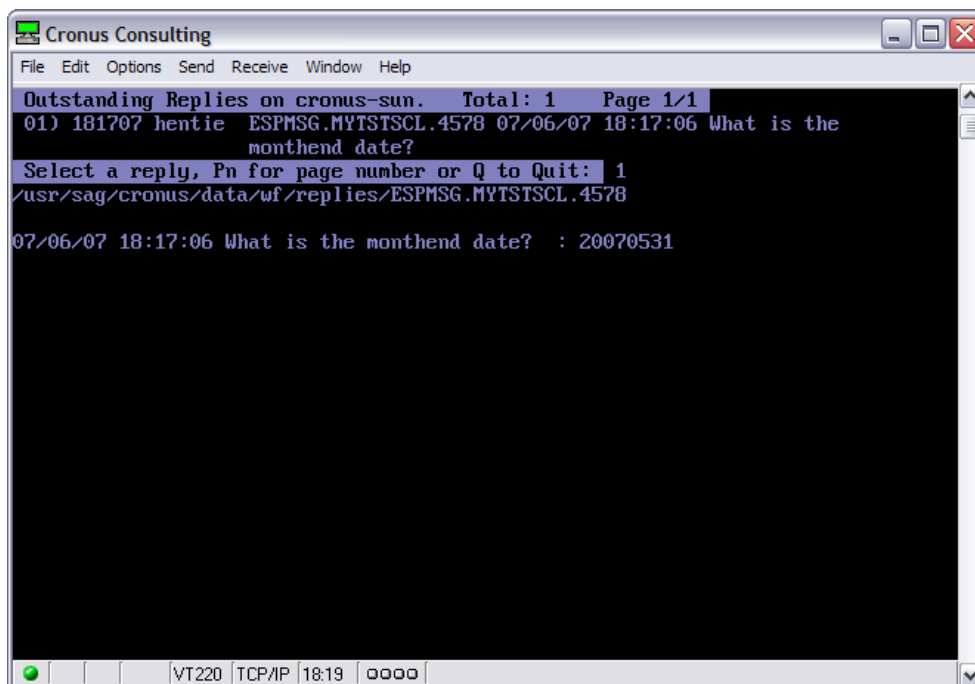
A screenshot of a terminal window titled "Cronus Consulting". The window has a menu bar with "File", "Edit", "Options", "Send", "Receive", "Window", and "Help". The main text area displays the following message:

```
Outstanding Replies on cronus-sun. Total: 1 Page 1/1
01) 181626 hentie WAITING JOB MYTSTACL.4578 07/06/07 18:16:26 Please make
      sure that the backup tape is loaded for job MYTSTACL.
      Confirm only if the device is ready!
Select a reply, Pn for page number or Q to Quit:
```

The status bar at the bottom shows "VT220 TCP/IP 18:17" and a series of status icons.

**Example ESPMSG message – as viewed from “espreply”**

The reply from the operator is written to a workfile under the \$CRONUS directory structure.





Contents of reply file generated by EspPause:

```

Cronus Consulting
File Edit Options Send Receive Window Help
-rw-rw-r-- 1 hentie sag 9 Jun 7 17:27 ESPMSG.ESPPAUSE.4575
-rw-rw-r-- 1 hentie sag 9 Jun 7 17:47 ESPMSG.ESPPAUSE.4576
-rw-rw-r-- 1 hentie sag 9 Jun 7 17:49 ESPMSG.MYTSTSC.4577
-rw-rw-r-- 1 hentie sag 9 Jun 7 18:18 ESPMSG.MYTSTSC.4578
/usr/sag/cronus/data/wf/replies > ls -lt
ksh: ls: not found
/usr/sag/cronus/data/wf/replies > ls -lt
total 26
-rw-rw-r-- 1 hentie sag 9 Jun 7 18:18 ESPMSG.MYTSTSC.4578
-rw-rw-r-- 1 hentie sag 9 Jun 7 17:49 ESPMSG.MYTSTSC.4577
-rw-rw-r-- 1 hentie sag 9 Jun 7 17:47 ESPMSG.ESPPAUSE.4576
-rw-rw-r-- 1 hentie sag 9 Jun 7 17:27 ESPMSG.ESPPAUSE.4575
-rw-rw-r-- 1 hentie sag 5 Jun 7 17:16 ESPMSG.ESPPAUSE.4574
-rw-rw-r-- 1 hentie sag 3 Jun 7 13:18 ESPMSG.ESPPAUSE.4563
-rw-rw-r-- 1 hentie sag 2 Jun 7 13:11 ESPMSG.ESPPAUSE.4562
-rw-rw-r-- 1 hentie sag 5 Jun 7 12:51 ESPMSG.ESPPAUSE.4561
-rw-rw-r-- 1 hentie sag 5 Jun 7 12:47 ESPMSG.ESPPAUSE.4560
-rw-rw-r-- 1 hentie sag 7 Jun 7 12:37 ESPMSG.ESPPAUSE.4558
-rw-rw-r-- 1 hentie sag 9 May 31 16:45 ESPMSG.ESPPAUSE.4313
-rw-rw-r-- 1 hentie sag 9 May 31 16:38 ESPMSG.ESPPAUSE.4312
-rw-rw-r-- 1 hentie sag 9 May 31 08:30 ESPMSG.ESPPAUSE.4282
/usr/sag/cronus/data/wf/replies > more ESPMSG.MYTSTSC.4578
20070531
/usr/sag/cronus/data/wf/replies >
VT220 TCP/IP 18:23 0000

```

Example ESPINF message using combination of Parameters and Workfile.

```

Cronus Consulting
File Edit Options Send Receive Window Help
Outstanding Replies on cronus-sun. Total: 1 Page 1/1
01) 181855 hentie ESPINF MYTSTSC.4578 07/06/07 18:18:54 This message is
line 2 of the parms but there is also a workfiles defi
ned for this step 1 This is the message from a workfile
in TEMP subsystem called MYMESS.FILE. 2 This is the
message from a workfile in TEMP subsystem called
MYMESS.FILE. 3 This is the message from a workfile in
TEMP subsystem called MYMESS.FILE. 4 This is the
message from a workfile in TEMP subsystem called
MYMESS.FILE. 5 This is the message from a workfile in
TEMP subsystem called MYMESS.FILE.
Select a reply, Pn for page number or Q to Quit:
VT220 TCP/IP 18:20 0000

```



Example messages via “espconsole”

The screenshot shows a window titled "Cronus Consulting" with a menu bar (File, Edit, Options, Send, Receive, Window, Help). The main area displays a table of messages:

NO	PROCID	USERID	MESSAGE
1	112238	root	ESPINF JOB ESPSTS1 RUNID 27364 Aborted On Program JSR017
2	112237	root	ESPINF JOB ESPSTS1 RUNID 27364 Started Executing Program
3	112234	hentie	ESPINF JOB ESPSTS1 RUNID 27364 Program JSR017 Resubmitted
4	112228	root	ESPINF JOB ESPSTS1 RUNID 27364 Aborted On Program JSR017
5	112226	root	ESPINF JOB ESPSTS1 RUNID 27364 Started Executing Program
6	112116	root	ESPINF JOB ESPSTS3 RUNID 27362 Completed Executing Progra
7	112111	root	ESPINF JOB ESPSTS3 RUNID 27362 Started Executing Program
8	112108	root	ESPINF JOB ESPSTS2 RUNID 27361 Completed Executing Progra
9	112104	root	ESPINF JOB ESPSTS2 RUNID 27361 Started Executing Program
10	111020	root	ESPINF JOB ESPSTS1 RUNID 27360 Completed Executing Progra
11	111015	root	ESPINF JOB ESPSTS1 RUNID 27360 Started Executing Program

The status bar at the bottom shows "VT220 TCP/IP 11:23" and a series of status icons.

To terminate the “espconsole” script – press Control C

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Logfiles:

Detailed log-files of inbound and outbound messages are kept in file **replug** and **replug_replid** located under the \$CRONUS/data/wf/reply structure.

Error Messages:

- ESP Error: **0001** – Error with Run ID
- ESP Error: **0003** – Error with Workfile
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied
- ESP Error: **0034** – Error with the execution of a Utility



3. EXECUTION OF OPERATING SYSTEM COMMANDS

The following utilities can be used to execute operating system commands, utilities, and scripts from within a batch job step.

ESPUTIL – Execute an External Utility or Script

ESPUTIL is used to execute utility commands on the local host. The utility generates a shell script that contains the parameters (PF6) defined in function is JS300. Note that all parameters are concatenated into a single command. Multiple commands must be delimited with a semi-colon “;”.

```

Parameters
SCL User: CRONUS      SCL Name.: ESPUTIL  SCL Type: SCL  V-- Library: ESPSOFT
Step No.: 1           Step Name: DB31CONT      Program: ESPUTIL

Control Card Format: @@CONTROL@@User Lib;Control Card
Setup Card Format...: @@PROC@@SubSystem;Work File Name
                    @@PROC@@$LABEL

Value
1 adarep
2 db=31
3 cont

```

The output of the command can be redirected to a workfile by defining a workfile with label **UTILOUT**.

```

Work File Parameters
SCL User: CRONUS      SCL Name.: ESPUTIL  SCL Type: SCL  V-- Library: ESPSOFT
Step No.: 1           Step Name: DB31CONT  WF No...: 01      Program: ESPUTIL

Name.....: adarep.db031.cont
Sub System.: TEMP
Disposition: N Complete: _ Abort: _ Permissions - Owner: _ Group: _ Other: _
Type.....: GDG.....: 0

```

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0032** – Invalid Parameter
- ESP Error: **0012** – Invalid Sub System
- ESP Error: **0034** – Error with the execution of a Utility
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied (Print scripts)
- ESP Error: **0002** – Error with Print of Workfile



ESPUTILN – Execute an External Utility or Script

ESPUTIL is used to execute utility commands on the local host. The utility generates a shell script that contains the parameters (PF6) defined in function is JS300. Note that parameters are not concatenated into a single command. To concatenate parameters the last character must be a hyphen '-'.

Add **ESPDEBUG=Y** as parameter to add "**set -x**" in script

```

Parameters
SCL User: CRONUS      SCL Name.: ESPUTIL  SCL Type: SCL  V-- Library: ESPSOFT
Step No.: 1          Step Name: DB31CONT      Program: ESPUTILN

Control Card Format: @@CONTROL@@User Lib;Control Card
Setup Card Format...: @@PROC@@SubSystem;Work File Name
                    @@PROC@@$LABEL

Value
1 ESPDEBUG=Y
2 adarep
3 db=31
4 cont

```

The output of the command can be redirected to a workfile by defining a workfile with label **UTILOUT**.

```

Work File Parameters
SCL User: CRONUS      SCL Name.: ESPUTIL  SCL Type: SCL  V-- Library: ESPSOFT
Step No.: 1          Step Name: DB31CONT  WF No....: 01      Program: ESPUTILN

Name.....: adarep.db031.cont
Sub System.: TEMP      Label.....: UTILOUT
Disposition: N Complete: _ Abort: _ Permissions - Owner: _ Group: _ Other: _
Type.....:             GDG.....: 0

```

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0032** – Invalid Parameter
- ESP Error: **0012** – Invalid Sub System
- ESP Error: **0034** – Error with the execution of a Utility
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied (Print scripts)
- ESP Error: **0002** – Error with Print of Workfile



ESPUTILR – Execute an External Utility or Script on Remote Machine

ESPUTIL is used to execute utility commands on a remote host. The utility generates a shell script that contains the parameters (PF6) defined in function is JS300 - this script is then transferred and executed remotely on the remote host. The first 3 parameters specified must be the IP Address/DNS Name, User ID and connection (e.g. ssh or rsh). Note that the remaining parameters are concatenated into a single command. Multiple commands must be delimited with a semi-colon “;”.

SSH keys must be defined for the specified User ID in order to connect to the remote server without a password.

```

Parameters
SCL User: CRONUS    SCL Name.: ESPUTILR SCL Type: SCL  V-- Library: ESPSOFT
Step No.: 1         Step Name: STEP0001                Program: ESPUTILR

Control Card Format: @@CONTROL@@User Lib;Control Card
Setup Card Format... @@PROC@@SubSystem;Work File Name
Value
1 Cronus-dev
2 myuser
3 ssh
4 adarep
5 dbid=31
6 cont

```

Required parameters:

- Parameter 1 - Remote Server Name or IP Address
- Parameter 2 - Remote User-id (User must have remote access via ssh or rsh)
- Parameter 3 - Remote Shell to use (ssh or rsh)
- Parameter 4-180 - Utility commands

The output of the command can be redirected to a workfile by defining a workfile with label **UTILOUT**.

```

Work File Parameters
SCL User: CRONUS    SCL Name.: ESPUTILR SCL Type: SCL  V-- Library: ESPSOFT
Step No.: 1         Step Name: STEP0001 WF No....: 01    Program: ESPUTILR

Name.....: adarep_31.txt
Sub System.: TEMP    Label.....: UTILOUT
Disposition: N Complete: _ Abort: _ Permissions - Owner: _ Group: _ Other: _
Type.....:          GDG.....: 0

```

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.



Error Messages:

ESP Error: **0002** – Error with Print of Workfile

ESP Error: **0012** – Invalid Sub System

ESP Error: **0009**– No Parameter Define or invalid

ESP Error: **0032** – Invalid Parameter

ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied (Print scripts)

ESP Error: **0034** – Error with the execution of a Utility



4. Other Routines

Please note - The following utilities have been developed specifically for certain customers and should not be used without first checking with Cronus Consulting (Pty) Ltd.



ESPXCOM – XCOM Interface

The XCOM interface for Linux/Unix.

The **normal** parameters (like to EXEC XCOMXFER.) are passed to ESPXCOM.

Mandatory parameters **IPNAME=** (must contain a valid IP add. See servers listed below.) and **SUBSYSTEM=** (location of local workfile) are required.

Some of the mainframe parameters used are no longer required but are still passed to this utility from the **JS300 SCL**. ESPXCOM will only pass the required parameters to the XCOM data transport program.

The supplied IP address will determine the server to transfer the data to.

Additional Supported Parameters:

QUEUE=

Indicates whether to execute the transmission requests immediately (specify NO) or to allow the request to be queued (specify YES). Note that if NO is specified, the request will abort if the remote system is unavailable. If YES is specified, the START_TIME and START_DATE parameters are read. See these parameters for more information.

START_TIME=

Specifies the time the transfer becomes eligible for execution. The format is hh:mm:ss. For example, 14:00:00 indicates a 2 p.m. start-time. If this field is blank, then current time is used.

START_DATE=

Specifies the date the transfer becomes eligible for execution. The format is mm/dd/yy. For example, 02/28/92 indicates a February 28, 1992 start date. If this field is blank, then current time is used.

ESPXCOM will produce an Error report is if any part of the XCOM call was unsuccessful.

The following example servers defined: (To be customised)

```
* Server1    = 53.035.003.106    = PTANTSRV3B
* Server2    = 53.254.172.016    = PTNNTSRV2
* Server3    = 53.035.134.029    = ELNNTSRV5
```



ESPRFTS – RFTS Interface

The RFTS interface for Linux/Unix.

The following parameters are supported under Linux/Unix and ESPRFTS. The syntax remains the same as with the mainframe routine.

Parameters:

- **SUBSYSTEM** = Location of DSN **** NEW**
- **DSN** = Complete workfile name (**DSN and HLQ Combined**)
- **EKZ** = Remote System Indicator EG I164
- **POOLID** = Pool Id (COMMON)
- **MFSUP** = Mainframe Support **** NEW ****
- **TRNTYPE** = Transfer Type **** NEW (B)inary or (T)ext**
- **FTSYS** = **Not required on** Linux/Unix
- **UEBTR** = **Not required on** Linux/Unix
- **DCB** = **DCB** Information. ****NEW**

ESPRFTS is setup using the above mentioned parameters.

ESPRFTS will produce an Error report is if any part of the RFTS call was unsuccessful.

- -f:i or t (trace/init) always i - unless you change it in program ESPRFTS.
- * -n: dataset name (#DSN and #HLQ) #DSN
- * -r: Remote System Identifier #EKZ
- * -st: M for Mainframe system or nothing no -st: #MFSUP
- * -b: common #POOLID
- * -d: t for text used with -st:M or b for binary #TRNTYPE
- * -u: RFT5 always RFT5 - unless you change it program ESPRFTS
- * -e: DCB Information – Organisation, RECFM,LRECL,BLOCKSIZE e.g. DCB=PS,F.300

MFSUB : If the remote system is not a Linux/Unix system this parameter should be set to **M**

TRNTYPE : Indicates if file should be sent in **Binary mode** or **text mode**. If the **MFSUB** parameter is set to **M** the system will automatically send the data as **Text**.



ESPMQCLR – Clear MQSeries Queue

Linux/Unix program used to “clear contents” of MQ Series Queue.

This program will receive parameters from the **SCL** defined in **JS300**.

- Parameter 1 - MQ Queue Name
- Parameter 2 - MQ Queue Manager Name

This program will clear out the contents of the supplied MQ Series queue.

ESPMQCLR will terminate if unsuccessful.



ESPGVCPY – Copy a Dataset to another Format

ESPGVCPY is used to copy a workfile from one format to another. This utility is only available where **MicroFocus Cobol** is implemented.

Required parameters:

- Parameter 1 - The input file format – it can be LSEQ, SEQ or IND
- Parameter 2 - The output file format – it can be LSEQ or SEQ (not IND)
- Parameter 3 - The format and record length specified as follows:
 - fnnn (f=fixed and nnn=record length)
 - vnnn-xxx (v=variable, nnn=minimum rec length, xxx=max rec length)
- Workfile 1 - **Input** workfile name
- Subsystem - Subsystem Name (location of input workfile)
- Disposition - Disposition of input workfile (Should be 'O' Old)
- Workfile 2 - **Output** workfile name
- Subsystem - Subsystem Name (location of workfile)
- Disposition - Disposition of output workfile (Should be 'N' New)

N.B - Only one input and output file may be specified.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0032** – Invalid Parameter
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied
- ESP Error: **0034** – Error with the execution of a Utility
- ESP Error: **0035** – Invalid Natural Security User-Id/Password



ESPGVEXP – Export an Isam File

ESPGVEXP is used to copy an **ISAM** file to a sequential dataset. The files created by this utility can be converted back to **ISAM** using **ESPGVRES**. This utility is only available where **MicroFocus Cobol** is implemented.

Required parameters:

- Workfile 1 - **Input** ISAM File name
- Subsystem - Subsystem Name (location of input ISAM File)
- Disposition - Disposition of input ISAM File (Should be 'O' Old)
- Workfile 2 - **Output** workfile name
- Subsystem - Subsystem Name (location of workfile)
- Disposition - Disposition of output workfile (Should be 'N' New)

N.B. - Only one input and output file may be specified.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0032** – Invalid Parameter
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied
- ESP Error: **0034** – Error with the execution of a Utility
- ESP Error: **0036** – Natural Security User-Id does not have access to Library



ESPGVRES – Restore an ISAM file

ESPGVRES is used to **restore** a **sequential** file, created using **ESPGVEXP**, to its original **ISAM** format. This utility is only available where **MicroFocus Cobol** is implemented.

Required parameters:

- Parameter 1 - The record length and format – specified as follows:
 - fnnn (f=fixed and nnn=record length)
 - vnnn-xxx (v=variable, nnn=minimum rec length, xxx=max rec length)
- Parameter 2 - Key 1 start position
- Parameter 3 - Key 1 length
- Parameter 4 - Key 2 start position (if required)
- Parameter 5 - Key 2 length (if required)
- Parameter 6 - Key 3 start position (if required) etc.
- Workfile 1 - **Input** workfile name
- Subsystem - Subsystem Name (location of input workfile)
- Disposition - Disposition of input workfile (Should be 'O' Old)
- Workfile 2 - **Output** ISAM File name
- Subsystem - Subsystem Name (location of ISAM file)
- Disposition - Disposition of output ISAM File (Should be 'N' New)

N.B. - Only one input and output file may be specified.

An ESP error message will be displayed if the ESP routine completes with a non-zero completion code. More detail on the exact error is available within the log file produced by the job step.

Error Messages:

- ESP Error: **0005** – No Workfiles specified
- ESP Error: **0032** – Invalid Parameter
- ESP Error: **0033** – Invalid or Inaccessible Directory Name Supplied
- ESP Error: **0034** – Error with the execution of a Utility
- ESP Error: **0035** – Invalid Natural Security User-Id/Password
- ESP Error: **0036** – Natural Security User-Id does not have access to Library

**** END ****