public Tuple<bool, string> StartProcess(string docx\_path, string output\_path, string LocalDocxPath, string IniName, string eventid, string workshopid, string proceedingid, string paperid, string acronym, string stage, string resubmit) //Resubmit module introduced 14\6\2018

{

bool \_resFlag = false;

string \_resultWebService = "Fail !!";

string \_processResult = "";

// start function on seperate thread

CancellationTokenSource cts = new CancellationTokenSource();

Task loop = Task.Factory.StartNew(() => \_processResult = StartWorkProcess(cts.Token, docx\_path, output\_path, LocalDocxPath, IniName, eventid, workshopid, proceedingid, paperid, acronym, stage, resubmit));

// wait for max seconds

if (Task.WaitAll(new Task[] { loop }, MaxProcessTime))

{

// Loop finished withion max seconds

var arr = \_processResult.Split('~');

\_resFlag = Convert.ToBoolean(arr[0]);

\_resultWebService = arr[1].ToString();

cts.Cancel();

}

else

{

// it did not finish within max seconds

cts.Cancel();

}

return new Tuple<bool, string>(\_resFlag, \_resultWebService);

}

public string StartWorkProcess(CancellationToken cancellationToken, string docx\_path, string output\_path, string LocalDocxPath, string IniName, string eventid, string workshopid, string proceedingid, string paperid, string acronym, string stage, string resubmit) // Resubmit module introduced 14\6\2018

{

string \_result = string.Empty;

string \_resFlag = string.Empty;

string \_resultWebService = string.Empty;

if (!cancellationToken.IsCancellationRequested)

{

// your loop goes here

var resultFlag = CreateDirectory\_Fetch\_File(docx\_path, output\_path, LocalDocxPath, IniName, eventid, workshopid, proceedingid, paperid, acronym, stage, resubmit);

\_resFlag = resultFlag.Item1.ToString();

\_resultWebService = resultFlag.Item2;

\_result = \_resFlag + "~" + \_resultWebService;

}

return \_result;

}