

Manual Background Print Call from Code

Alex Clemons - 2023-12-01 - Comments (0) - Lasetnet FO Connector FAQs

Lasetnet for Dynamics 365

When using the Performance Suite, the following code example can be used to manually call Background printing from the code:

CODE

```
/// <summary>
/// This class gives an example how to call Batch print function from the code
/// </summary>
class LACTestBatchPrintJobScheduleCallExample
{
    /// <summary>
    /// Tutorial to use Batch print function for Multithread big data processing
    /// </summary>
    [SuppressBPWarning('BPUpgradeCodeRunBaseRunCalled', 'Intentional call of the run
instead of runOperation')]
    static void runBatchPrintJobSchedule()
    {
        LACTestBatchPrintJobSchedule    batchPrintJobSchedule;
        QueryBuildDataSource             qbds;
        SRSPrintDestinationSettings      pds;
        batchPrintJobSchedule =
LACTestBatchPrintJobSchedule::construct(LACTestBatchPrintApproach::BatchBundling);
        batchPrintJobSchedule.parmReportName = 'ReportName'; // Name of the LAC report
        batchPrintJobSchedule.parmTableName = 'TableForReport'; // Name of the table to be
used in Query to pass records to generate Reports
        batchPrintJobSchedule.parmNumberOfThreads = 4; // Number of threads/tasks. Note
that number of actual parallel threads is specified on batch servers. These tasks will be
executed when available Batch Thread picks it up
        batchPrintJobSchedule.parmMaxRecords = 100; // Maximum of records (limit) to be
```

processed. Takes all from query if 0 is set

```
batchPrintJobSchedule.runInBatch = NoYes::Yes; // Parameter to run in Batch
```

```
batchPrintJobSchedule.batchGroupId = ""; // Batch group ID for the Batch jobs
```

```
batchPrintJobSchedule.bundleReports = NoYes::Yes; // Flag to make multiple reports  
per XML
```

```
batchPrintJobSchedule.bundleReportFirstSize = 2; // Number of records bundled into  
1st XML in each thread/task. It is not generated if 0 set.
```

```
batchPrintJobSchedule.bundleReportSize = 20; // Number of records bundled into  
middle (larger) XML in each thread/task. It is calculated if 0 set by: bundleReportSize =  
(parmMaxRecords div parmNumberOfThreads) - bundleReportFirstSize -  
bundleReportLastSize
```

```
batchPrintJobSchedule.bundleReportLastSize = 3; // Number of records bundled into  
last (3rd) XML in each thread/task. It is not generated if 0 set.
```

```
batchPrintJobSchedule.logPerReport = NoYes::Yes; // Flag to use logs per XML  
generation
```

```
batchPrintJobSchedule.logWithLACArchive = NoYes::No; // Flag to wait for response  
from Lasernet. DO NOT USE IT on big bundles!
```

```
batchPrintJobSchedule.logSummary = NoYes::Yes; // Flag to use logs for execution
```

```
batchPrintJobSchedule.logCPU = NoYes::No;
```

```
batchPrintJobSchedule.logMemory = NoYes::No;
```

```
// Setup query for the Scheduler which should pick up only required records. The best  
that it has single data source with ranges. No sorting or grouping.
```

```
batchPrintJobSchedule.updateQueryRunByTable(true);
```

```
qbds = batchPrintJobSchedule.queryRun().query().dataSourceNo(1);
```

```
//qbds.addRange(...); // Add required ranges on status fields or dates
```

```
// Setup print settings
```

```
pds = new SRSPrintDestinationSettings();
```

```
pds.lacDestinationSettings().skipDelivery(false);
```

```
pds.printMediumType(SRSPrintMediumType::LACFax); // Setup required destination.
```

You can use FAX for testing

```
pds.lacDestinationSettings().destTypes4Print(LACDestinationSettings::SRSPrintMediumType  
2LACDestTypes4Print(pds.printMediumType()));
```

```
pds.lacDestinationSettings().fax('TEST'); //
```

```
batchPrintJobSchedule.printSettings = pds.pack();
```

```
batchPrintJobSchedule.run();
```

```
}
```

```
static void main(Args _args)
{
    LACTestBatchPrintJobScheduleCall::runBatchPrintJobSchedule();
}

}
```