

# Files Were Not Being Sent to File Processing

We had an incident where files were not sent to file processing for about 1 hour. We wrote a PostgreSQL query to get all the required details (that are required for sending to file processing) as a JSON array for the files uploaded in that time period, saved the JSON array in a file, then queued them one at a time just using a C# console app.

## PostgreSQL Query

```
select json_agg(
  json_build_object(
    ['NodeID', public."DbNodes"."Id",
    ['AccountID', public."DbNodes"."AccountID",
    ['AccountIdentifier', public."DbAccounts"."Identifier",
    ['FileInfold', public."DbNodes"."FileInfold",
    ['FileSize', public."DbFileInfoes"."SizeInBytes",
    ['FileIdentifier', public."DbFileInfoes"."FileIdentifier",
    ['CreatedByUserID', public."DbNodes"."CreatedByUserID",
    ['FileName', public."DbNodes"."Name",
    ['EncryptionVersion', public."DbFileInfoes"."EncryptionVersion",
    ['TryCountRemaining', 3))
from public."DbNodes"
left outer join public."DbAccounts" on public."DbNodes"."AccountID" = public."DbAccounts"."Id"
left outer join public."DbFileInfoes" on public."DbNodes"."FileInfold" = public."DbFileInfoes"."Id"
where public."DbNodes"."Id" in (
  [select "NodeID" from public."DbAuditLogsToNodes" where "AuditLogID" in (
    [select "Id" from public."DbAuditLogs"
    [where public."DbAuditLogs"."ItemType" = 0
    [and public."DbAuditLogs"."ActionType" = 100
    [and public."DbAuditLogs"."Date" > '2024-06-10 16:50:00.00'
    [and public."DbAuditLogs"."Date" < '2024-06-10 17:53:00.00']])
```

## C# Code from Console App

I put breakpoints where the comments indicate, so I could manually retry when errors occurred (which did happen several times due to 429 errors w/ Cosmos)

```
using ConsoleApp1;
using Newtonsoft.Json;
using System.Text;

using HttpClient client = new HttpClient();
var nodesToProcess = JsonConvert.DeserializeObject<IndexOCRInput[]>(File.ReadAllText(@"path to file with
JSON data"));
foreach (var node in nodesToProcess)
{
    try
    {
        string json = JsonConvert.SerializeObject(node);
        var httpContent = new StringContent(json, Encoding.UTF8, "application/json");

        string url = $"https://revver-fileprocessingmanager-
us.azurewebsites.net/api/account/{node.AccountIdentifier}/node/{node.NodeID}/index?code={app key here}";
        var response = await client.PostAsync(url, httpContent);

        if (!response.IsSuccessStatusCode)
        {
            var v = ""; // put a breakpoint here
        }

        await Task.Delay(100); // this may not be needed if there is not much traffic
    }
    catch (Exception ex)
    {
        var w = ""; // put a breakpoint here
    }
}

var s = ""; // put a breakpoint here

/* Classes for Main Method to Function */

public class IndexOCRInput
{
```

```

public long NodeID { get; set; }
public int AccountID { get; set; }
public Guid AccountIdentifier { get; set; }
public AccountFeatureEnum AdvancedOcrAccountFeature { get; set; } = AccountFeatureEnum.AccusoftOCR;
public PermanentFileStorageTypeEnum AccountStorageType { get; set; } =
PermanentFileStorageTypeEnum.AWS_S3;
public EncryptionVersionEnum EncryptionVersion { get; set; }
public long FileInfolD { get; set; }
public long? FileSize { get; set; }
public string FileIdentifier { get; set; }
public long CreatedByUserID { get; set; }
public string FileName { get; set; }
public int TryCountRemaining { get; set; }
}

public enum AccountFeatureEnum
{
    FullUserLicense = 1,
    GuestUserLicense = 2,
    Governance = 3,
    FileStorageSize = 4, // Specifies the storage size limit
    FileVersioning = 5,
    Templates = 6,
    CheckInOut = 7,
    AccusoftPreview = 8,
    //SideKickLicense = 9,
    //MobileAppLicense = 10,
    ZonalOCRLicense = 11,
    AccountTemplateFeature = 12,
    FullTextSearch = 13,
    PersonalProviders = 14,
    DocumentRequests = 15,
    GovernanceLock = 16,
    AccusoftOCR = 17,
    Workflow = 19,
    Salesforce = 20,
    ItemStatus = 21,
    EnterpriseSecurity = 22,
    GovernanceUnlock = 23,
    Branding = 24,

```

```

GuestUserPreview = 25,
FilePassword = 26,
GuestUserSearch = 27,
ConcurrentLicensing = 28,
EmailFiles = 29,
BetaAccount = 30,
Records = 31,
SSO_SAML = 32,
EmailImport = 33,
PreviewerImageStamps = 34,
ESignature = 35,
DualScreenPreview = 36,
ESignatureKBA = 37,
ESignatureOTP = 38,
AnonymousAccessLinks = 39,
SearchReports = 40,
Watermark = 41,
Reporting = 42,
FormFill = 43,
PriorityOCR = 44,
O365 = 45,
WorkflowStepsPerWorkflow = 46,
AbbyFineReaderOCR = 47,
//LDAP = 48, //LDAP may have already been issued in the hub, so don't reuse this number and if we need to
re-key LDAP, just use this number
EssentialsUserLicense = 49,
PublicShareSearch = 50,
ComplianceUserLicense = 51,
}

```

```

public enum PermanentFileStorageTypeEnum
{
    AzureBlobStorage = 0,
    AWS_S3 = 1,
    Local = 2,
}

```

```

public enum EncryptionVersionEnum
{
    UtopiaV1 = 0,
}

```

```
LegacyDesktop = 1,
```

```
None = 3
```

```
}
```

---

Revision #4

Created 18 June 2024 23:54:05 by Quinn Godfrey

Updated 19 June 2024 00:15:44 by Quinn Godfrey