

Business Requirement

- It is required to be able to keep a comprehensive record of all Batch Jobs that run on the Salesforce system, in a standardized manner.

Functional Description

- Apex classes that implement batch processing functionality are based on a standard framework, which provides the following functionality:
 - A Job Type object, which documents each batch job type, and provides an anchor for all executions of Jobs of that Type.
 - A Job object, which documents each execution of a batch Job, including:
 - start and end time
 - status
 - a count of batches executed
 - a count of notifiable errors reported
 - a Log field
 - A Batch object, which documents each batch executed for the batch Job, including:
 - start and end time
 - status
 - a count of notifiable errors reported
 - a Log field
 - A standard way of managing JSON-formatted parameters to the batch Job
 - A standard way of handling entire rollback, for test purposes
 - A standard way of handling execution in Test mode
 - A standard way of handling reportable errors, and sending notifications of failures
 - Helper methods in the JobSupport class, to simplify the submission of batch jobs:
 - submitJob
 - scheduleJob
 - multiScheduleJob
 - Helper methods in the JobSupport class, to simplify the coding of Apex batch classes:
 - cleanupOldJobs
 - getJobForId
 - addLineToJobLog
 - addLineToBatchLog
 - Helper methods in the DebugSupport class, to simplify error handling for database DML functions:
 - sendDebugEmail
 - insertAndReturnErrors
 - updateAndReturnErrors
 - deleteAndReturnErrors

Technical Implementation

- Custom Object: Job_Type__c
- Custom Object: Job__c
- Custom Object: Batch__c
- Apex class: JobSupport
- Apex class: DebugSupport
- Apex class: BatchTestFramework (example and template, see below)

Job Type Detail

EditDeleteClone

Job Type Name	BatchTestFramework	Owner	Alan Davies [Change]
Batch Size	25	Parameters	
Disabled	☐		
Created By	Alan Davies, 7/28/2016 10:49 AM	Last Modified By	Alan Davies, 7/28/2016 10:49 AM

EditDeleteClone

Jobs							Jobs Help
Action	Job Ref	Status	Started	Finished	Parameters	Batches	Log
Edit Del	J-05508	Finished	7/28/2016 4:31 PM	7/28/2016 4:31 PM	("testParam1":"DEFAULT OVERRIDDEN")	4	Started Parameters: Parameters:[testParam1=DEFAULT OVERRIDDEN, updateDatabase=null] Finished
Edit Del	J-05507	Repeat Scheduled			("testParam1":"DEFAULT OVERRIDDEN")	0	
Edit Del	J-05506	Finished	7/28/2016 11:03 AM	7/28/2016 11:03 AM	("testParam1":"Override Default")	4	Started Parameters: Parameters:[testParam1=Override Default, updateDatabase=null] Finished
Edit Del	J-05505	Repeat Scheduled			("testParam1":"Override Default")	0	
Edit Del	J-05504	Finished	7/28/2016 11:01 AM	7/28/2016 11:01 AM		4	Started Parameters: Parameters:[testParam1=Default, updateDatabase=true] Finished

Job Detail

EditDeleteClone

Job Ref	J-05508	Owner	Alan Davies [Change]
Job Type	BatchTestFramework		
Status	Finished		
Job Id	707560000FMGdXAAX		
Log	Started Parameters: Parameters:[testParam1=DEFAULT OVERRIDDEN, updateDatabase=null] Finished		
Parameters	("testParam1":"DEFAULT OVERRIDDEN")		
Started	7/28/2016 4:31 PM		
Finished	7/28/2016 4:31 PM		
Batches	4		
Notifiable Errors	4		
Parent Job	J-05507		
Scheduled Hour			
Scheduled Minute			
Created By	Alan Davies, 7/28/2016 4:31 PM	Last Modified By	Alan Davies, 7/28/2016

EditDeleteClone

Batches						
Action	Batch Ref	Status	Started	Finished	Log	
Edit Del	B-152706	Finished	7/28/2016 4:31 PM	7/28/2016 4:31 PM	Started Processed Account OK: Business Network Team Processed Account OK: Business Technology Partners, Inc. Processed Account OK: Business Telecom Solutions Processed Account OK: Business World Processed Account OK: C & B Telecom Inc. Processed Account ...	
Edit Del	B-152707	Finished	7/28/2016 4:31 PM	7/28/2016 4:31 PM	Started Processed Account OK: Client Server Technology Group Processed Account OK: Cloud Communications Group LLC Processed Account OK: Cloud Contineo Processed Account OK: Cloud Source Services Processed Account OK: Cloud9 VoIP, LLC Processed Account OK...	

```
global with sharing class BatchTestFramework
{
    implements Database.Batchable<SObject>,
        Database.Stateful,
        Database.AllowsCallouts,
        Schedulable
    {

        class Parameters
        {
            String testParam1 = 'Default';
            Boolean updateDatabase = true;
        }

        Boolean testMode = Test.isRunningTest();
        Job__c job;
        Parameters params = new Parameters();

//*****

        global BatchTestFramework ()
        {
            // nothing to do
            // invoke like this:

            // JobSupport.submitJob ('BatchTestFramework', '{"testParam1":"DEFAULT OVERRIDDEN"}');
            // JobSupport.scheduleJob ('BatchTestFramework', '{"testParam1":"DEFAULT OVERRIDDEN"}', 6, 0);
        }

//*****

        global void execute (SchedulableContext sc)
        {
            Id scheduledJobId = sc.getTriggerId();
            JobSupport.submitJob('BatchTestFramework', scheduledJobId);
        }

//*****

        global database.querylocator start (Database.BatchableContext bc)
        {
            this.job = JobSupport.getJobForId (bc.getJobId());
            JobSupport.addLineToJobLog (this.job, 'Started', true);
            update this.job;

            if (this.job.Parameters__c != null)
            {
                this.params = (Parameters) JSON.deserialize (this.job.Parameters__c, Parameters.class);
            }

            System.debug(this.params);

            JobSupport.addLineToJobLog (this.job, 'Parameters: ' + this.params, false);

            String query = 'SELECT Id, Name FROM Account' +
                (this.testMode ? ' LIMIT 10' : 'LIMIT 100');

            Database.Querylocator ql = Database.getQueryLocator(query);

            return ql;
        }

//*****

        global void execute (Database.BatchableContext bc, list<Account> accounts)
        {
            Batch__c batch = new Batch__c (Job__c = this.job.Id);

            JobSupport.addLineToBatchLog (batch, 'Started', true);
            insert batch;

            Savepoint sp = Database.setSavepoint();

            for (Account a : accounts)
            {
                JobSupport.addLineToBatchLog (batch, 'Processed Account OK: ' + a.Name, false);
            }

            for (String m : DebugSupport.updateAndReturnErrors(accounts))
            {
                JobSupport.addLineToBatchLog (batch, 'Reportable error for Account ' + m, false, true);
            }

            if (this.params.updateDatabase == false) Database.rollback(sp);

            JobSupport.addLineToBatchLog (batch, 'Finished', true);
            update batch;
        }

//*****

        global void finish (Database.BatchableContext bc)
        {
            JobSupport.addLineToJobLog (this.job, 'Finished', true);
            update this.job;
        }
    }
}
```