DETAILED LEVEL DESIGN FOR BATCH JOBS

FILE ARCHIVE / BACKUP

Change History:

Date	Version	Description	Updated by
06/24/2011	1.0	Initial Draft	David Shapiro
07/04/2011	1.1	Updated with review comments	Cognizant

Approval Signatures:

Name	Title	Version	Date	Signature

TABLE OF CONTENTS

1. Overview

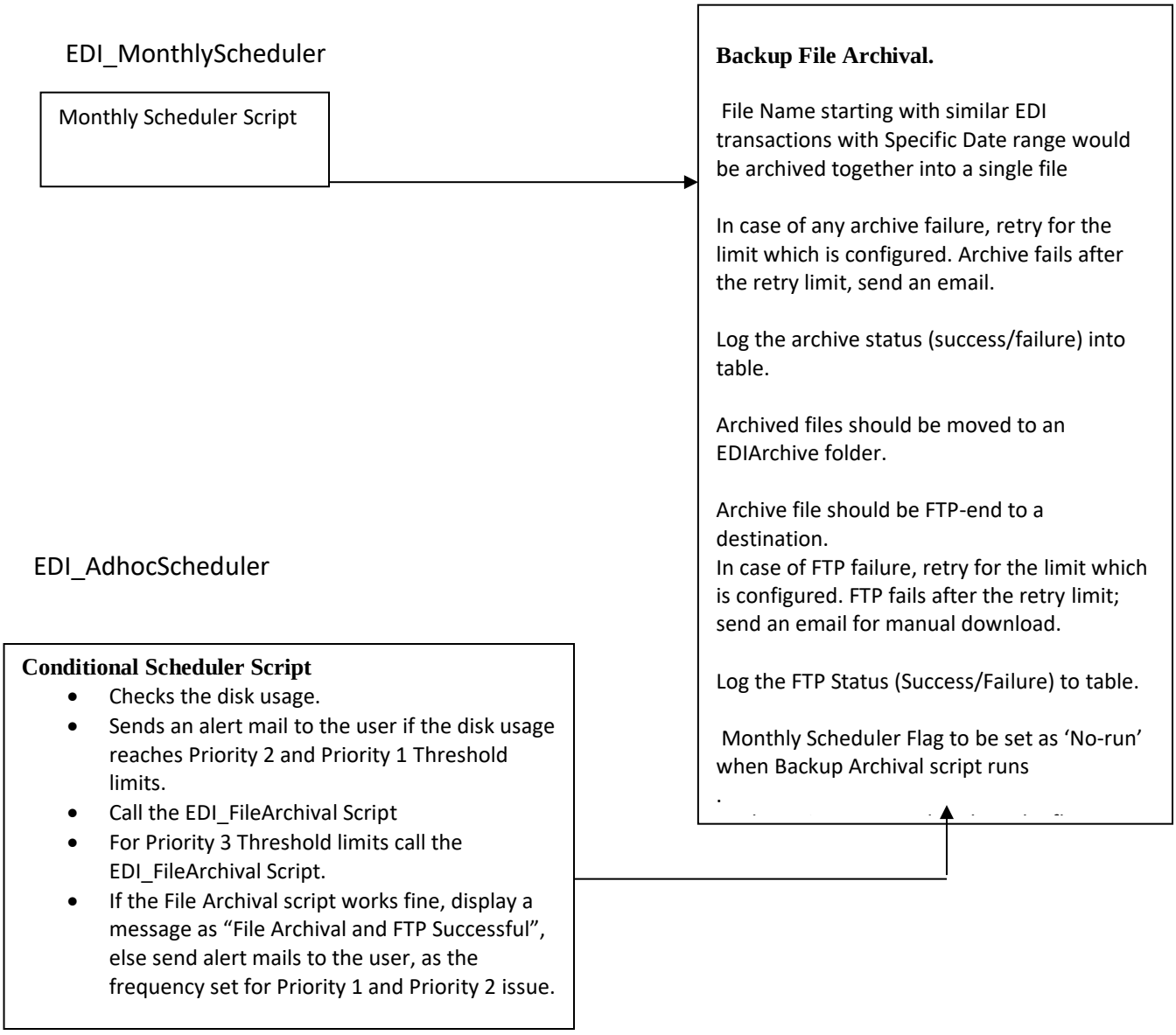
This document covers the high level design for the introduction of new File Archival process incorporated into the system. This Script will be compress the existing EDI Files in the Multiple Backup Folders and transfer it to another server. The files that had been transferred will be purged from the existing server.

1.1. References

Serial No	Document	Version
1.	EDI File Archive Backup Business_Requirement_Document_V3.0	3.0
2.		
3.		

2. Use Case Details

2.1 Backup/Archive



3. Detailed Job Design

3.1 Job Properties

3.1.1.1. EDI_FileArchival

Job Description	The job will be called for File Archival of the EDI Backup Files in multiple folders
Job Type	FTP
Job Parameterized	Parameter to be fed from Database
Job Frequency	Monthly, Daily
Job Predecessors	EDI_MonthlyScheduler & EDI_AdhocScheduler
Job Successors	N/A
Job Shared/Common	N/A
Job Process ID	Not yet assigned

3.1.1.2. EDI_MonthlyScheduler

Job Description	The job will call the EDI_FileArchival once in a month
Job Type	Monthly Scheduler
Job Parameterized	Parameter to be fed from Database
Job Frequency	Monthly
Job Predecessors	N/A
Job Successors	EDI_FileArchival
Job Shared/Common	N/A
Job Process ID	Not yet assigned

3.1.1.3. EDI_Adhoc Disk Space Scheduler

Job Description	The job will check the Disk Usage after every specified interval.
Job Type	Adhoc Scheduler
Job Parameterized	Parameter to be fed from Database
Job Frequency	Daily
Job Predecessors	N/A
Job Successors	EDI_FileArchival
Job Shared/Common	N/A
Job Process ID	Not yet assigned

3.2 Job Parameters

Parameter Name	Parameter Scope (Project/Job)	Parameter Data Type	Parameter Default Value	Parameter Description
Job Id	Job		10001	Job Id for this particular job
DB Server Name	Job		AMP5	Database server name from property file
DB Server Name	Job		AMP9	Database server name from property file
DB Schema Name for AMP5 Server	Job		CDPOLIB	Database server name from property file
DB Schema Name for AMP9 Server	Job		CSPDLIB.	Database server name from property file
DB Connection Url for AMP5	Job		Given URL	Database server name from property file
DB Connection Url for AMP5	Job		Given URL	Database server name from property file
DB User Id for AMP5	Job		ymtest	Database server name from property file
DB Pwd for AMP5	Job		ymtest	Database server name from property file
DB User Id for AMP9	Job		VNDRPORT	Database server name from property file
DB User Id for AMP9	Job		portal12	Database server name from property file
No of Retry Frequency	Job	Number	3	If FTP fails, the no of retry mechanism
Priority No	Job	Number	3	Will determine the no. of priorities

Parameter Name	Parameter Scope (Project/Job)	Parameter Data Type	Parameter Default Value	Parameter Description
Priority 3 Threshold limit	Job	Number	75%	The Priority 3 Threshold limit
Priority 2 Threshold limit	Job	Number	80%	The Priority 2 Threshold limit
Priority 1 Threshold limit	Job	Number	90%	The Priority 1 Threshold limit
Email id of the recipient	Job	Varchar		The Email id of the recipient
FTP Server name	Job	Varchar		The Remote Server name
FTP User Id	Job	Varchar		The Remote Server User Id
FTP Pwd	Job	Varchar		The Remote Server Password
NOMHPI	Job	Number	3	No of mails if FTP fails in case of Priority 1 Issue
FOMHPI	Job	Number	5min	Freq of mails if FTP fails in case of Priority 1 Issue
NOMMPI	Job	Number	2	No of mails if FTP fails in case of Priority 2 Issue
FOMMPI	Job	Number	5min	Freq of mails if FTP fails in case of Priority 2 Issue
Flag	Job	Varchar	Active/No Run	The flag is set as per the flow of FTP job

3.1.1.3 Detailed Job Design Description

- **Pre-Requisite for this Job**

1. Read the DB Server name, credentials, Schema name from the property file.
2. Read the source parameters from the database, table (ED500P).
3. There is a Root Folder would be created as /usr/local/<<EDIArchive>>
4. Backup files under the following folder and all its subfolder would be considered for Archiving and Moving.
   - **/usr/local/WagVP/trans/appl/fromWag**
   - **/usr/local/WagVP/trans/appl/toWag**

- **Archiving Procedure**

1. Files starting with similar EDI transactions(810,856 etc) will be archived together into a single archive file
2. File Structure of the Archive File is **fromWag_VP_<folder1>_…….<folder n>_<First Word of the Filename>_<DateRangFrom>_<DateRangeTo>**
3. Find the below examples:

- File Names(echoback.d052010.t070126.rdy) Starting with echoback will be archived together into a single file as fromWag_VP_bkup_ echoback_0601_0615.tar.gz
- FileName starting with VP.0180 (VP.O180.D100202.T175400.RDY) will be archived together into a single file as fromWag_VP_bkup_VP.O180_0601_0615.tar.gz
- FileNames starting with VP.0850 (VP.O850.D100202.T175400.RDY) will be archived together into a single file as fromWag_VP_bkup_VP.0850_ 0615_0630.tar.gz
- FileNames starting with VP.0861 (VP.O861.D100202.T175400.RDY) will be archived together into a single file as fromWag_VP_bkup_VP.0861_0615_0630.tar.gz
- FileNames starting with VP.861 (VP.O850.D100202.T175400.RDY) will be archived together into a single file as fromWag_VP_bkup_VP.0850_0615_0630.tar.gz
- File Names Starting with POASNInvoice (POASNInvoice_07-01-2009_0843.xml) will be archived together as a single file as fromWag_WM_bkup_POASNInvoice_0615_0630.tar.gz
- File NameStarting with I856 (I856.VMAP.D090604.T160000.V033815.ASN00489512466CL.0000000000000000) will be archived together as a single archive file as toWag_bkup_errorfiles_I856_0615_0630.tar.gz
- File Name starting with FilePolling would be archived together into a single archive file as toWag_bkup_filepolling_error_FilePolling_0615_0630.tar.gz

4. In case of any failures while archiving, there should be retry mechanism and if retry fails, an email to be triggered to re-run the process.

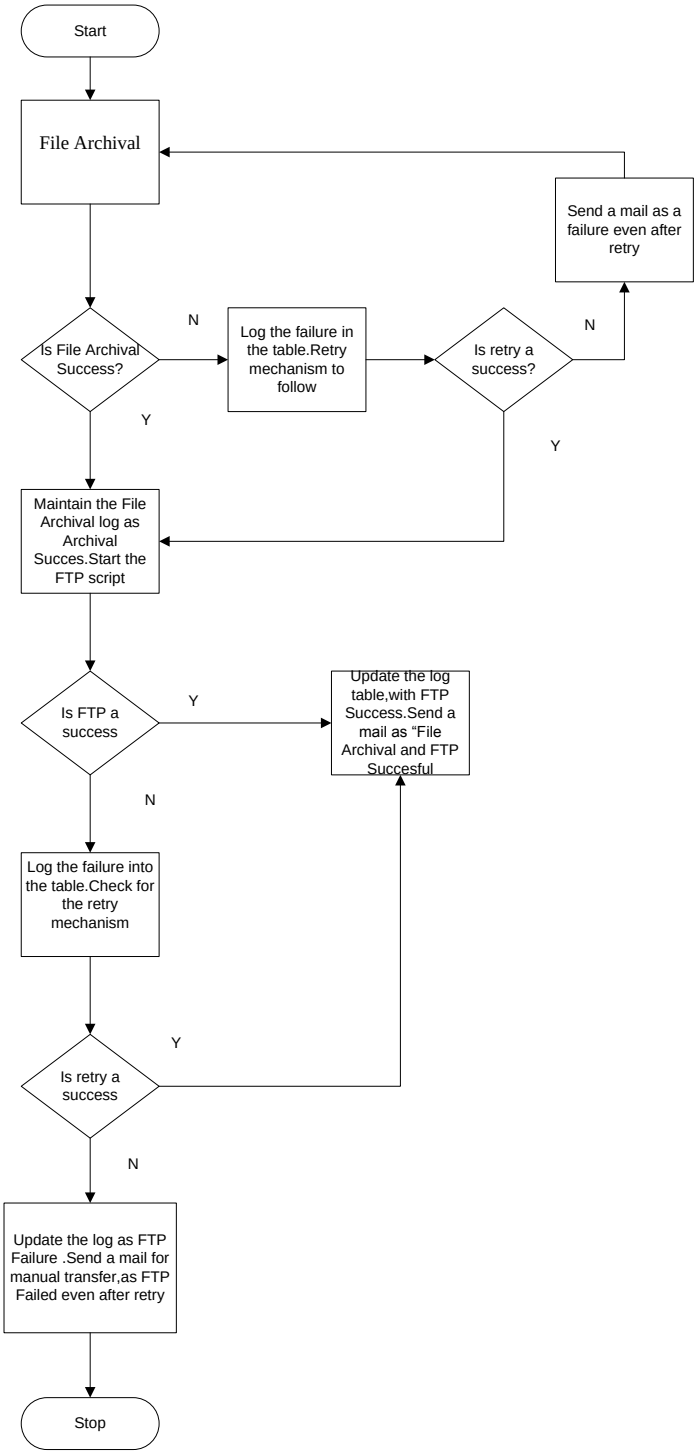
❖ **File Backup/Move**

1. FTP the files in the archive folder (for ex: **EDIArchive**) to the remote server.
2. FTP Status whether it is a failure or success should be logged in the database.
3. In case the moving fails, there are retry mechanism. No of retry frequency is store in the database.
4. In case the moving fails even after retry, an email to be sent to the user for manual transfer.

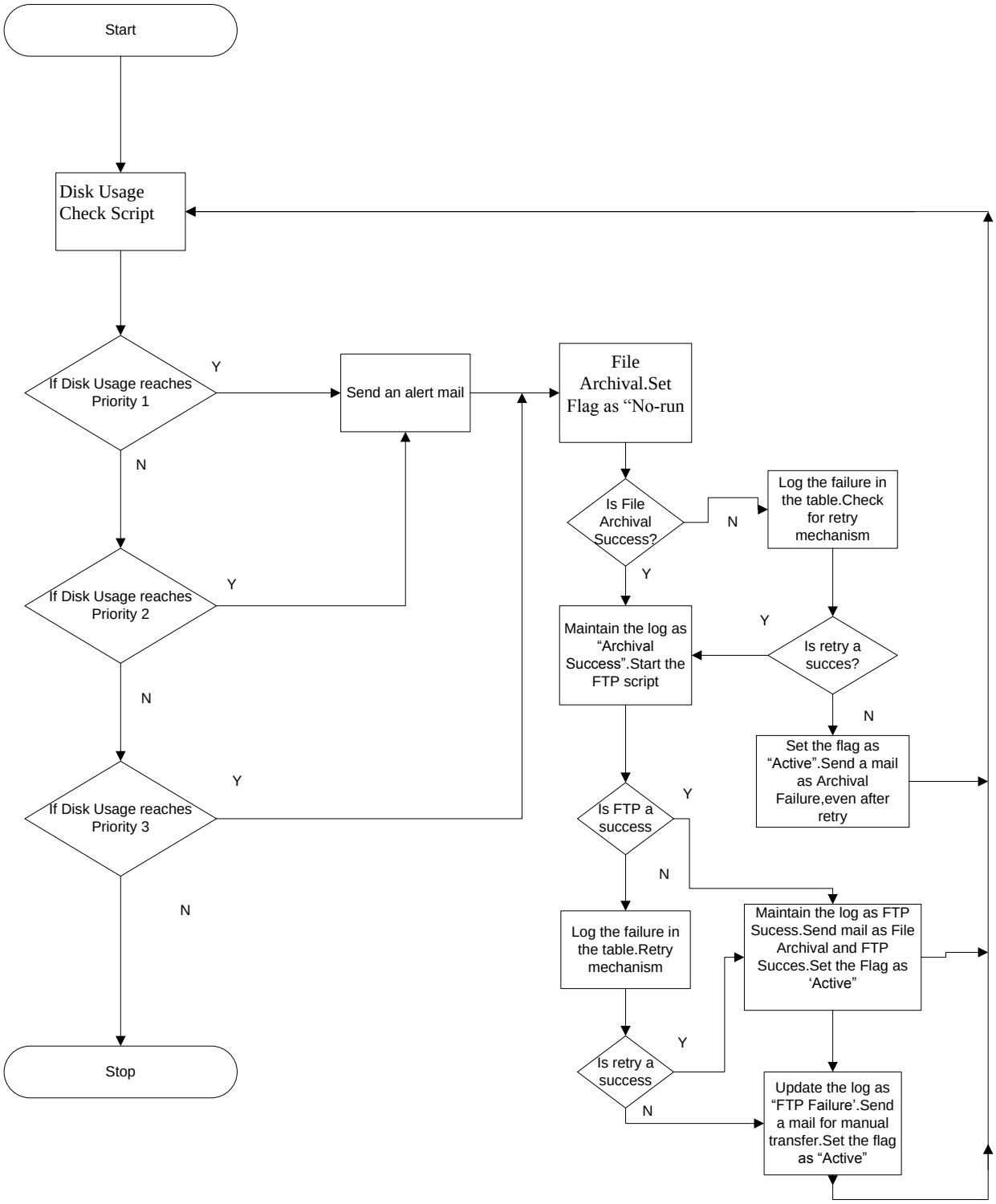
❖ **Disk Space Monitoring**

1. Another script is scheduled to check the disk usage in a defined interval, daily.
2. If the disk usage reaches the Priority 3 threshold limit call the main script, set the flag as 'no-run'. The 'no-run' flag stops the execution of the disk usage calculation script, while the file archival script is in progress. As soon as the File Archival script gets completed set the flag as 'active' and feed the control to the disk usage calculation script.
3. If the disk usage reaches the Priority 2 Threshold Limit, send an alert mail; call the main script, set the flag as 'no-run'. The 'no-run' flag stops the execution of the disk usage calculation script, while the file archival script is in progress. As soon as the File Archival script gets completed set the flag as 'active' and feed the control to the disk usage calculation script.
4. If the disk usage reaches the Priority 1 Threshold Limit, send an alert mail call the main script, set the flag as 'no-run'. The 'no-run' flag stops the execution of the disk calculation script, while the file archival script is in progress. As soon as the File Archival script gets completed set the flag as 'active' and feed the control to the disk usage calculation script.
5. Once the FTP gets successfully completed it shows a message "File archival and FTP successful".
6. If FTP fails we are sending mail depending on the number and frequency as set for the Priority 1 and Priority 2 issues. As soon as the FTP is success the mailing stops with a success mail as "FTP successful".

➢ FILE ARCHIVE FLOW



➢ DISK USAGE CHECKING FLOW



❖ **Reject/Error Handling**

Reject/Error	File Description	Handling Instructions
FTP Failure	Automatic FTP fails	Retry mechanism, if retry fails, send mail for manual transfer
Archival Failure	File Archival fails	Retry mechanism, if retry fails, send mail, stating to re run the job
DB Server	Connectivity Issue with the Database Server	Retry mechanism, if retry fails, send mail, stating to re run the job

❖ **Additional Scripts**

Script Name	Type of Script	Description
Monthly Scheduler Script	Unix Shell Script	Will call the main script monthly once
Disk Space usage Script	Unix Shell Script	Will check the disk usage and call the main script depending on the threshold limit.

4. Database Design

- **Table_Name =ED500P EDI Files Backup Source Parameters SQL Table**

Column Name	Data type	Constraint (if any)
Parameter Key	VarChar[60]	Not Null
Parameter Value	VarChar[60]	Not Null

Parameter Key can have the following parameters and also any other parameters if required.

Job Id
No of Retry Frequency
Priority 3 Threshold limit
Priority 2 Threshold limit
Priority 1 Threshold limit
Email id of the recipient
FTP Server name
FTP User Id
FTP Password
NOMHPI
FOMHPI
NOMMPI
FOMMPI
Flag
Archive Folder Name

- **Table_Name =ED501P EDI Files Archival & Transfer Log SQL Table**

Column Name	Specification	Constraint(If Any)	Primary Key
Job Id	Number	Unique Key	
Folder Name	Varchar[50]	Not null	
Archive File Name	Varchar[50]	Not null	
Files Date Range Start	Date	Not Null	
Files Date Range End	Date	Not null	
File Archival Status	Varchar[50]	Not null	File Name with path
File Archival Timestamp	Timestamp	Not null	

FTP Status	Varchar[50]		
FTP TimeStamp	Timestamp		

- The DB server names, credentials and schema details should be in property files

5. Clarifications

6. Open Issues